Final Report

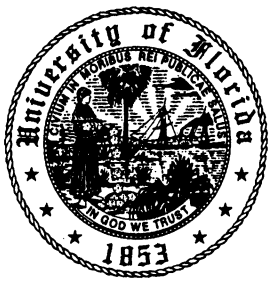
DEEP FOUNDATION DATABASE

SPONSORED BY

FLORIDA DEPARTMENT OF TRANSPORTATION

State Project No.: 99700-7603-119
UF Project No.: 49104504561 - 12
WPI No.: 0510699
Contract No.: BA - 527

November 2000



Investigators: John L. Davidson
Roberto Fernandez

Department of Civil and Coastal Engineering
University of Florida
Gainesville, Florida 32611-6580

Engineering & Industrial Experimentation Station

Final Report

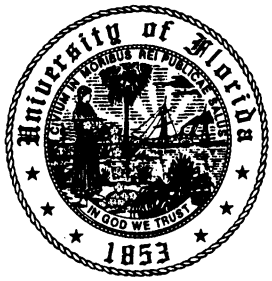
DEEP FOUNDATION DATABASE

SPONSORED BY

FLORIDA DEPARTMENT OF TRANSPORTATION

State Project No.: 99700-7603-119
UF Project No.: 49104504561 - 12
WPI No.: 0510699
Contract No.: BA - 527

November 2000



Investigators: John L. Davidson
Roberto Fernandez

Department of Civil and Coastal Engineering
University of Florida
Gainesville, Florida 32611-6580

Engineering & Industrial Experimentation Station

Technical Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
WPI 0510699		

4. Title and Subtitle	5. Report Date
Deep Foundations Databases, Phase III	November 2000
	6. Performing Organization Code
7. Author(s)	8. Performing Organization Report No.
Dr. J. L. Davidson and R. Fernandez	4910-4504-561-12
9. Performing Organization Name and Address	10. Work Unit No. (TRAIS)
University of Florida Department of Civil and Coastal Engineering 345 Weil Hall / P.O. Box 116580 Gainesville, FL 32611-6580	99700-7603-119
	11. Contract or Grant No.
	BA-527
12. Sponsoring Agency Name and Address	13. Type of Report and Period Covered
Florida Department of Transportation Research Management Center 605 Suwannee Street, MS 30 Tallahassee, FL 32301-8064	Final Report August 29, 1996 - August 31, 2000
	14. Sponsoring Agency Code

15. Supplementary Notes

Prepared in cooperation with the Federal Highway Administration

16. Abstract

The University of Florida Geotechnical group has over the years developed a number of deep foundation databases. These used the spreadsheet application Lotus 123, which provided excellent data manageability but had limited storage capacity.

As part of this phase of the research, a completely new database was designed, combining previous driven pile and drilled shaft databases into a single database, capable of storing and managing the data more efficiently. Microsoft Access 97, a database application, was chosen as the design tool. The database which contained 213 piles and 222 shafts was intended for data storage exclusively. Data could be exported to Microsoft Excel and it used to create input files for capacity programs, graphs, etc. The database was structured after the earlier Lotus pile and shaft databases, which were designed for evaluation of foundation capacities. Only piles or shafts with static load tests were included and only the closest (or most representative) SPT boring was included. Each record consisted of a single pile or shaft and a single SPT log.

A second database has been developed, using Microsoft Access 2000, which is instead based on Projects. Any record can now include multiple piles and/or shafts and multiple insitu tests. Since an important future use of the database will be to query based on location, a major effort of the current project has been to locate all piles, shafts and insitu borings. These are referenced (x and y, East and North) to a single identifiable point, e.g., the center of the bridge. Later it is expected that these reference points will be accurately located as to latitude and longitude. The database currently contains 189 projects, 441 piles, 419 drilled shafts and 1063 insitu tests.

17. Key Words	18. Distribution Statement
Database, Piles, Drilled Shaft, Insitu test, Load test	No restrictions. This document is available to the public through the National Technical Information Service, Springfield, VA, 22161

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	259	

Form DOT F 1700.7 (8-72)

DISCLAIMER

"The opinions, findings and conclusions expressed in this publication are those of the authors and not necessarily those of the Florida Department of Transportation or the U.S. Department of Transportation.

Prepared in cooperation with the State of Florida Department of Transportation and the U.S. Department of Transportation."

SI* (MODERN METRIC) CONVERSION FACTORS

APPROXIMATE CONVERSIONS TO SI UNITS

Symbol	When You Know	Multiply By	To Find	Symbol
	LENGTH			
in	inches	25.4	millimeters	mm
ft	feet	0.305	meters	m
yd	yards	0.914	meters	m
mi	miles	1.61	kilometers	km
	AREA			
in^2	square inches	645.2	square millimeters	mm^2
ft^2	square feet	0.093	square meters	m^2
yd^2	square yards	0.836	square meters	m^2
ac	acres	0.405	hectares	ha
mi^2	square miles	2.59	square kilometers	km^2
	VOLUME			
fl oz	fluid ounces	29.57	milliliters	ml
gal	gallons	3.785	liters	l
ft^3	cubic feet	0.028	cubic meters	m^3
yd^3	cubic yards	0.765	cubic meters	m^3
NOTE: Volumes greater than 1000 l shall be shown in m^3.				
	MASS			
oz	ounces	28.35	grams	g
lb	pounds	0.454	kilograms	kg
T	short tons (2000 lb)	0.907	megagrams	Mg
	TEMPERATURE (exact)			
°F	Fahrenheit temperature	5(F-32)/9 or (F-32)/1.8	Celcius temperature	°C
	ILLUMINATION			
fc	foot-candles	10.76	lux	l
fl	foot-Lamberts	3.426	candela/m^2	cd/m^2
	FORCE and PRESSURE or STRESS			
lbf	poundforce	4.45	newtons	N
psi	poundforce per square inch	6.89	kilopascals	kPa

APPROXIMATE CONVERSIONS FROM SI UNITS

Symbol	When You Know	Multiply By	To Find	Symbol
	LENGTH			
mm	millimeters	0.039	inches	in
m	meters	3.28	feet	ft
m	meters	1.09	yards	yd
km	kilometers	0.621	miles	mi
	AREA			
mm^2	square millimeters	0.0016	square inches	in^2
m^2	square meters	10.764	square feet	ft^2
m^2	square meters	1.195	square yards	yd^2
ha	hectares	2.47	acres	ac
km^2	square kilometers	0.386	square miles	mi^2
	VOLUME			
ml	milliliters	0.034	fluid ounces	fl oz
l	liters	0.264	gallons	gal
m^3	cubic meters	35.71	cubic feet	ft^3
m^3	cubic meters	1.307	cubic yards	yd^3
	MASS			
g	grams	0.035	ounces	oz
kg	kilograms	2.202	pounds	lb
Mg	megagrams	1.103	short tons (2000 lb)	T
	TEMPERATURE (exact)			
°C	Celcius temperature	1.8C + 32	Fahrenheit temperature	°F
	ILLUMINATION			
lx	lux	0.0929	foot-candles	fc
cd/m^2	candela/m^2	0.2919	foot-Lamberts	fl
	FORCE and PRESSURE or STRESS			
N	newtons	0.225	poundforce	lbf
kPa	kilopascals	0.145	poundforce per square inch	psi

* SI is the symbol for the International System of Units. Appropriate rounding should be made to comply with Section 4 of ASTM E380.

(Revised August 1992)

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Brief History of UF Databases

The University of Florida Geotechnical group has developed a number of deep foundation databases. In 1987 Michael Sharp established the first driven pile database, using Lotus 123 Version 2.0. This database contained 28 piles. The data related to each pile were stored in individual Lotus files. An index file summarized the data. Macros in each pile file generated input files for four pile capacity prediction programs. In 1992 Pedro Ruesta designed a new driven pile database using Lotus 123 Version 3.1. This database contained 62 piles and had a maximum storage capacity of 120 piles. The data related to each pile were stored in different sheets contained in the single large Lotus file. Twenty-two macros were included to automate the manipulation of data, generate input files for capacity programs, plot graphs, etc.

In 1993 Lawrence Spears created a corresponding drilled shaft database also using Lotus 123 Version 3.1. This database contained 86 shafts and had a maximum storage capacity of 120 shafts. The data related to each shaft were stored in different sheets contained in a single Lotus file. Again, macros were included to facilitate data manipulation.

In 1995 Steven Dornic developed a new driven pile database using Lotus Version 3.1. This database contained 213 driven piles and had a maximum storage capacity of 1500 piles. The data related to each pile were stored in individual Lotus files and then accessed from an index file. This complex database contained 30 menus, 179 macros and

over 2000 lines of macro. In 1997 Erica Smith made some improvements to this database while using it to evaluate the static capacity program SPT94. Later in 1998, Ike Egboh converted the database from Lotus DOS Version 3.1 to the Windows Version Release 5.

In 1999 Roberto Fernandez designed a completely new database combining the previous driven pile and drilled shaft databases into a single database, capable of storing and managing the data more efficiently. All previous databases were developed using the spreadsheet application Lotus 123; this type of application provides excellent data manageability but has limited storage capacity. Microsoft Access 97, a database application, was chosen as the design tool. The database which contained 213 piles and 222 shafts was intended for data storage exclusively. Data could be exported to Microsoft Excel, currently the most popular spreadsheet application, and it used to create input files for capacity programs, graphs, etc. The database was entirely menu driven; the user did not need to be at all familiar with Access to use it. The progress report, dated August 1999, describing this database is included in this final report as Appendix C.

1.2 The Latest UF Database

This first Access 97 database was structures after the earlier Lotus pile and shaft databases, which were designed for evaluation of foundation capacities. Only piles or shafts with static load tests were included and only the closest (or most representative) SPT boring was included. Each record consisted of a single pile or shaft and a single SPT log. The latest database, developed using Microsoft Access 2000, is instead based on Project. Any record can now include multiple piles and/or shafts and multiple insitu tests. Projects could even be included which involved no deep foundation testing, just

insitu test logs.

Since an important future use of the database will be to query based on location, e.g., all Static Pile Load Testing performed in the Tampa area, a major effort of the current project has been to locate all piles, shafts and insitu borings. These are referenced (x and y, East and North) to a single identifiable point, e.g., the center of the bridge. Later it is expected that these reference points will be accurately located as to latitude and longitude.

The database currently consists of 189 projects, 441 piles, 419 drilled shafts and 1063 insitu tests.

1.3 Report Structure

Chapter 2 of this report covers in some detail the database architecture. Sections describe the tables and their relationships, the types of forms used, reports and queries. A final section lists all the units and conversions employed.

Chapter 3 discusses how to use the database and contains sections on opening the database, creating new records, searching for, editing and outputting existing records. A final section describes how to optimize database performance.

Chapter 4 provides some detail on the current content of the database and introduces the equations and procedures for locating piles, shafts and insitu tests relative to a chosen origin. Appendix A provides a listing of all the records in the database.

Appendix B consists of a listing of all the Visual Basic code contained in the database. A great number of comment statements have been included to assist anyone who in the future may need to add to or modify the database.

Appendix C consists of the August 1999 progress report describing the first pile and shaft based Access database.

1.4 The Future of the UF Database

The primary purpose of recent database research, performed under a no cost extension provision, was to provide the initial set-up for a future major database effort. The software was switched from Lotus 123 to Access and the database structure was changed from one based on individual piles and shafts to one based on projects. A large amount of new data has been added and the bases of a global positioning system introduced.

It is expected that among the goals of this future research will be to:

(a) convert the database Access format to web based Oracle software

(b) locate all project sites (latitude and longitude) by accurate GPS procedures

(c) use the software ArcView to query and analyze data and map results

(d) continue the process of adding and updating the data content.

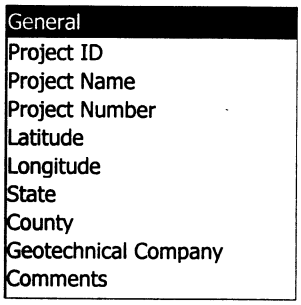


Figure 2.1 The General Table

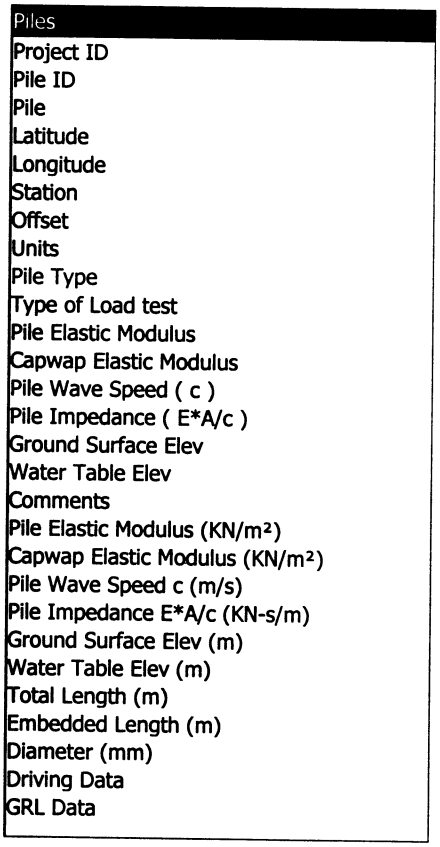


Figure 2.2 The Piles Table

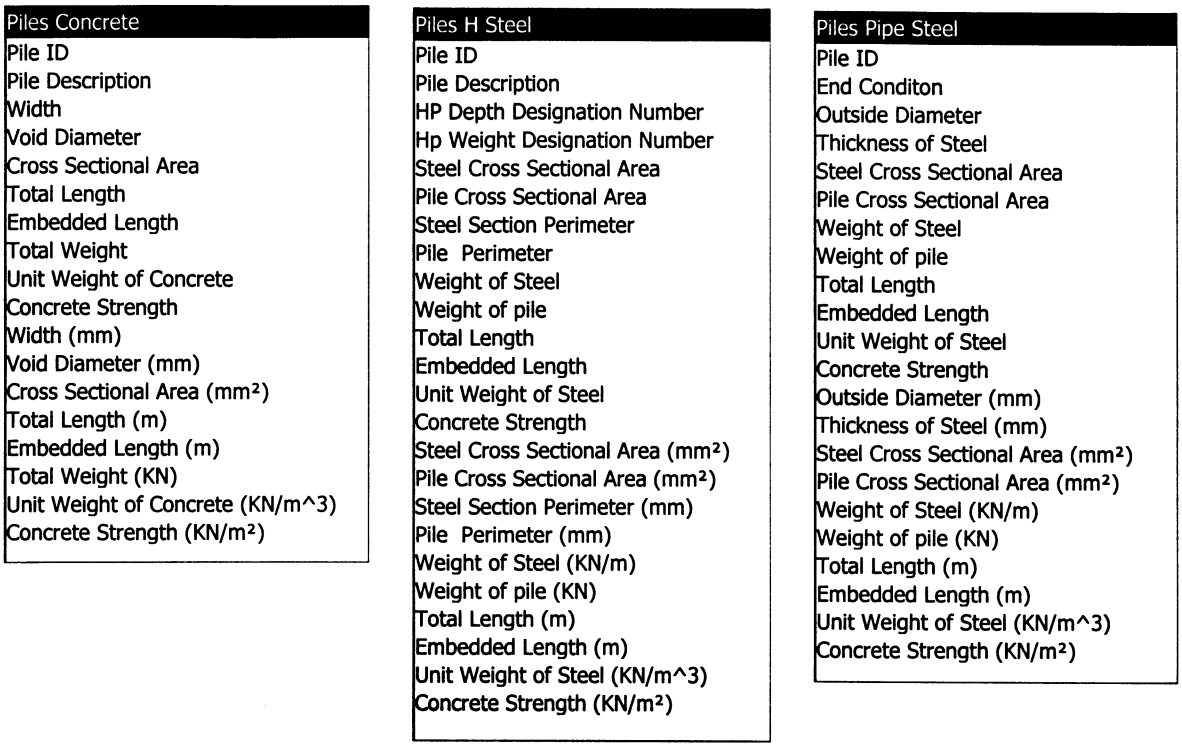


Figure 2.3 The Concrete, H-Pile and Pipe Pile Tables

The data related to pile penetration are stored in two tables, Figure 2.4; the Pile Driving table contains information that can be stored in a single-record field while the Piles Penetration table contains the penetration log. Similarly, the Pile Driving Analyzer (PDA) data are stored in two tables, Figure 2.5; the GRL Data table contains all the information that can be stored in a single-record field while the GRL PDA Trace table contains the PDA log.

The Shafts table, Figure 2.6, contains the data related to the drilled shaft geometry and material properties. The analysis of drilled shafts founded in rock (Intermediate Geomaterials or IGM) using capacity prediction software such as SHAFT98 requires some additional parameters, which can be stored in a multi-record table named Shafts IGM.

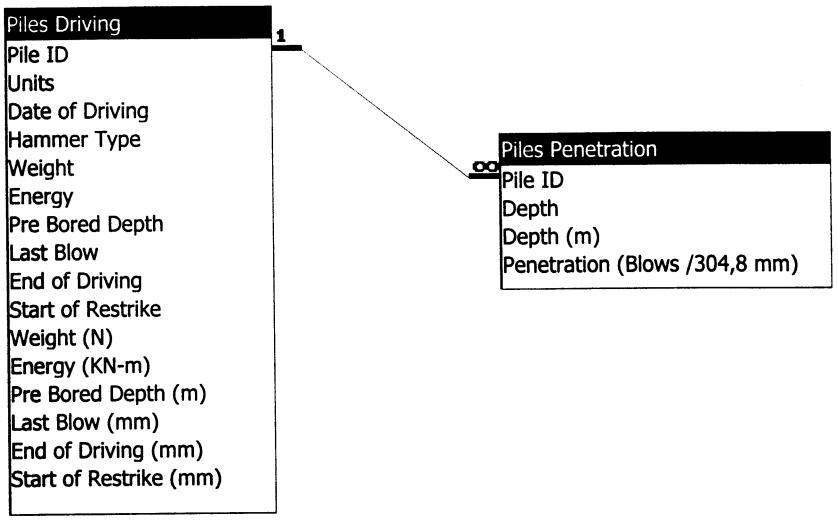


Figure 2.4 The Pile Driving and Piles Penetration Tables

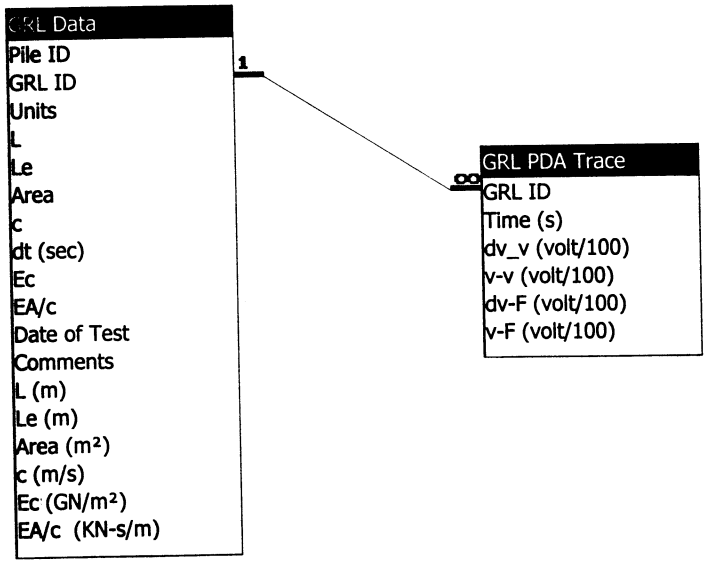


Figure 2.5 The GRL Data and GRL PDA Trace Tables

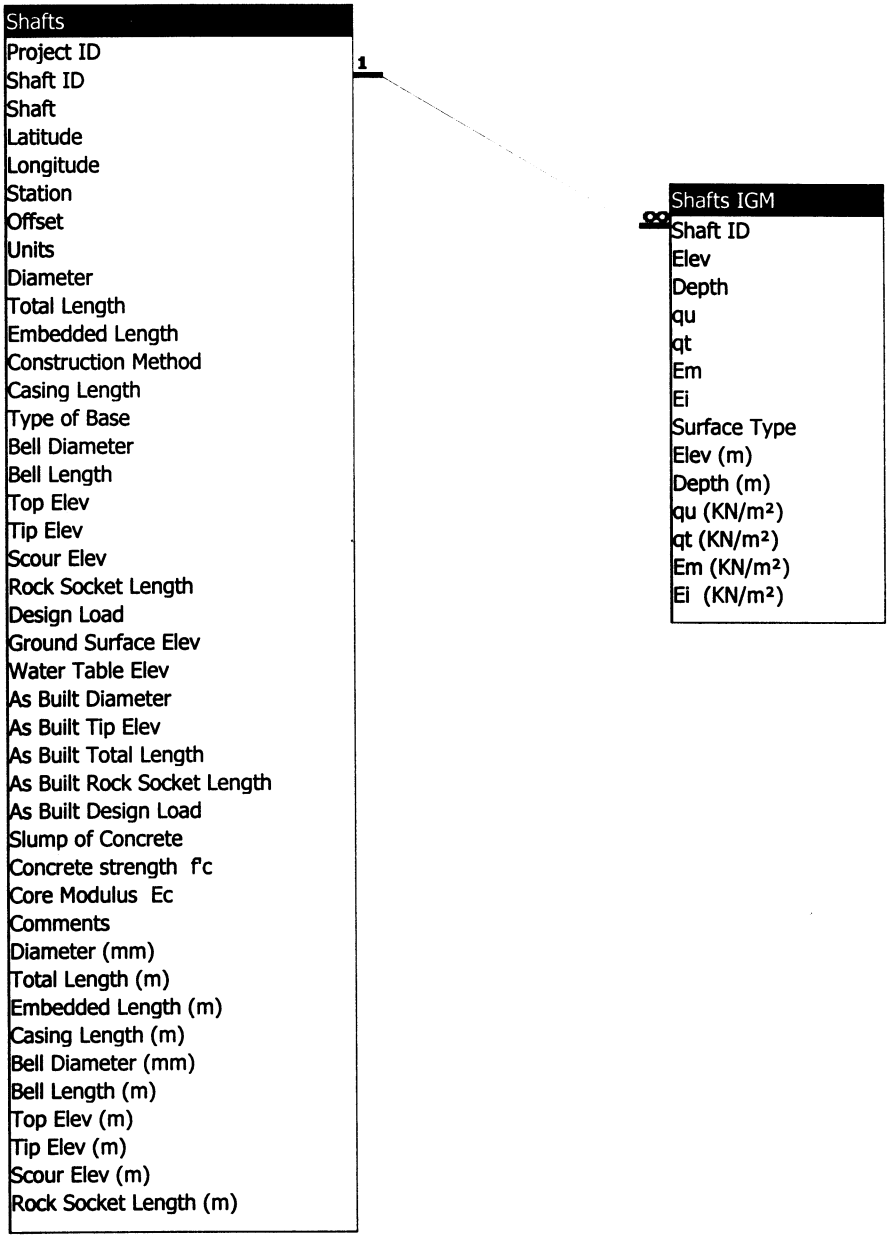


Figure 2.6 The Shafts and Shafts IGM Tables

The InSitu Tests table contains the data related to the insitu tests that can be stored in single-record fields. Since several insitu tests can be related to a single project, this table also functions as an index for the Insitu SPT, Insitu CPT, Insitu DMT and Insitu PMT tables. These five tables are shown in Figure 2.7.

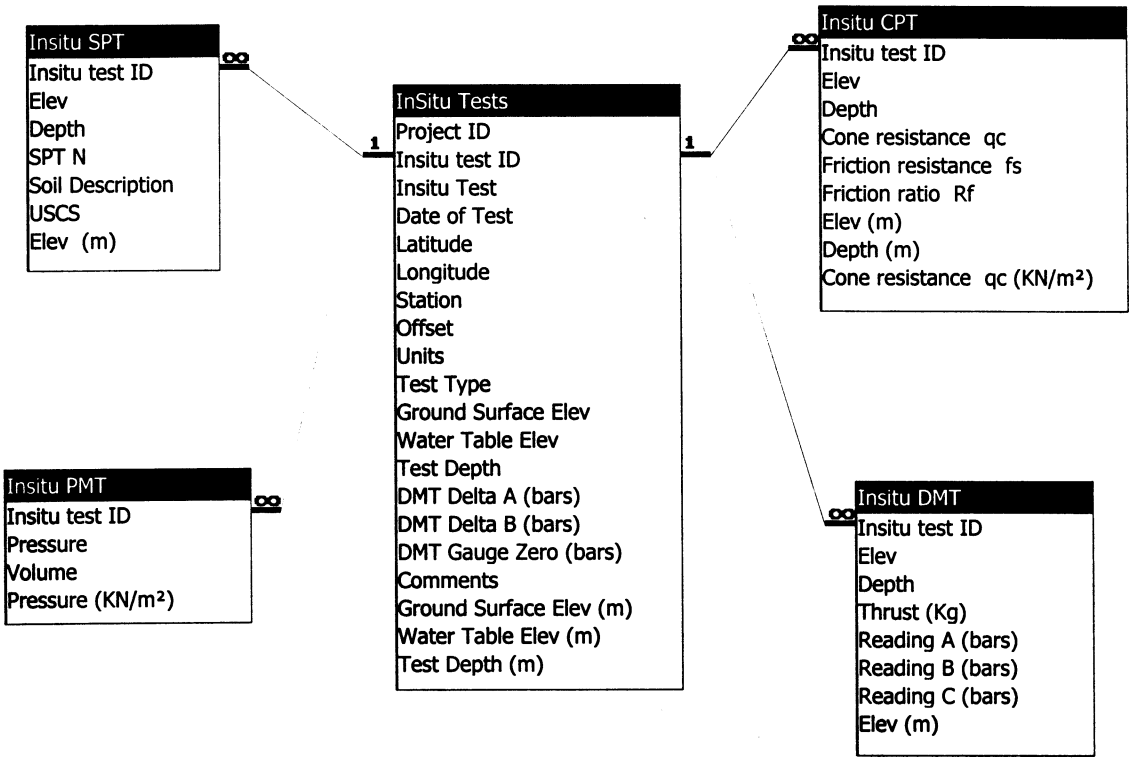


Figure 2.7 The InSitu, Insitu SPT, Insitu CPT, Insitu DMT and Insitu PMT Tables

In like manner, the Load Tests table contains data related to the load tests that can be stored in single-record fields and provides an index for the Load Test Static, Load Test Statnamic, Load Test Osterberg and the Load Test Strain Gauge tables. These tables are included in Figure 2.8.

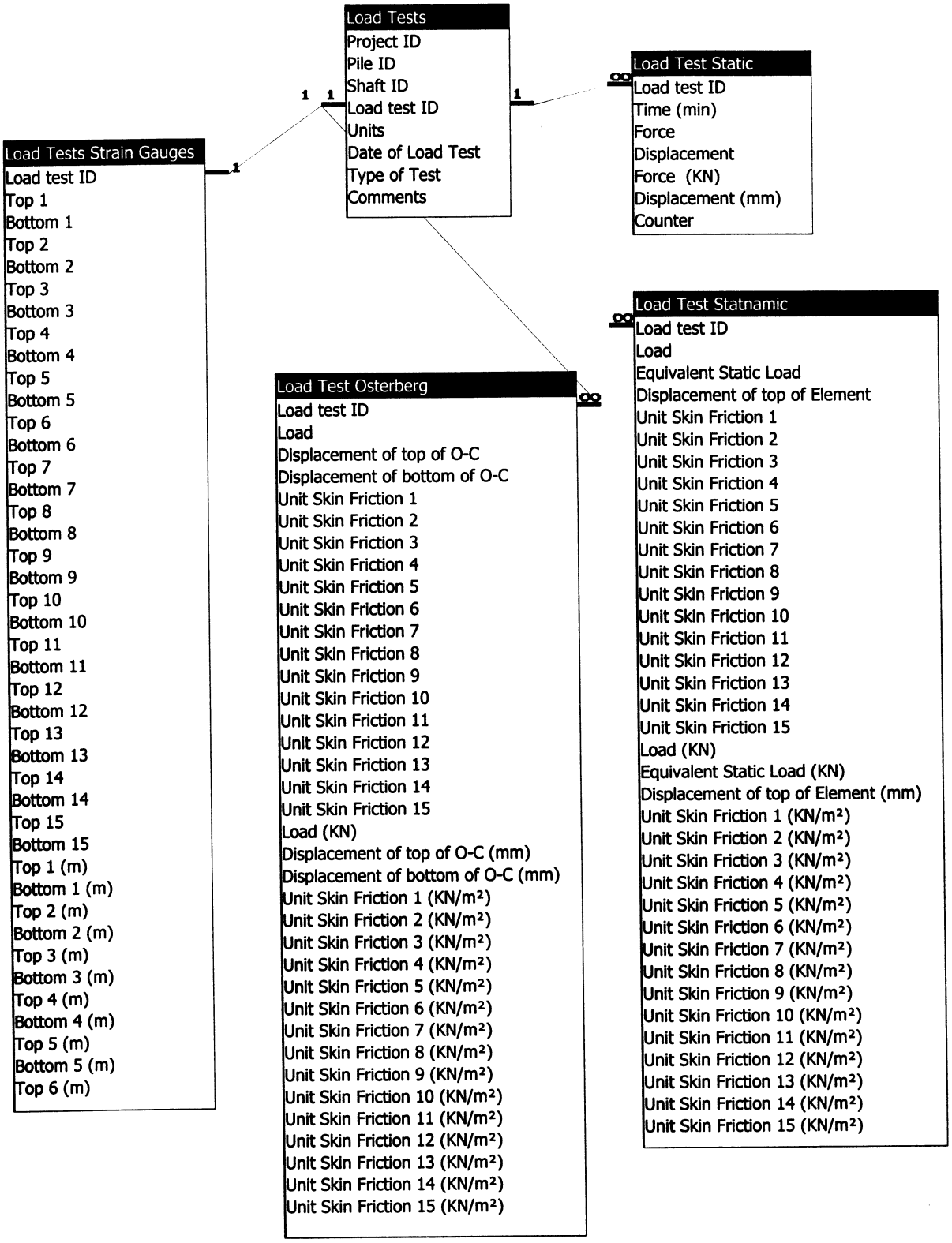


Figure 2.8 The Load Tests, Load Test Static, Load Test Statnamic, Load Test Osterberg and Load Test Strain Gauges Tables

Relationships between the tables are defined using index fields to relate the data in the different tables and to assure data integrity. These relationships are presented in Figure 2.9. The General table is the primary table of the database. Many of the other tables in the database are directly related to it through an index field named Project ID. Other tables are related to associated index tables, which are then related to the General table. For example, the insitu tests tables are related to the InSitu Tests table through the Insitu Test ID index field, and this table is related to the General table through the Project ID field. In a similar manner, the load tests are related to the Load Tests table through the Load Test ID index field; this table is then related to the Piles or Shafts tables through the Pile ID or Shaft ID index fields, and finally to the General table through the Project ID field.

Each time a record is erased from an index table, the database automatically removes the record from the related tables, assuring the integrity of the data. An infinity symbol at the end of a relationship line means that the related table can have more than one record for each record in the primary table.

2.2 Database Forms

The forms in the Deep Foundation Database can be classified into two groups; those used to add, edit or view data in the database tables and those used to control database operations.

2.2.1 Add/Edit Forms

Data processing forms can operate directly on the tables when only one table is involved. However, if a single form is used for adding or editing data in more than one table, a query containing the fields from the different tables has to be created and the form has to be based on the query. Alternatively, a form based on a single table and containing

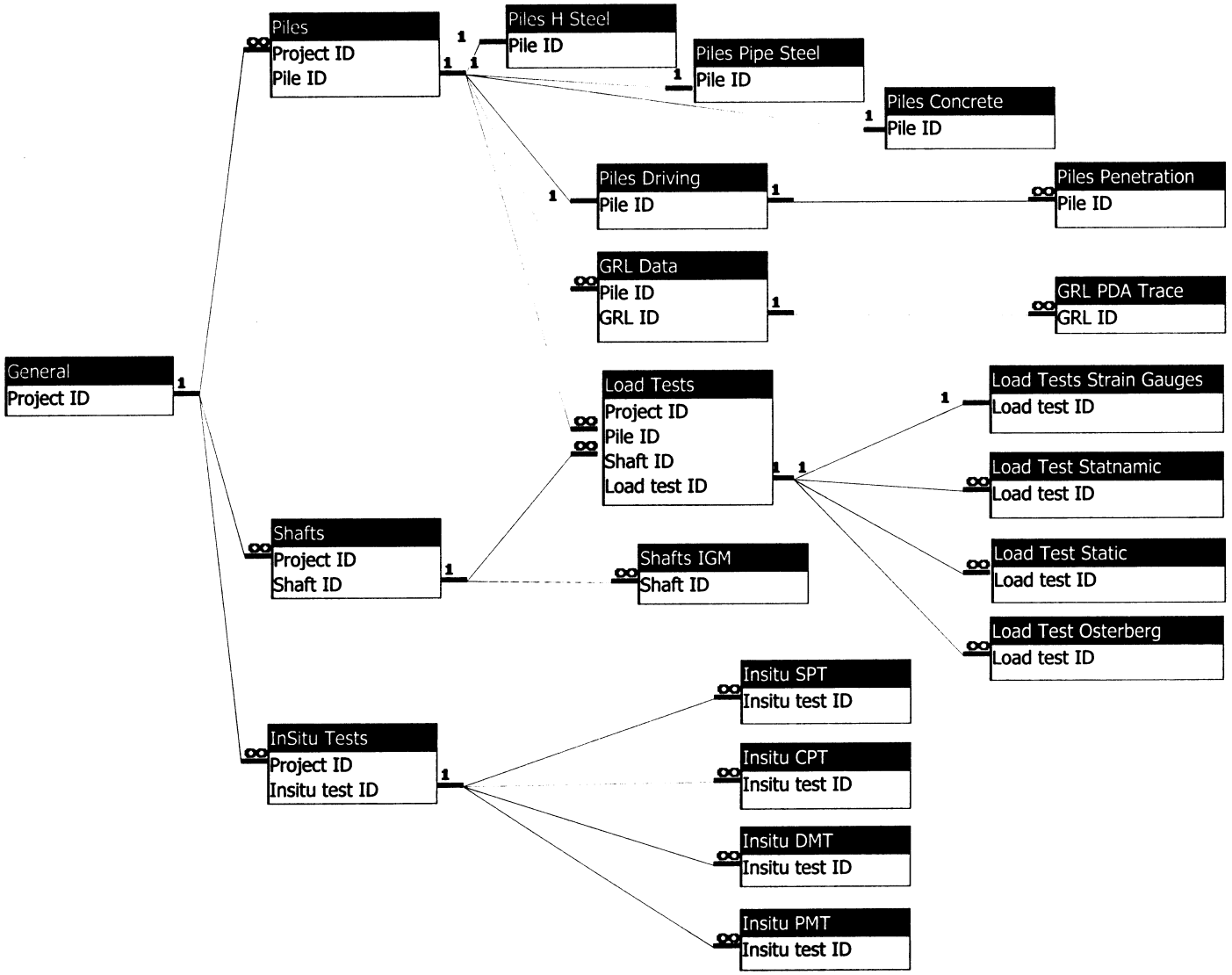


Figure 2.9 Table Relationships

subforms on other tables can be implemented. The second approach was considered when creating the forms for adding, editing and displaying data in this database. The forms are related one to another in the same manner as the tables. The master forms are based on pimary tables and child forms (or subforms) are based on the related tables.

The forms for adding and editing the data are presented in Figures 2.10 through 2.27. All the data are entered through a single form named frmGeneral, which contains several subforms related to the different database tables. The accessibility of the subforms is controlled by means of tabs. For example, on the opening form there are four tabs, General, Piles, Shafts and Insitu Tests. If Piles is chosen four additional tabs are available, General, Driving, GRL Data and Load Tests. These input forms will be discussed in more detail in Chapter 3.

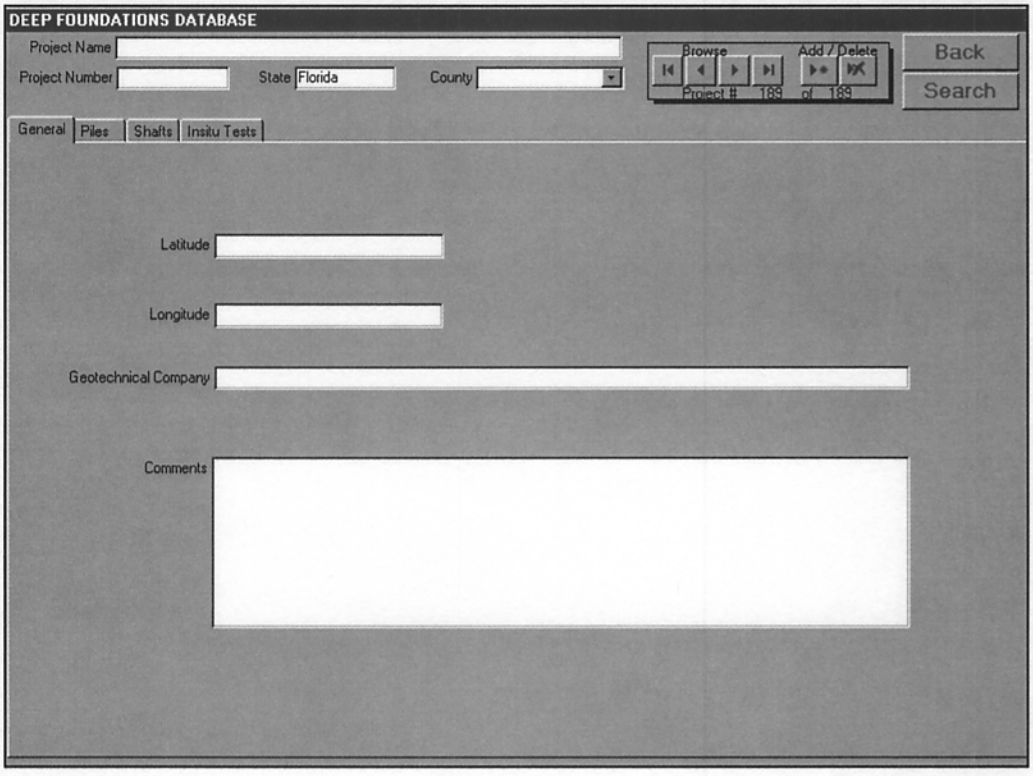


Figure 2.10 Add/Edit Form (General)

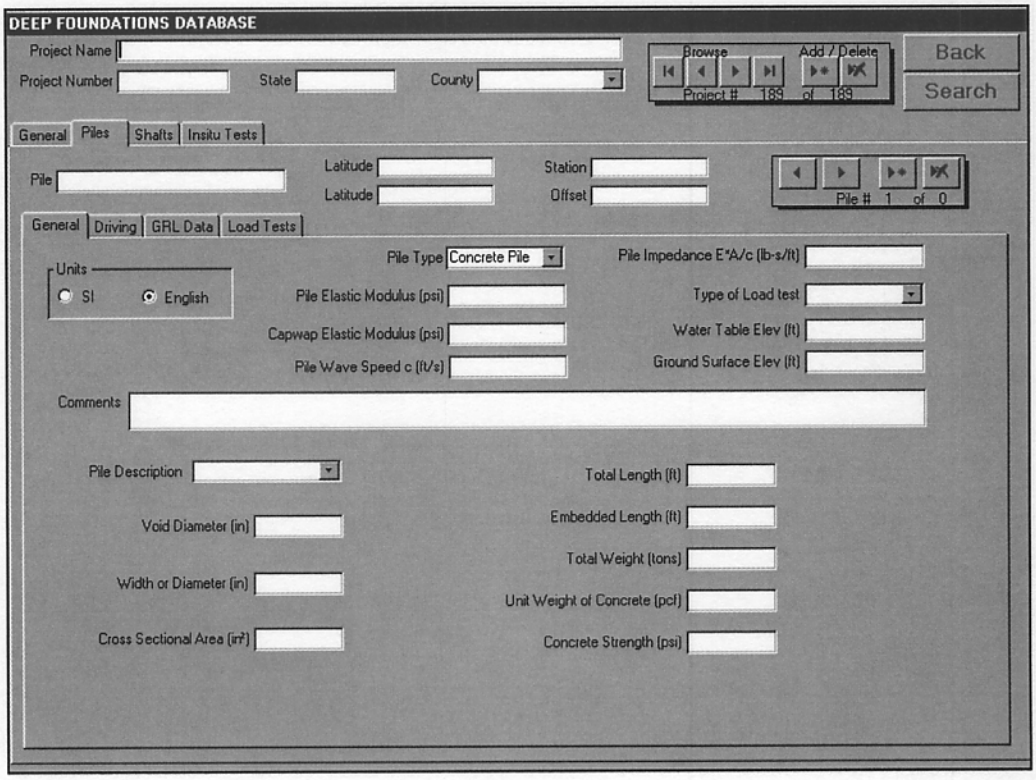


Figure 2.11 Add/Edit Form (Piles, Concrete Piles)

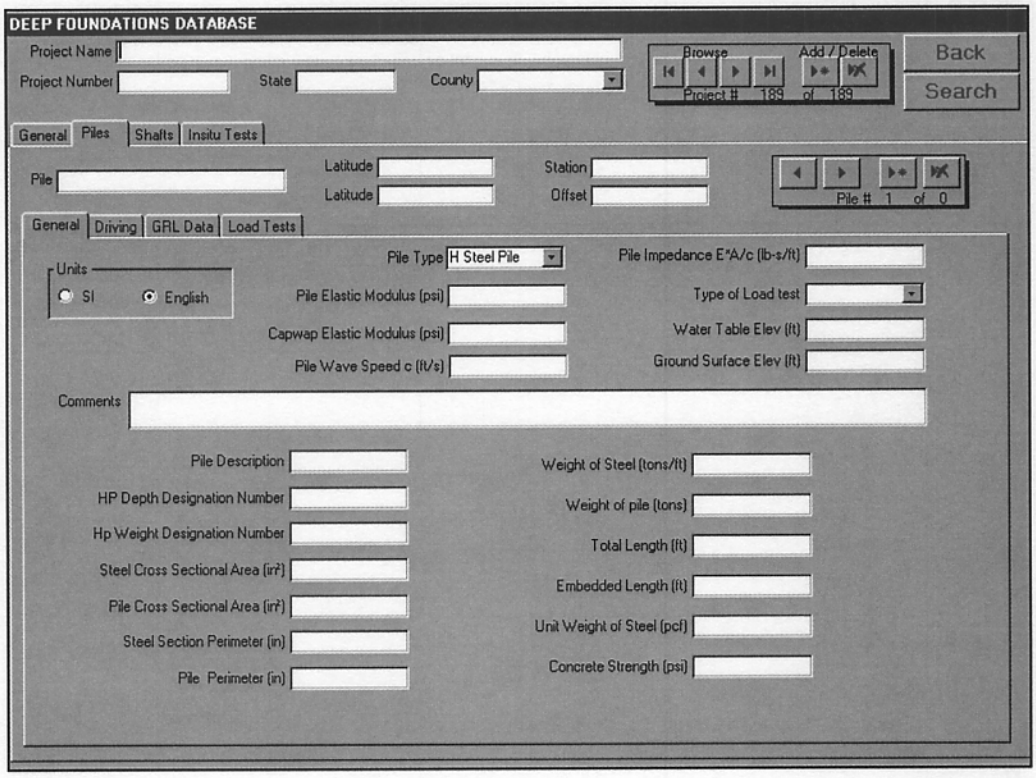


Figure 2.12 Add/Edit Form (Piles, Steel H-Piles)

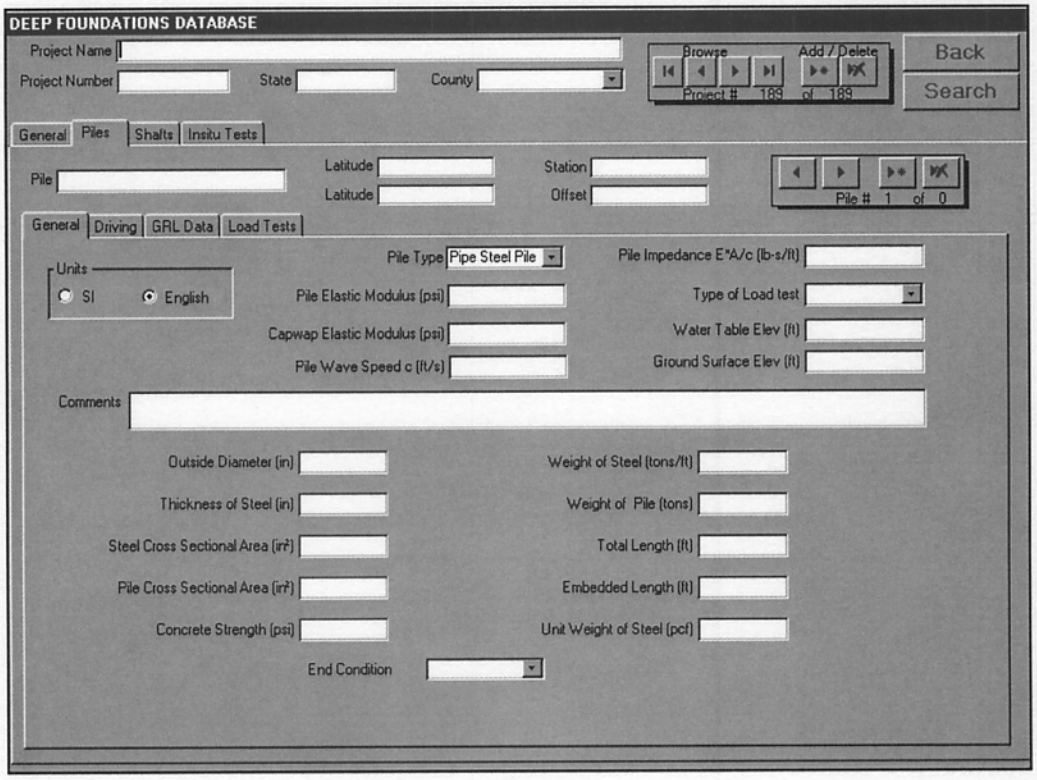


Figure 2.13 Add/Edit Form (Piles, Steel Pipe Piles)

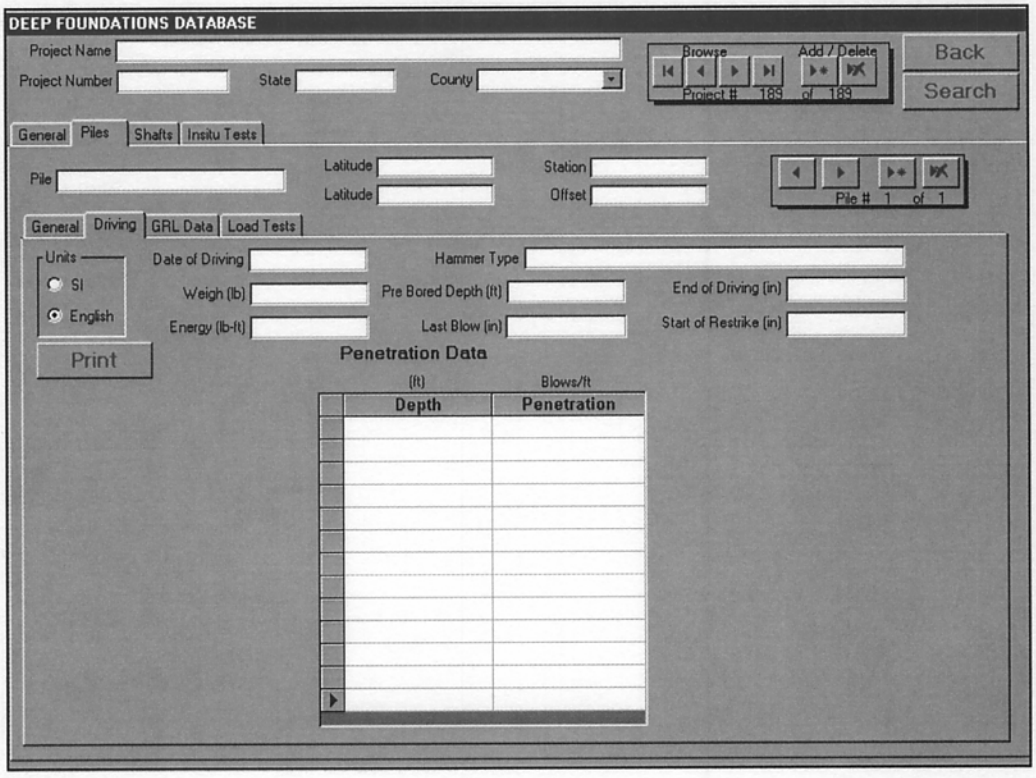


Figure 2.14 Add/Edit Form (Piles, Pile Driving Data)

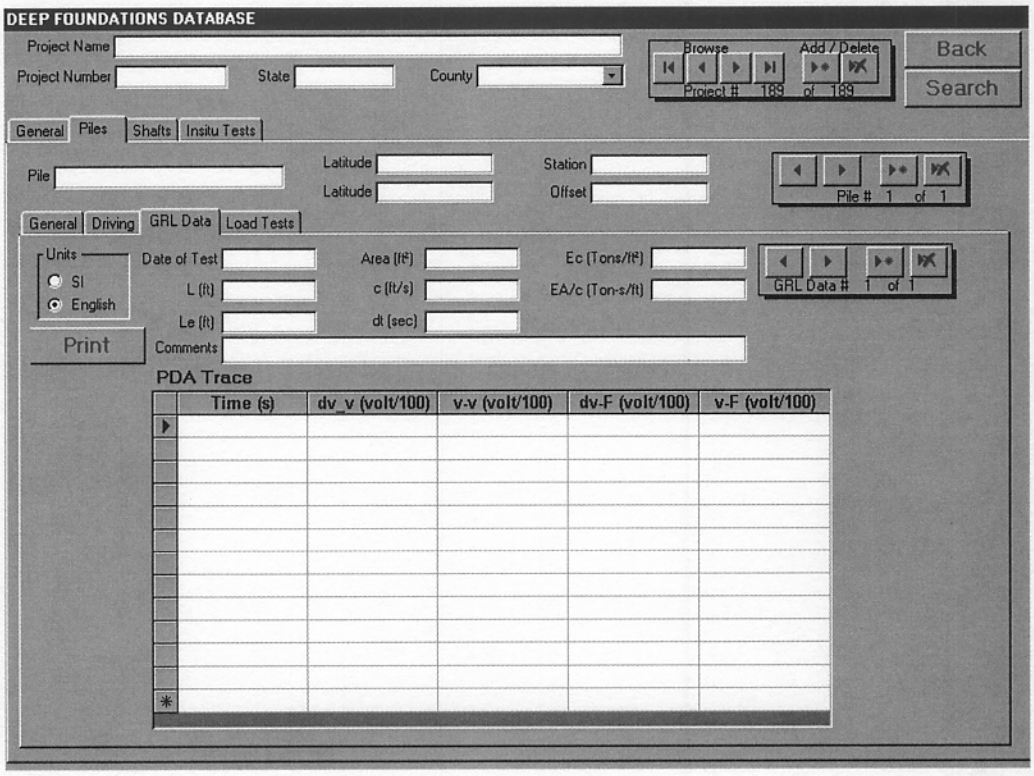


Figure 2.15 Add/Edit Form (Piles, GRL Data)

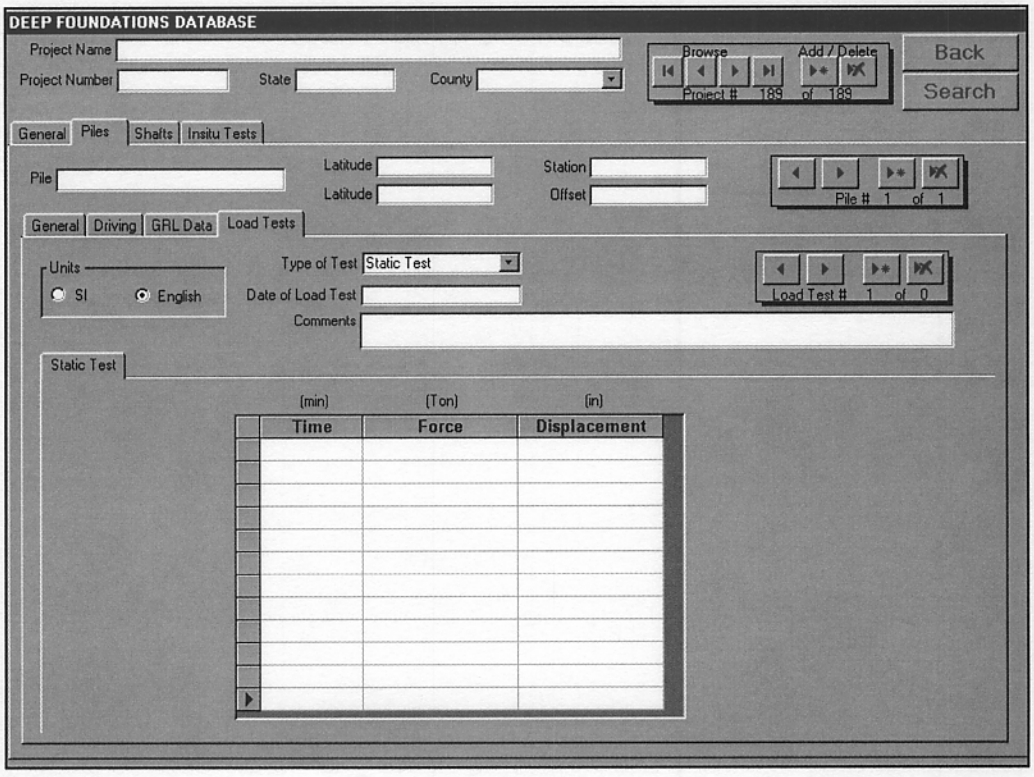


Figure 2.16 Add/Edit Form (Piles, Static Load Test)

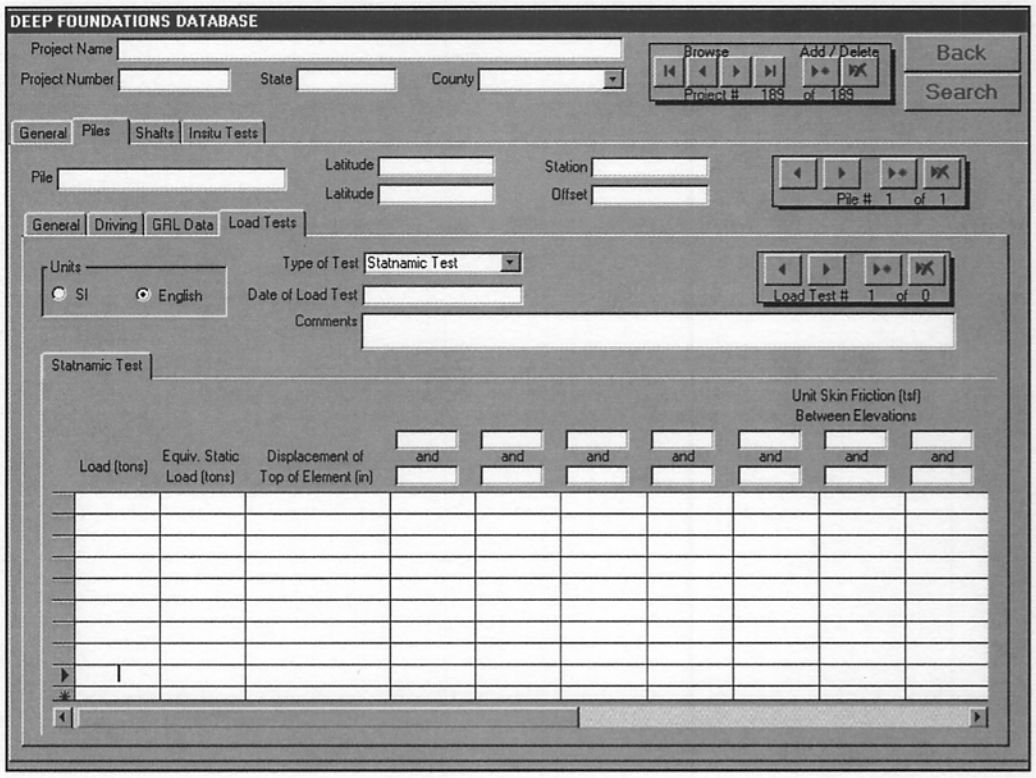


Figure 2.17 Add/Edit Form (Piles, Statnamic Load Test)

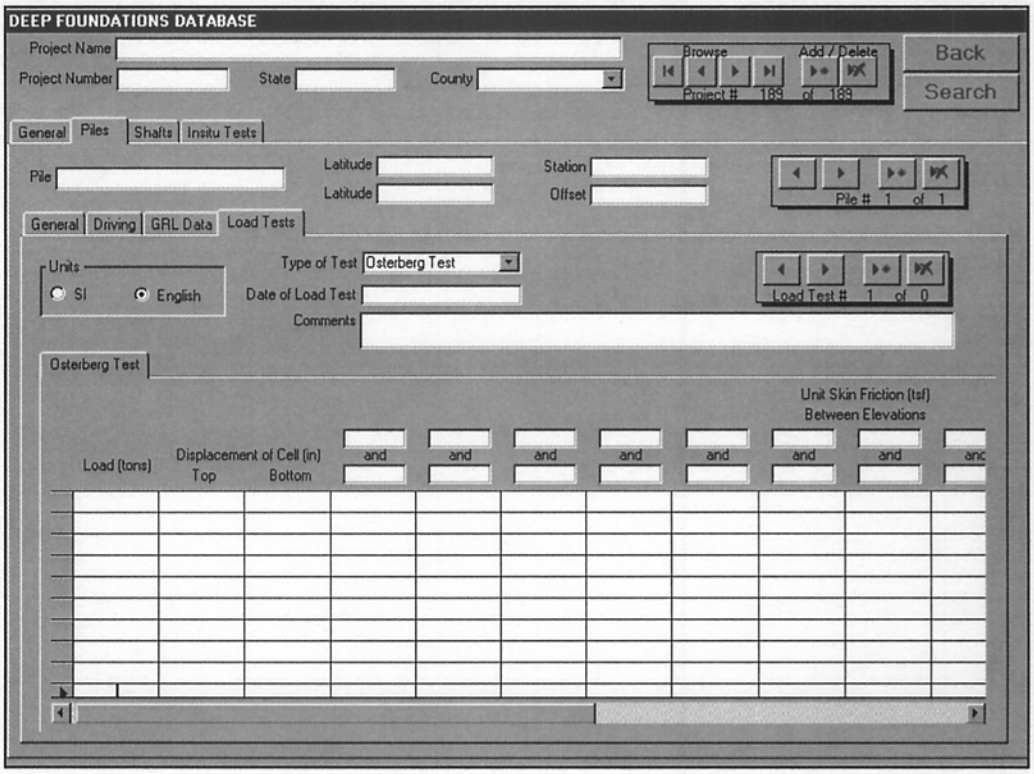


Figure 2.18 Add/Edit Form (Piles, Osterberg Load Test)

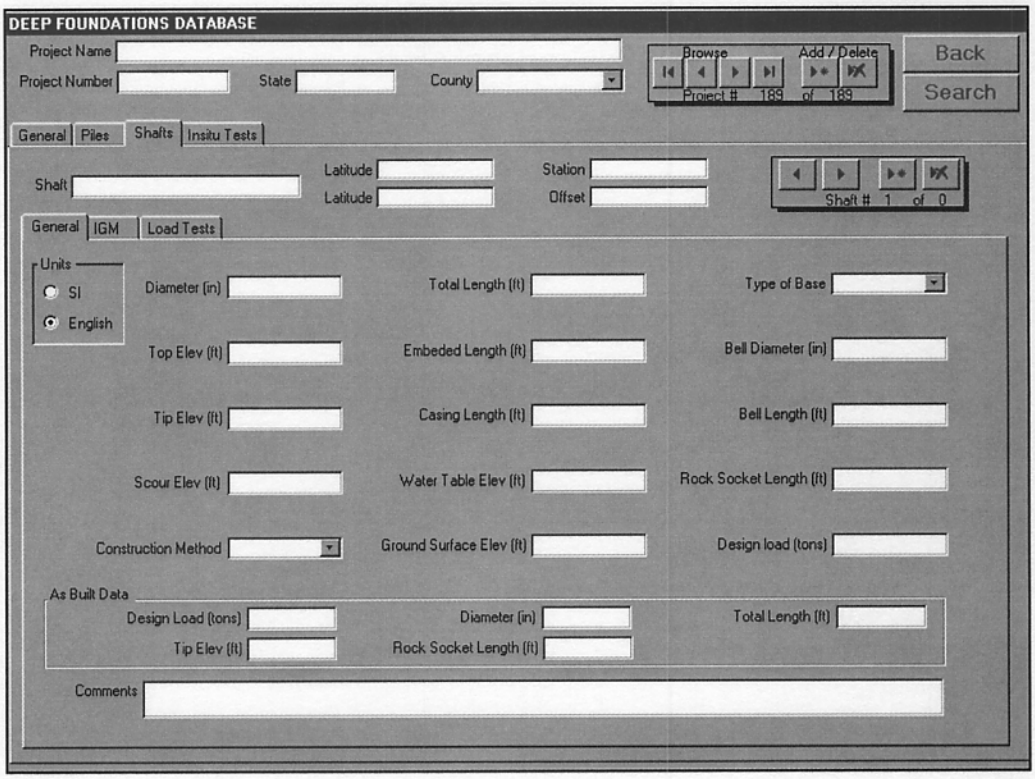


Figure 2.19 Add/Edit Form (Shafts)

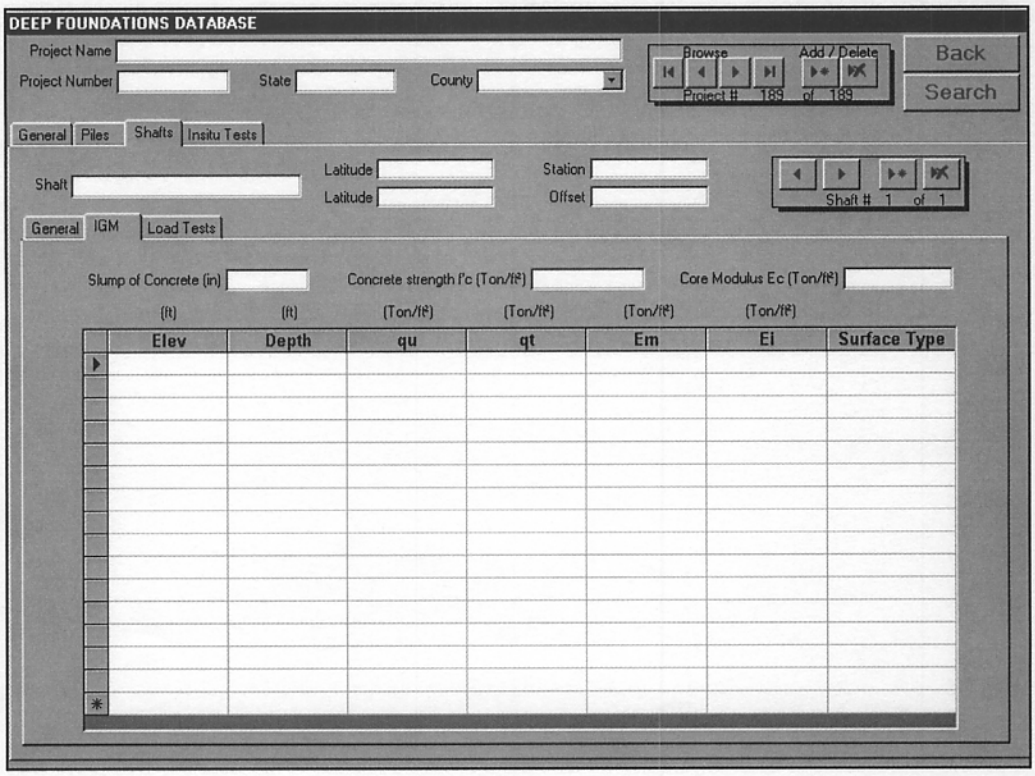


Figure 2.20 Add/Edit Form (Shafts, IGM)

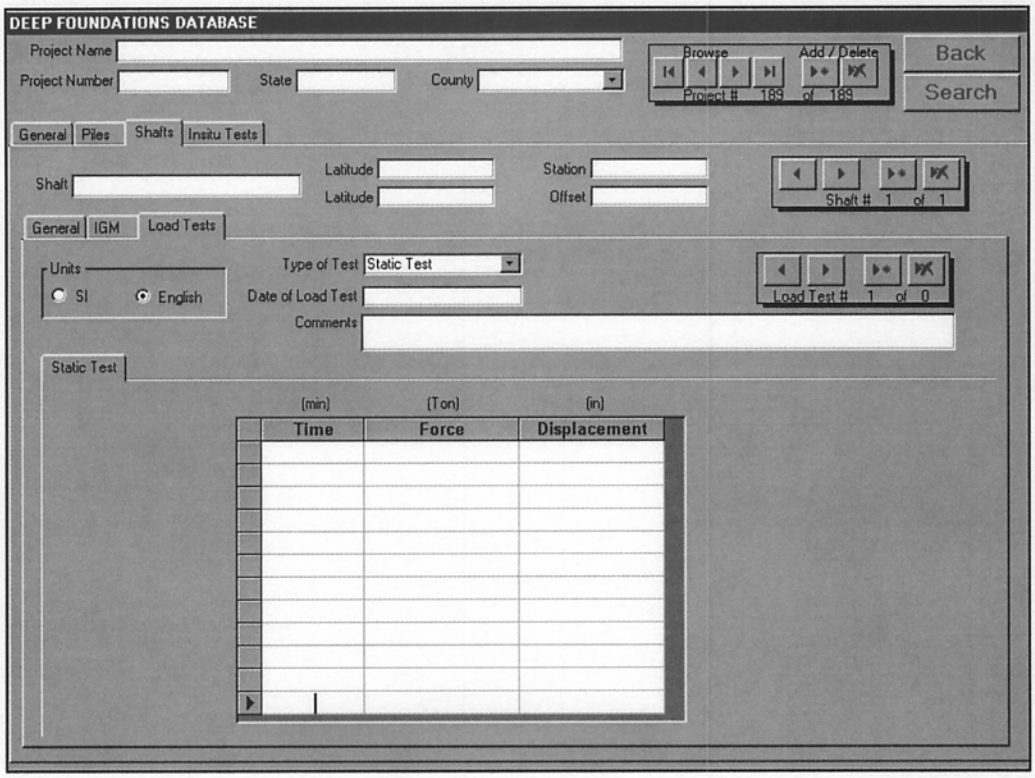


Figure 2.21 Add/Edit Form (Shafts, Static Load Test)

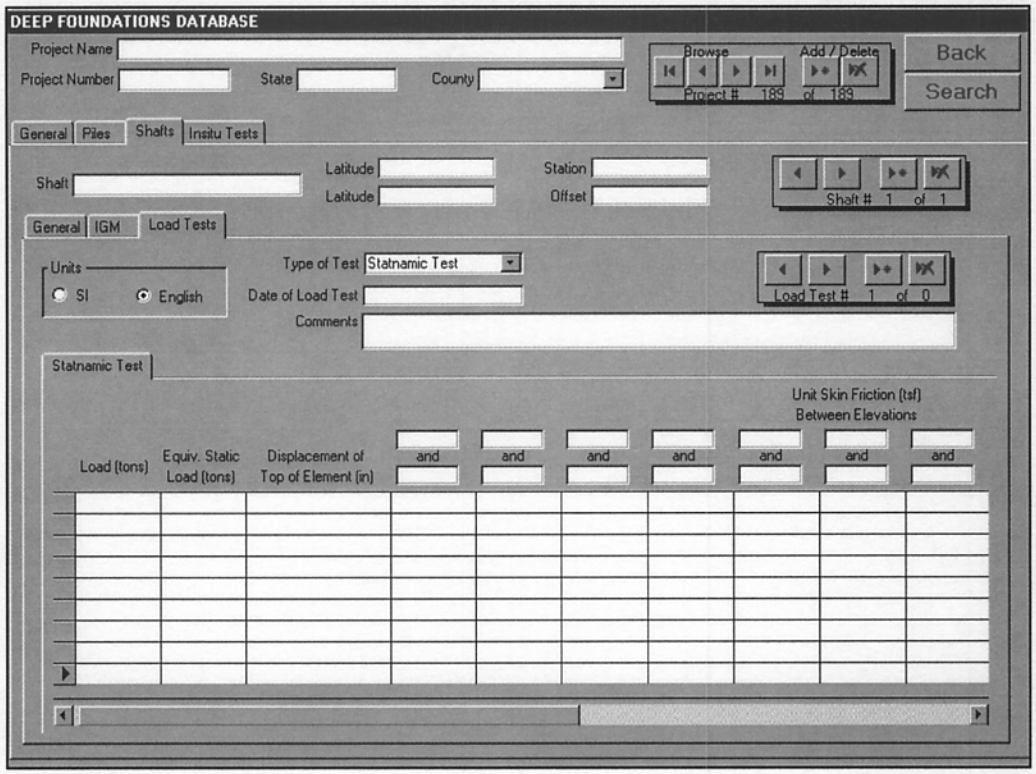


Figure 2.22 Add/Edit Form (Shafts, Statnamic Load Test)

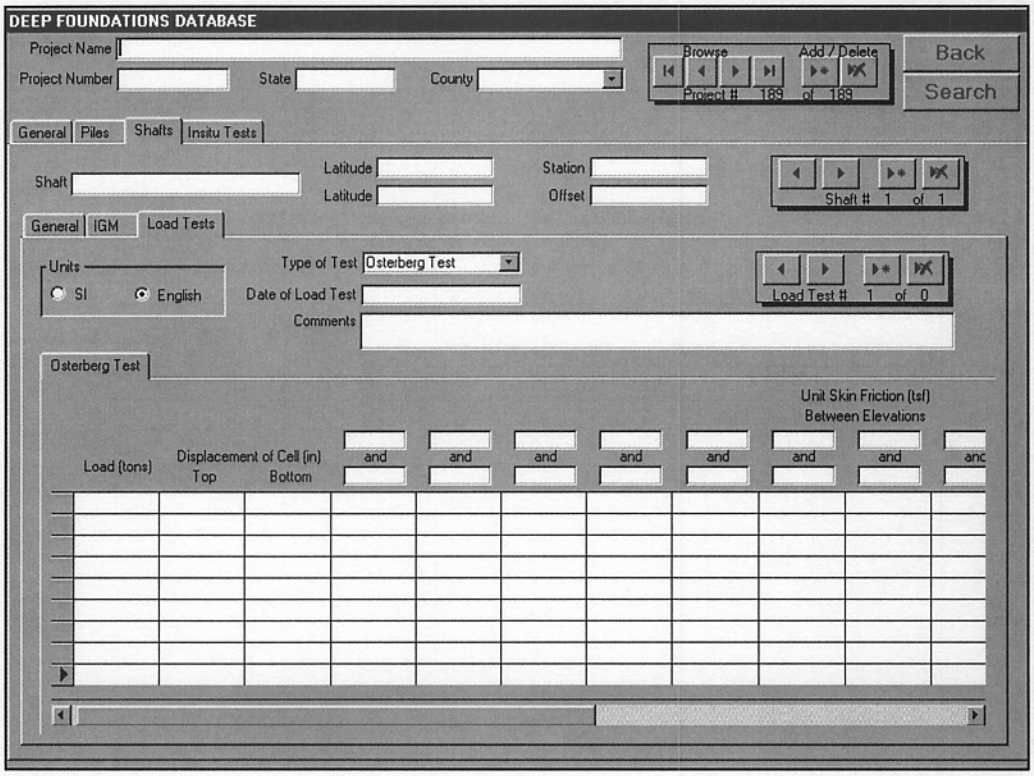


Figure 2.23 Add/Edit Form (Shafts, Osterberg Load Test)

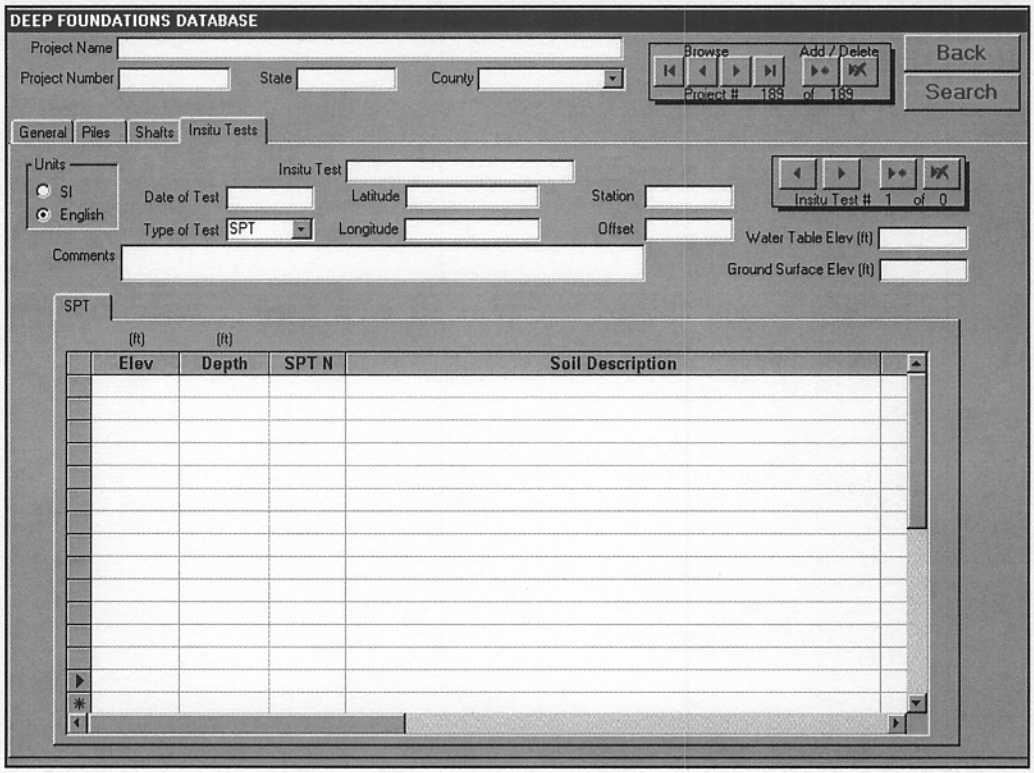


Figure 2.24 Add/Edit Form (Insitu Tests, SPT)

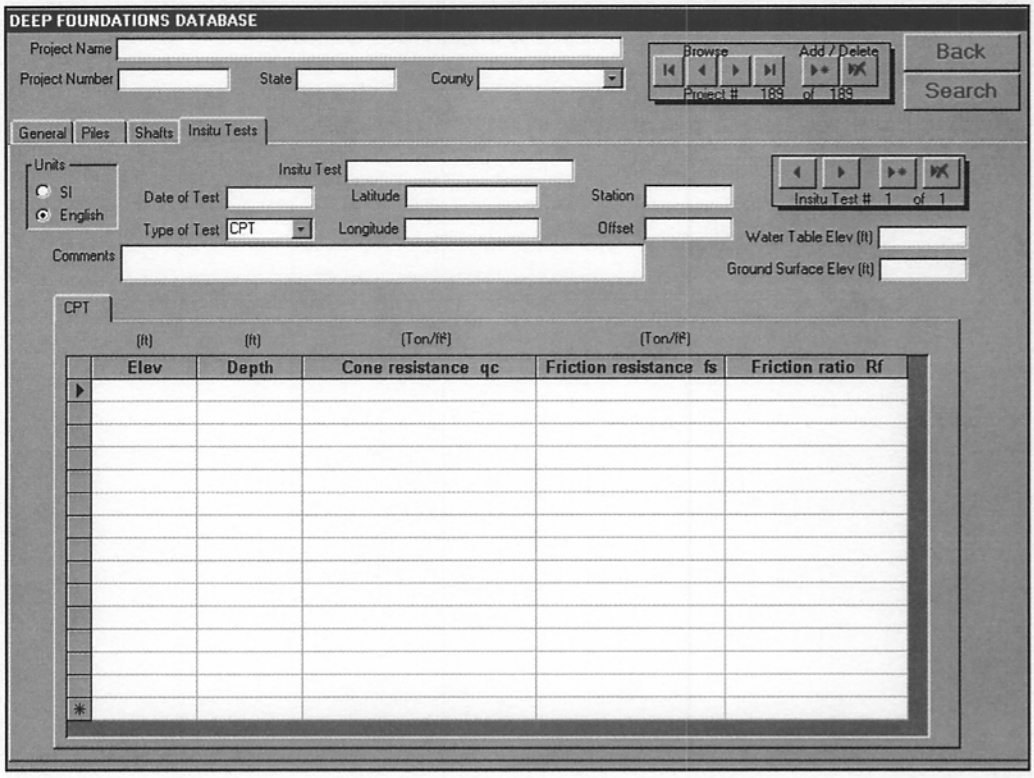


Figure 2.25 Add/Edit Form (Insitu Tests, CPT)

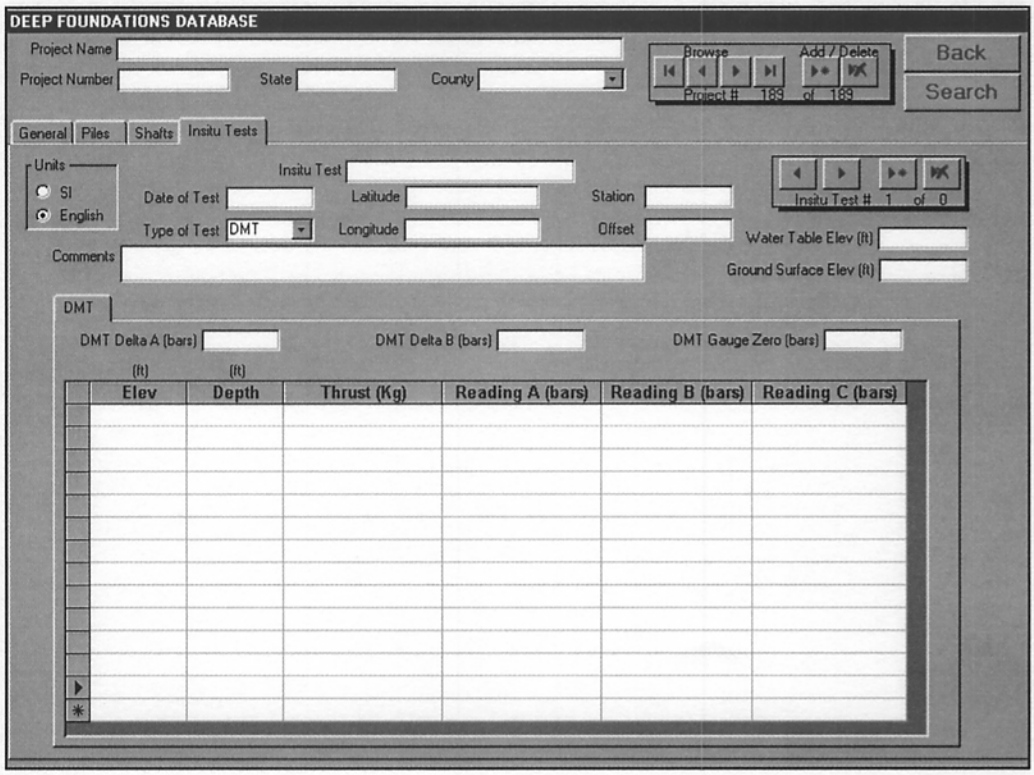


Figure 2.26 Add/Edit Form (Insitu Tests, DMT)

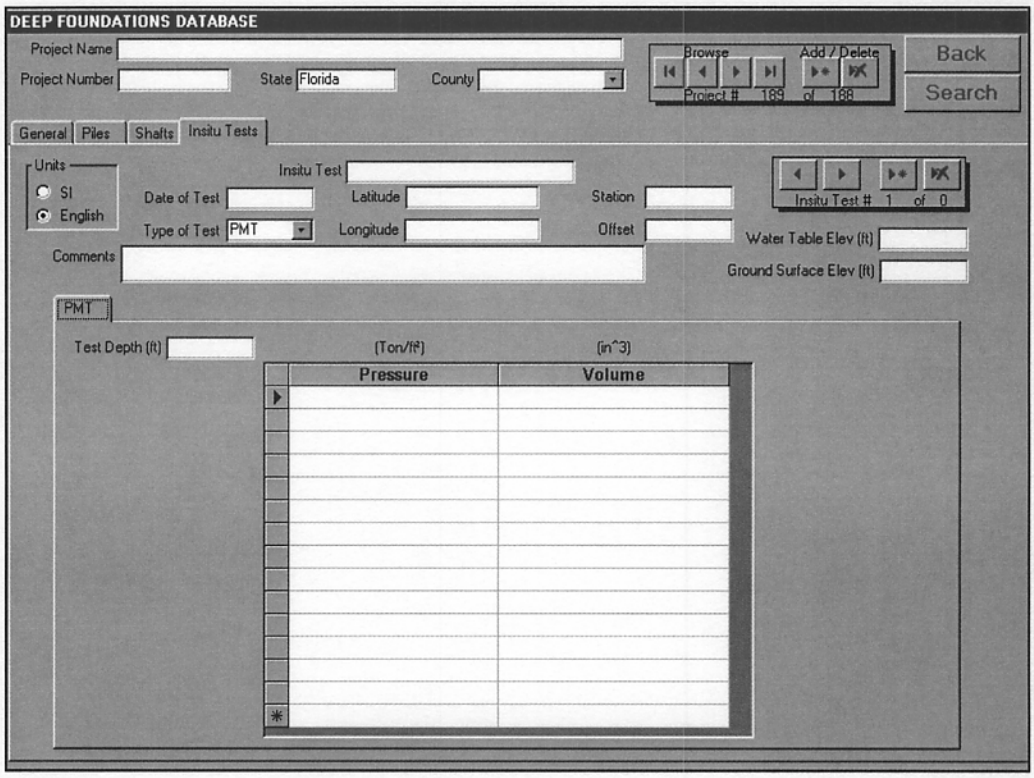


Figure 2.27 Add/Edit Form (Insitu Tests, PMT)

2.2.2 Data Display Forms

The forms used for adding and editing the data on the database tables are also used for displaying search results. Search operations are performed by means of select queries. Five queries are created each time the user performs a search. These are Query-General, Query-Insitu, Query-Load, Query-Piles and Query-Shafts. Eachof these queries serves as a base table for the data display forms. For example, the View-General form is based on the Query-General. These queries are temporary tables and are deleted and created anew each time the user performs a search operation. Consequently, the data have to be displayed in read only mode since changes made to the data on the Data Display forms would not be recorded on the data bearing tables. Navigation buttons had to be modified to comply with the read only view mode. The buttons used for adding new records and deleting existing records are no longer active and the number of records

matching the search criteria is displayed instead of the total number of records in the database. Command buttons for printing and editing the current record are also available. The example Data Display form in Figure 2.28 shows the third of twenty three SPT borings for the fourth of five projects identified in a search of project names which contained the word *orange*. Note the Add and Delete buttons are faded (not active).

2.2.3 Database Control Forms

The Menu form and the Search form are Database Control Forms. The Menu form identifies the database and displays the main menu while the Search form displays a set of options for the searching criteria. These forms are presented in Figures 2.29 and 2.30 respectively.

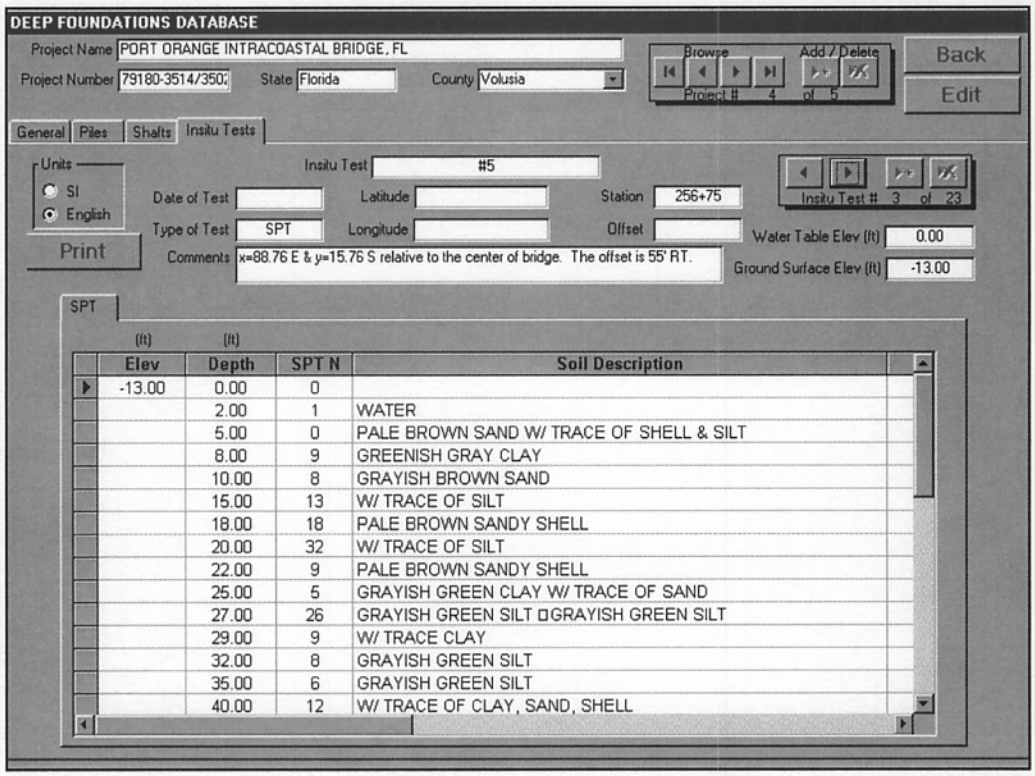


Figure 2.28 Data Display Form

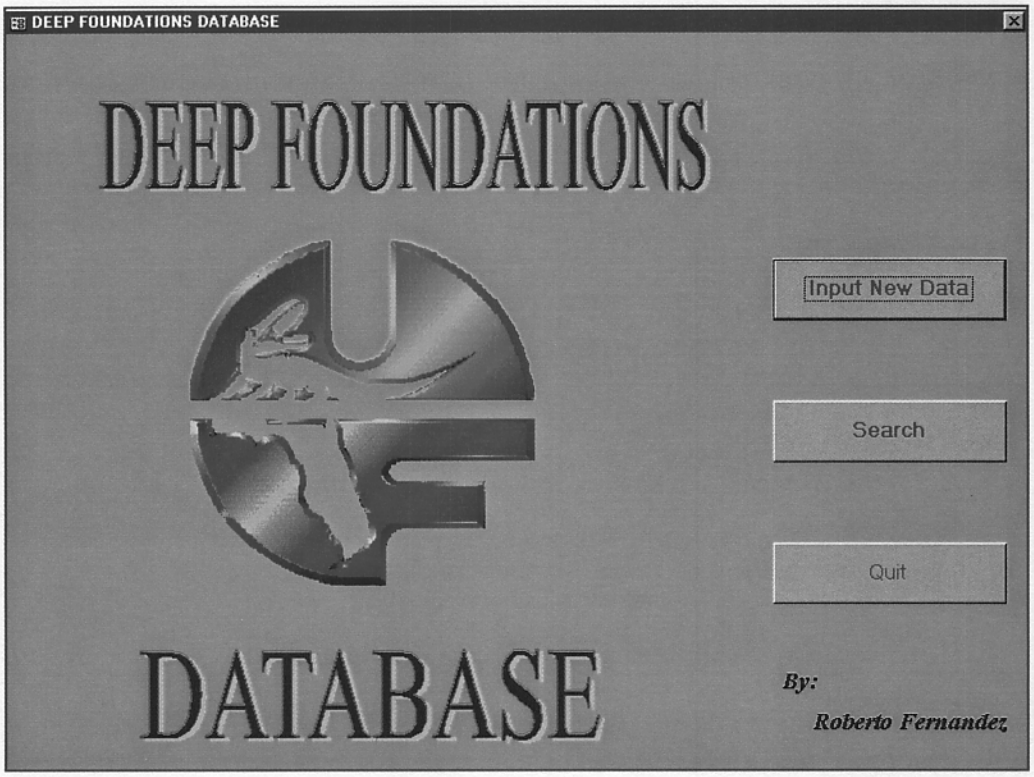


Figure 2.29 Menu Form

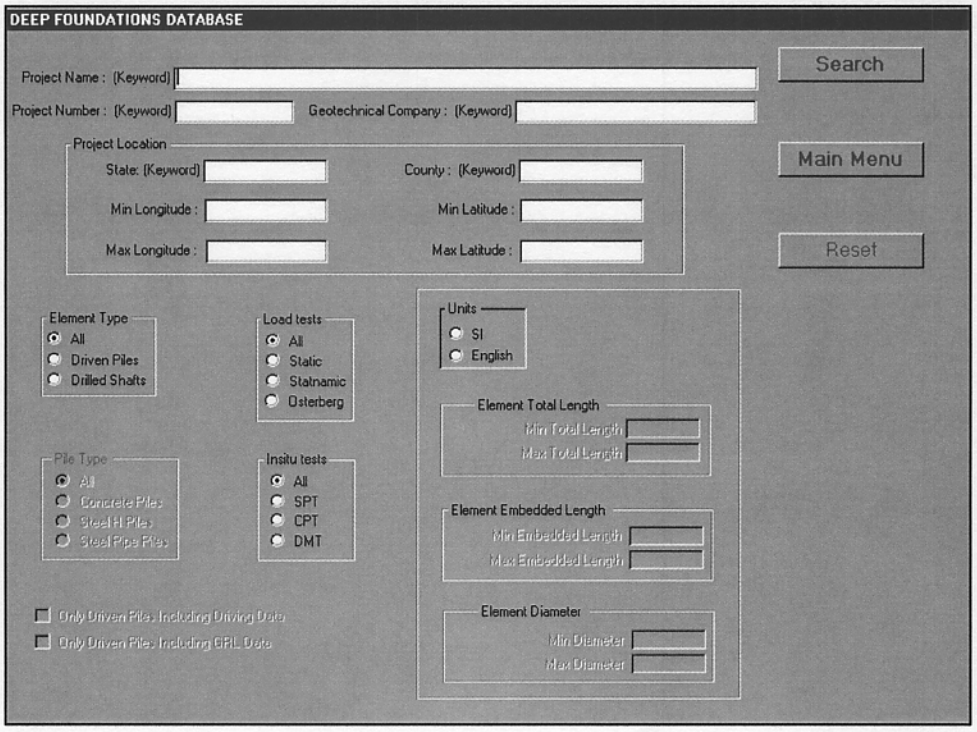


Figure 2.30 Search Form

2.3 Database Reports

The Deep Foundation Database uses Access reports to output the data; reports can be printed or exported to other applications. The data are divided into twelve sections, namely General Report, Pile Report, Shaft Report, Pile Penetration Report, GRL PDA Trace Report, Insitu SPT Report, Insitu CPT Report, Insitu DMT Report, Insitu PMT Report, Static Test Report, Statnamic Test Report and Osterberg Test Report. There are also a number of subreports, such as the Pile Concrete, Pile H Steel and Pile Pipe Steel Subreports. Most reports also have an English units version, e.g., Pile_English, which is used when the English Units option is selected. In total, there are forty-eight reports listed in the database.

2.4 Database Queries

The Deep Foundation Database uses select queries for all search operations. Select queries are tables that display only the records that meet specific criteria. Queries in Access can be created graphically using the design interface provided by the program (Query by example "QBE") or by including Structured Query Language "SQL" statements in Visual Basic Modules. All queries in this database were defined using SQL.

The search process involves five queries; Query-General, Query-Insitu, Query-Load, Query-Piles and Query-Shafts. Each time the user clicks the Search command button the existing queries are erased and new queries with the same names are created from SQL statements, based on the criteria specified by the user. Each Search query retrieves the information from all the fields in the associated table corresponding to the records that meet the specified criteria. The resulting queries function as source tables for the data displaying forms.

2.5 Database Units

The Deep Foundation Database allows the use of two different sets of units, English and SI. All the fields that involve units are stored twice in their corresponding tables, as entered through the forms (raw data, English or SI) and as converted to SI units. In addition, the table contains a field specifying the set of units used to input the data. The purpose of this is to have all the data stored in the same set of units in order to be able to query it, and still have the ability to display the data in their original format.

When data are entered through the specified form, they are automatically converted to SI units and stored in a different control on the form by means of a Visual Basic module. The input control is linked to the raw data field while the calculated control is linked to the SI field in the corresponding table. The calculated fields are not shown in form view.

The user has the ability to define the search criteria in either set of units. The criteria are converted to SI units and the search query selects the records based on the fields containing the data in SI units. However, the Data Display forms are linked to the fields containing the raw data to present the results in their original format.

In order to provide the user with the ability to output the data in English units, additional reports are created based on the same queries but with calculated controls to convert the data from SI to English units. The name of the existing report followed by the word "English" is given to each of the new reports (e.g., the Insitu SPT_English Report).

The units and conversion factors used in the database are presented below:

	English	To	SI	Multiply by
Piles				
Pile Elastic Modulus	psi		kN/m^2	6.8948
Capwap Elastic Modulus	psi		kN/m^2	6.8948
Pile wave speed	ft/s		m/s	0.3048
Pile Impedance	lb-s/ft		kN-s/m	0.01459
Concrete Piles				
Diameter	in		mm	25.40
Void Diameter	in		mm	25.40
Cross Sectional Area	in^2		mm^2	645.16
Total Length	ft		m	0.3048
Embedded Length	ft		m	0.3048
Total Weight	tons		kN	8.8964
Unit Weight of Concrete	pcf		kN/m^3	0.1571
Concrete Strength	psi		kN/m^2	6.8948
Steel H Piles				
Steel Cross Sectional Area	in^2		mm^2	645.16
Pile Cross Sectional Area	in^2		mm^2	645.16
Steel Section Perimeter	in		mm	25.40
Pile Perimeter	in		mm	25.40
Weight of Steel	tons/ft		kN/m	29.1878
Weight of pile	tons		kN	8.8964
Total Length	ft		m	0.3048
Embedded Length	ft		m	0.3048
Unit Weight of Steel	pcf		kN/m^3	0.1571
Concrete Strength	psi		kN/m^2	6.8948
Steel Pipe Piles				
Outside Diameter	in		mm	25.40
Thickness of Steel	in		mm	25.40
Steel Cross Sectional Area	in^2		mm^2	645.16
Pile Cross Sectional Area	in^2		mm^2	645.16
Weight of Steel	tons/ft		kN/m	29.1878
Weight of pile	tons		kN	8.8964
Total Length	ft		m	0.3048

	English	To	SI	Multiply by
Steel Pipe Piles (cont.)				
Embedded Length	ft		m	0.3048
Unit Weight of Steel	pcf		kN/m^3	0.1571
Concrete Strength	psi		kN/m^2	6.8948
Driving Data				
Weight of Hammer	lb		kN	4.4482
Energy	lb-ft		kN-m	0.001355
Pre-bored Depth	ft		m	0.3048
Penetration (Last Blow)	in		mm	25.40
Penetration (End of Driving)	in		mm	25.40
Penetration (Start of Restriking)	in		mm	25.40
Penetration Log				
Depth	ft		m	0.3048
GRL Data				
L	ft		m	0.3048
Le	ft		m	0.3048
Area	ft^2		m^2	0.0929
C	ft/s		m/s	0.3048
Ec	$tons/ft^2$		GN/m^2	0.00009576
EA/c	tons-s/ft		kN-s/m	29.1878
Drilled Shafts				
Diameter	in		mm	25.40
Total Length	ft		m	0.3048
Embedded Length	ft		m	0.3048
Casing Length	ft		m	0.3048
Bell Diameter	in		mm	25.40
Bell Length	ft		m	0.3048

	English	To	SI	Multiply by
IGM				
Slump of Concrete	in		mm	25.40
f'_c	tons/ft^2		kN/m^2	95.7605
E'_c	tons/ft^2		kN/m^2	95.7605
Elev	ft		m	0.3048
Depth	ft		m	0.3048
q_U	tons/ft^2		kN/m^2	95.7605
q_T	tons/ft^2		kN/m^2	95.7605
E_M	tons/ft^2		kN/m^2	95.7605
Ei	tons/ft^2		kN/m^2	95.7605
Insitu tests				
Water Table Elevation	ft		m	0.3048
Ground Surface elevation	ft		m	0.3048
SPT				
Elevation	ft		m	0.3048
Depth	ft		m	0.3048
CPT				
Elevation	ft		m	0.3048
Depth	ft		m	0.3048
Qc	tons/ft^2		kN/m^2	95.7605
Fs	tons/ft^2		kN/m^2	95.7605
DMT				
Elevation	ft		m	0.3048
Depth	ft		m	0.3048
PMT				
Pressure	tons/ft^2		kN/m^2	95.7605
Volume	in^3		mm^3	16387.06

	English	To	SI	Multiply by
Static Load Test				
Force	tons		kN	8.8964
Displacement	in		mm	25.40
Statnamic Load Test				
Max Force	tons		kN	8.8964
Max Displacement	in		mm	25.40
Permanent Displacement	in		mm	25.40
Displacement	in		mm	25.40
Velocity	ft/s		m/s	0.3048
Acceleration	ft/s^2		m/s^2	0.3048
Osterberg Load Test				
Load	tons		kN	8.8964
Displacement	in		mm	25.40
Unit Skin friction	$tons/ft^2$		kN/m^2	95.7605

CHAPTER 3

USING THE DATABASE

3.1 Opening the Database

Start Microsoft Access. If the Access startup box appears, select "More Files..." on the Open an Existing Database section and click "OK"; otherwise, click on the "File" menu and choose "Open Database...". When the open database dialog box appears, find and select the database file Geotech.mdb and click "Open" to continue. Since the Deep Foundation Database is contained in a single file, it can be stored in any convenient location. The database can also be loaded directly by double clicking on its icon when using the Windows Explorer.

Once loaded, the database Menu form will appear as shown in Figure 3.1.

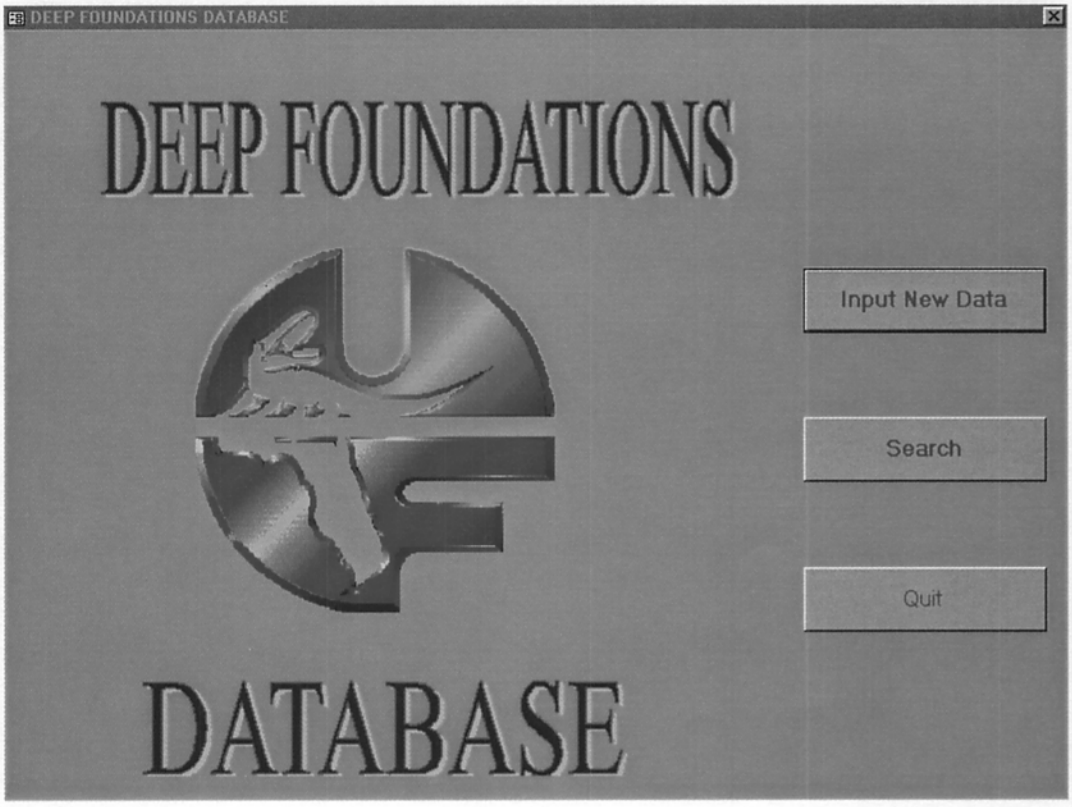
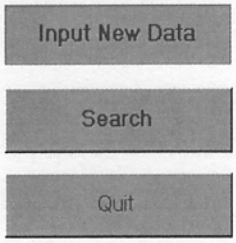


Figure 3.1 Menu Form

The Menu form contains three buttons providing the following options:



Input New Data. Opens the Add/Edit form and starts a new record.

Search. Opens the Search form.

Quit. Closes the database and terminates Access.

3.2 Creating a New Record

Click on the "Input New Data" button on the Menu form. The General Add/Edit form, as shown in Figure 3.2, will appear.

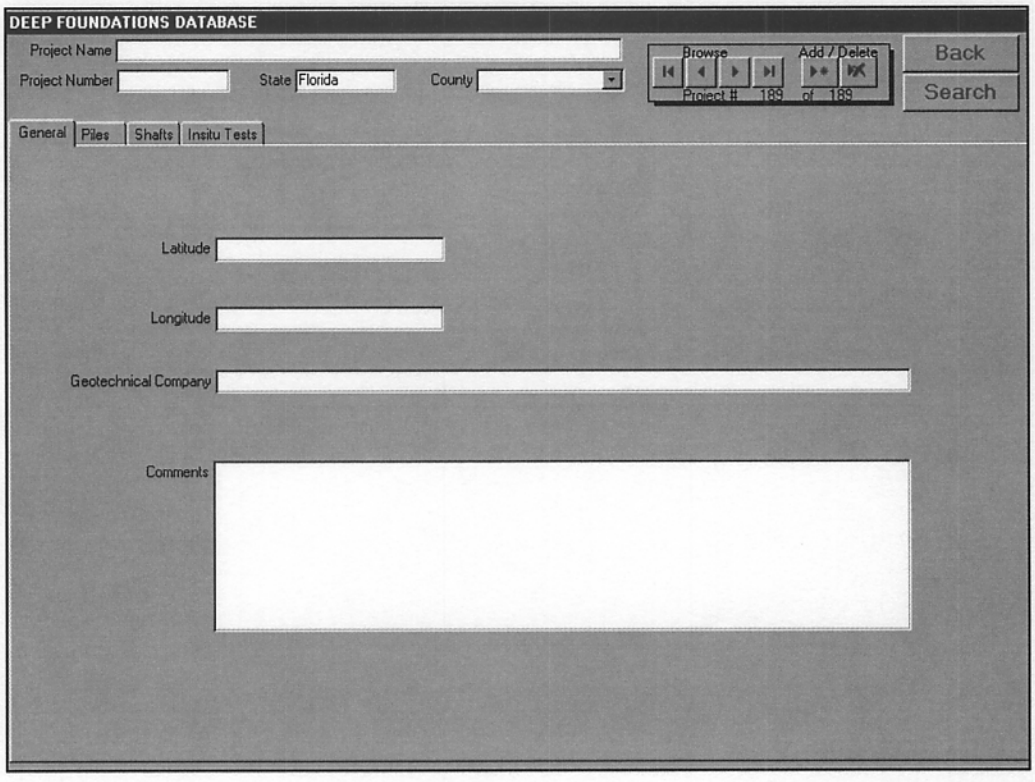


Figure 3.2 Add/Edit Form (General)

This form defines the project. It includes the project name, number, location (state, county, latitude and longitude), Geotechnical Company which wrote the project report and a space for comments. Tabs then permit the addition of pile, shaft and/or insitu test data. Data can be entered in any order and are automatically saved. If the value of a field is changed it is automatically updated.

When a section is accessed, all the control labels display the name of the fields without units. Once the user selects the desired set of units, the labels will display the name of the field with the specified units. Figure 3.3 shows the Concrete Pile add/edit form before units are selected. Figures 3.4 and 2.11 show the form after SI and English units, respectively, have been chosen. Units cannot be changed. If the user attempts to change the units the data in that tab section will be erased and will have to be reentered. Similarly, if the user attempts to change the pile type (Concrete, H-pile or Pipe), the insitu test type (SPT, CPT, DMT or PMT) or the load test type (Static, Statnamic or Osterberg) in their respective sections, all the data in that section will be erased and will have to be reentered.

Multiple pile, shaft and/or insitu test data can be available for a single project and multiple GRL records and/or load tests can exist for a particular pile or shaft. A record navigation box is located at the upper right corner of each such section as well as in the main form. Figure 3.5 shows an example from the database. The upper navigation box shows that this White City Bridge project is the 178th project in the database. The middle navigation box indicates that the third of nine pile records is current. The lower navigation box shows that the first of two load test records (actually a single load test with displacements measured in two different ways) is current.

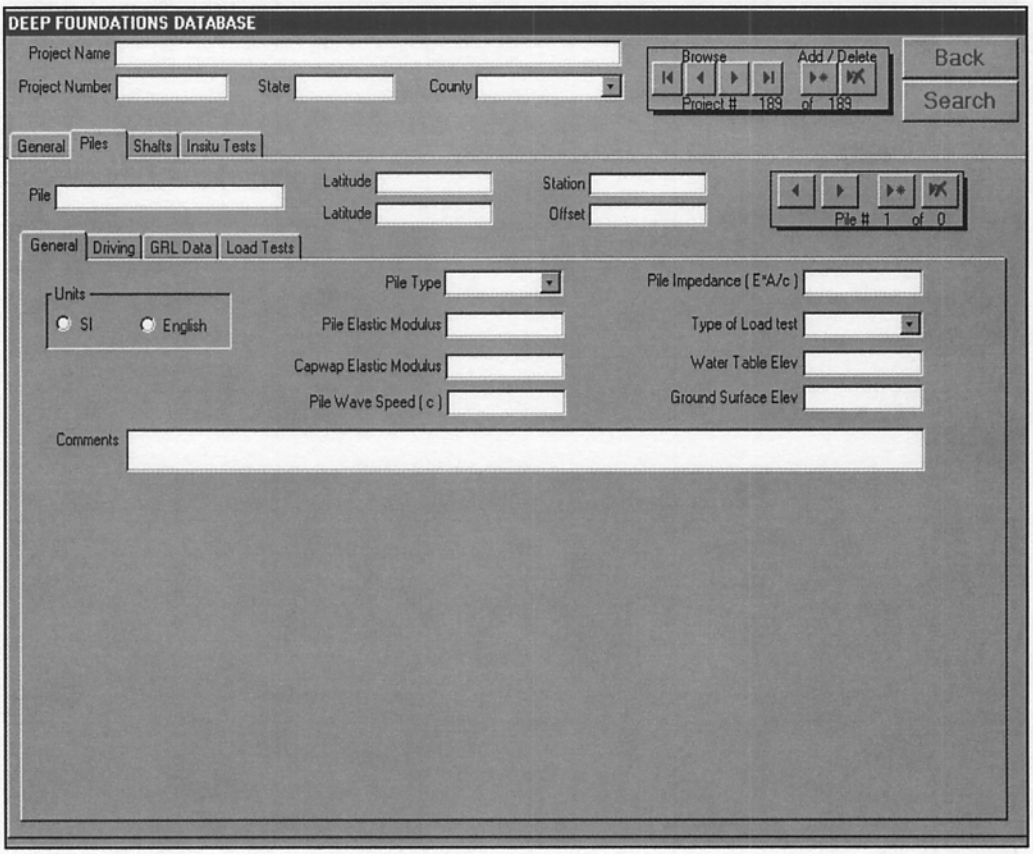


Figure 3.3 Add/Edit Form (Piles, Concrete Piles) No Units

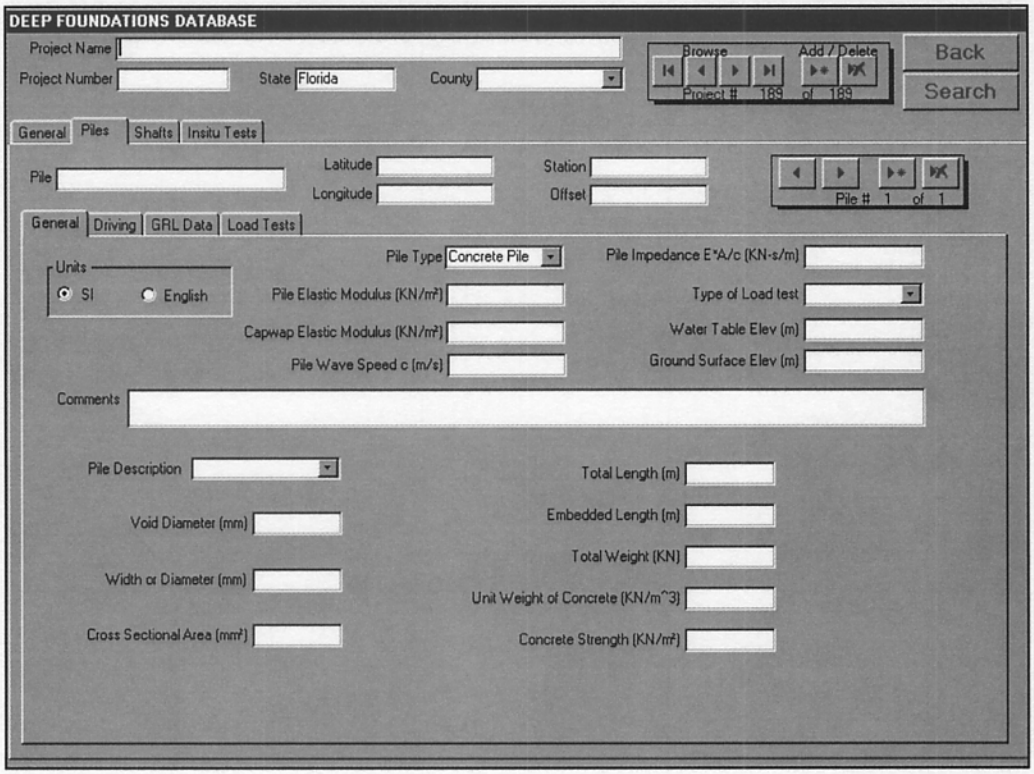


Figure 3.4 Add/Edit Form (Piles, Concrete Piles) SI Units

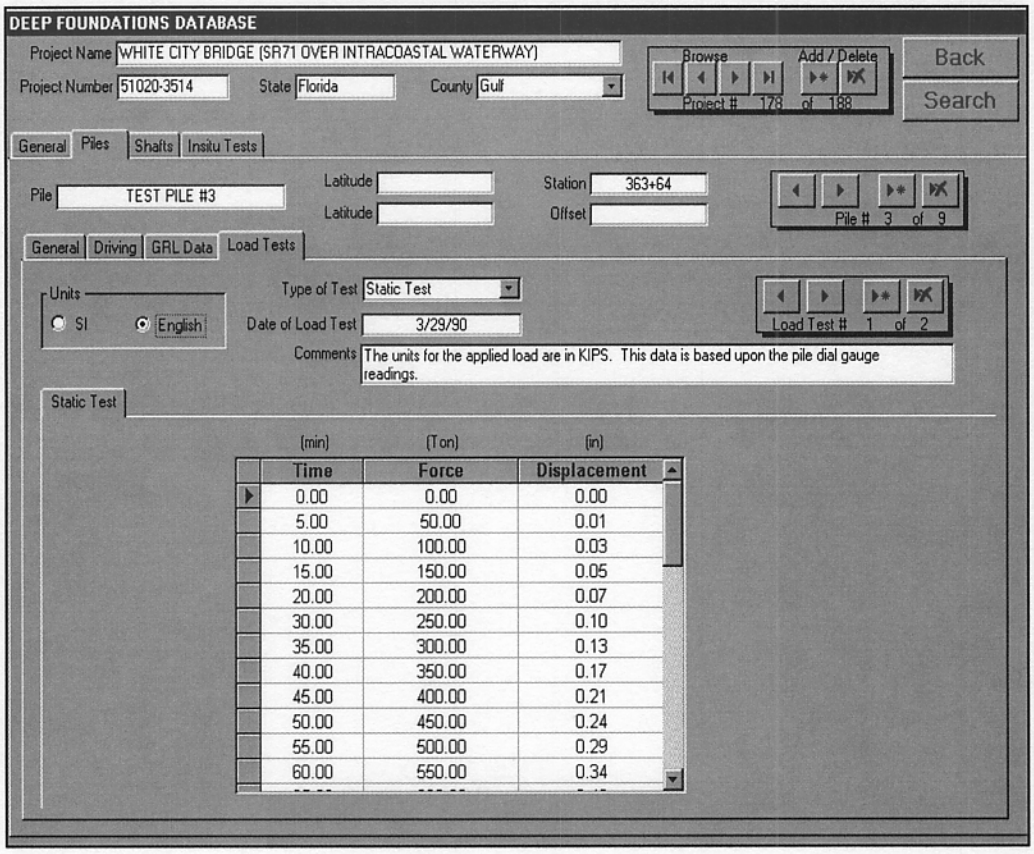


Figure 3.5 An Example of Multiple Record Navigation Boxes

The functions of the controls in the navigation box are explained in Figure 3.6.

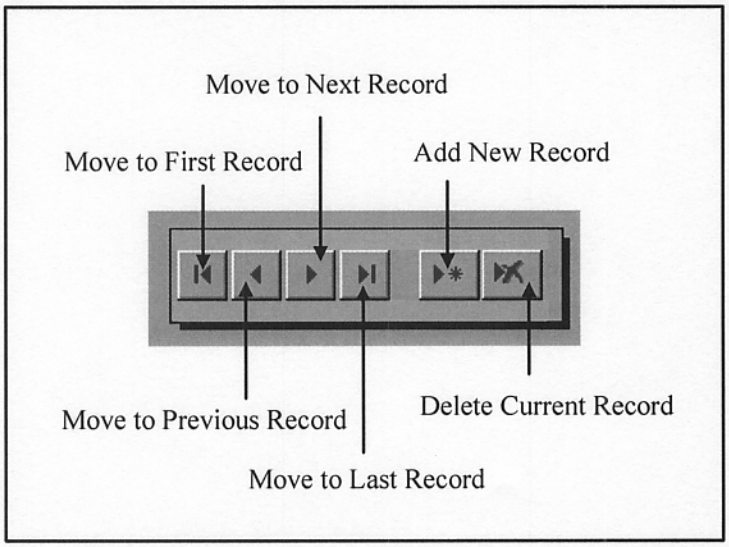


Figure 3.6 Record Navigation Box

From the opening General Add/Edit form data inputting is straightforward; highlight a tab, choose units, use drop-down boxes where available and type in data. The seventeen Add/Edit forms are included as Figures 3.7 through 3.23. These are identical to Figures 2.11 through 2.27 except that in this chapter, the SI rather than English versions have been printed. When data input is complete click "Back" to return to the menu form.

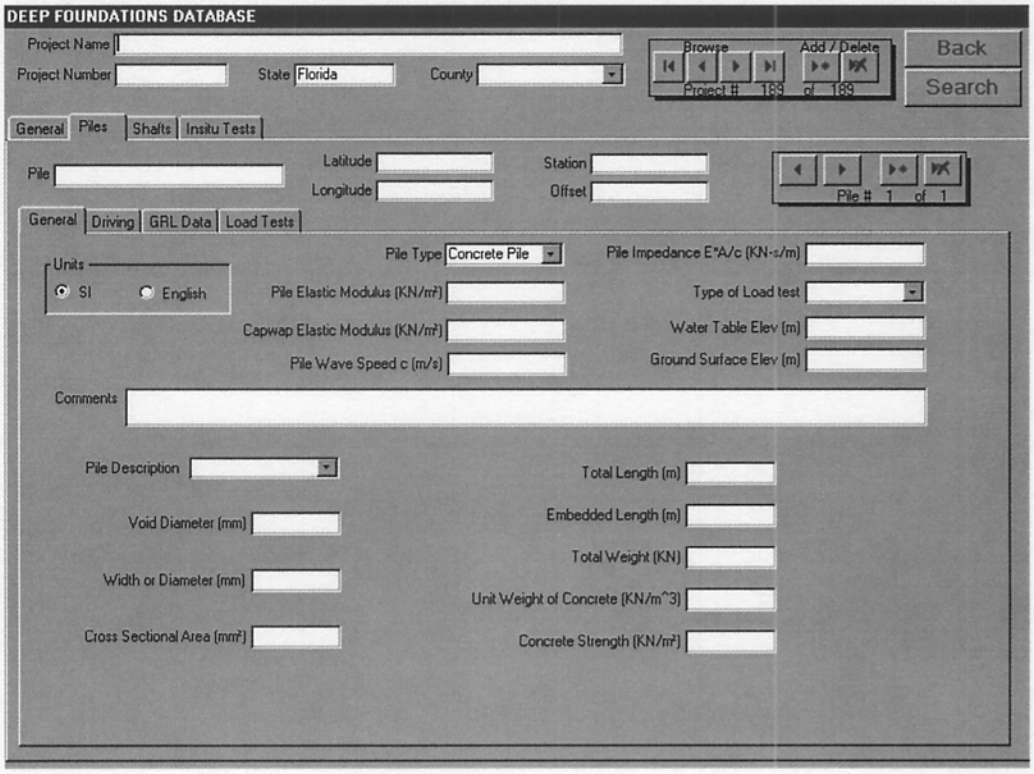


Figure 3.7 Add/Edit Form (Piles, Concrete Piles)

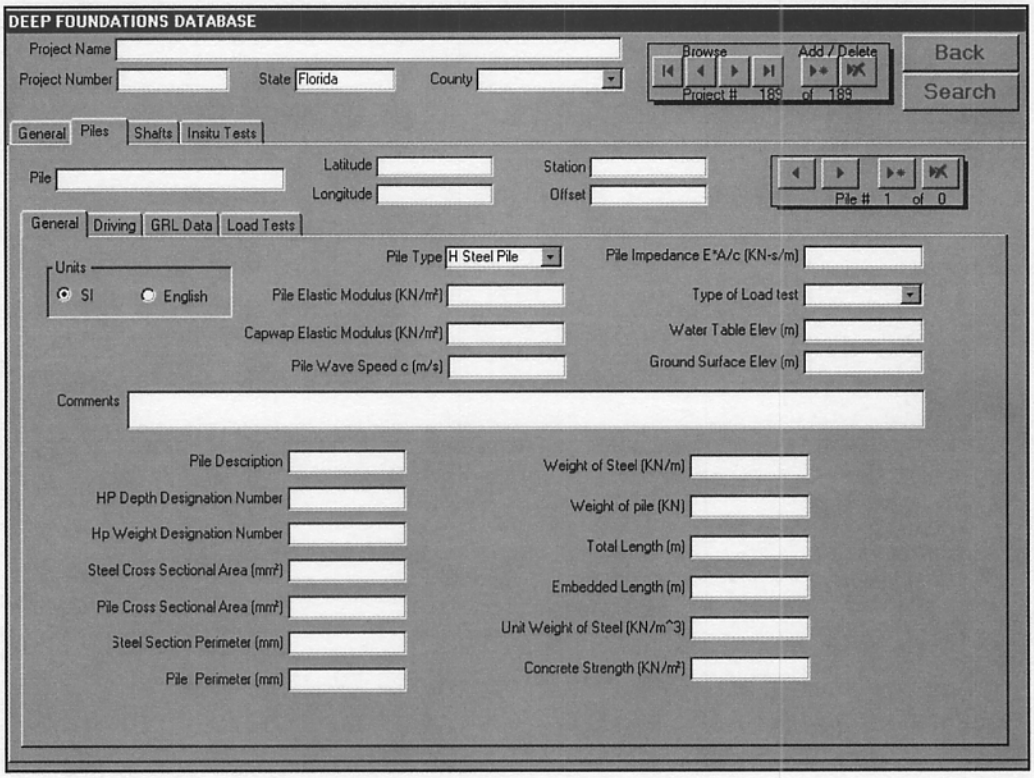


Figure 3.8 Add/Edit Form (Piles, Steel H-Piles)

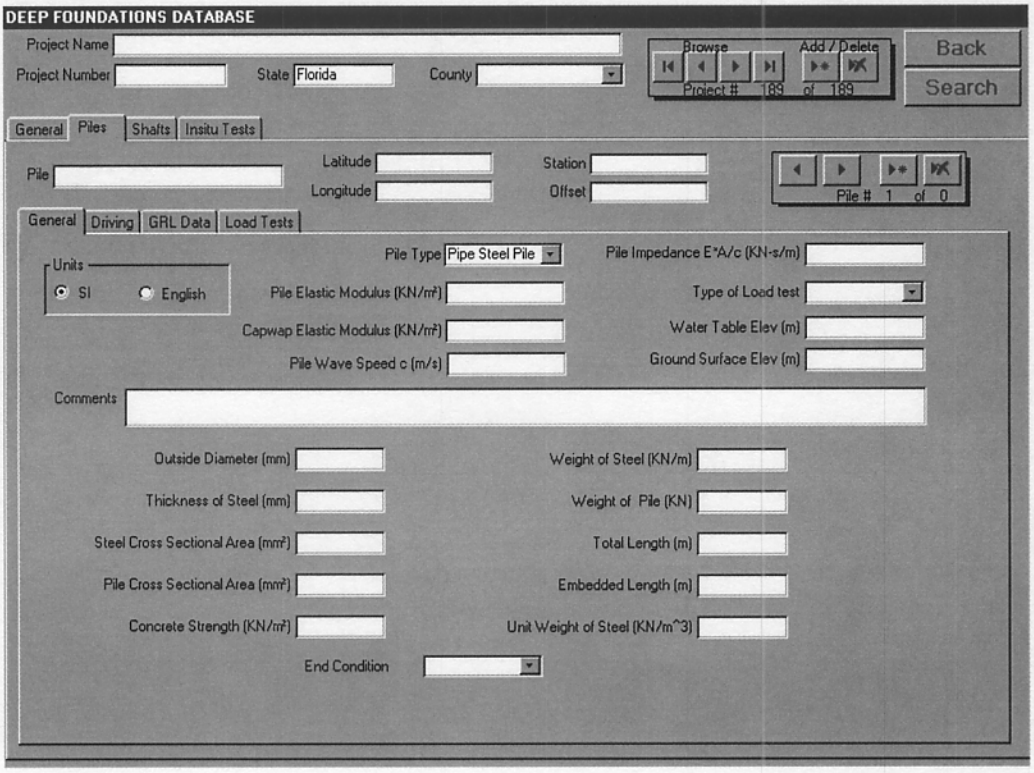


Figure 3.9 Add/Edit Form (Piles, Steel Pipe Piles)

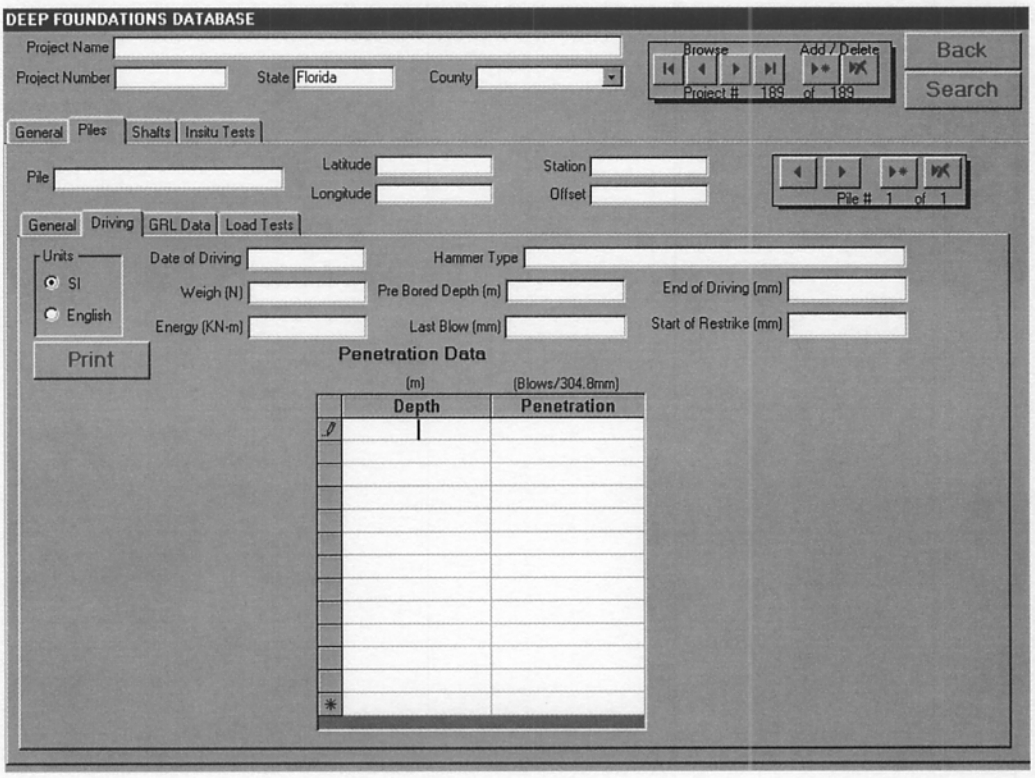


Figure 3.10 Add/Edit Form (Piles, Pile Driving Data)

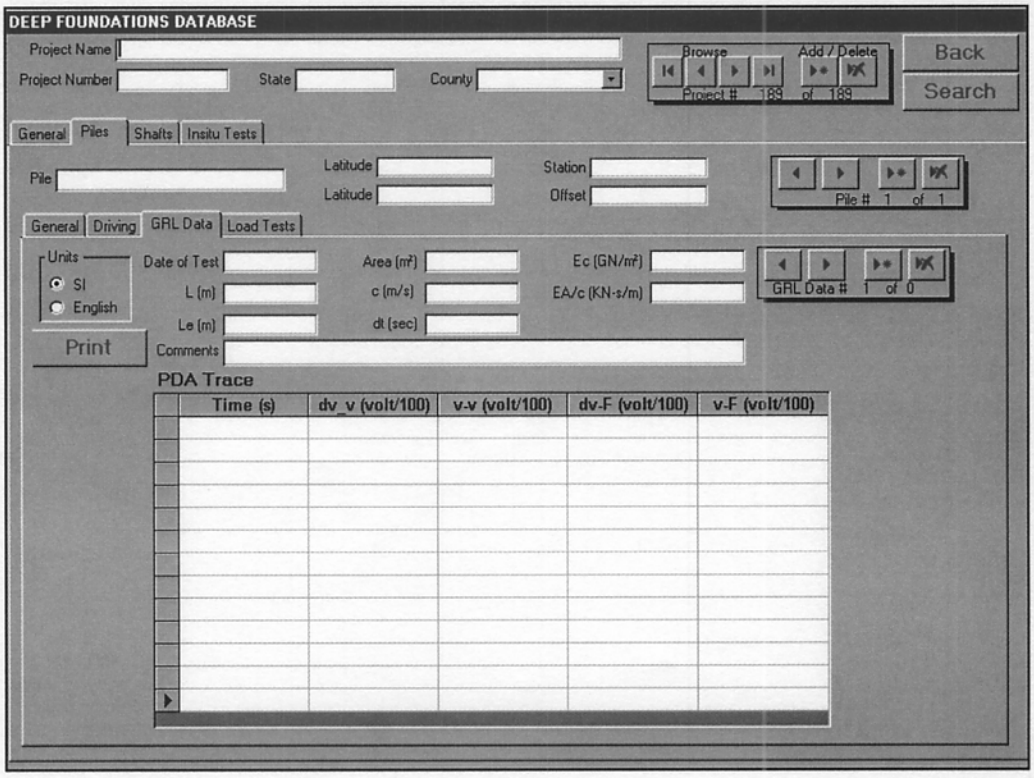


Figure 3.11 Add/Edit Form (Piles, GRL Data)

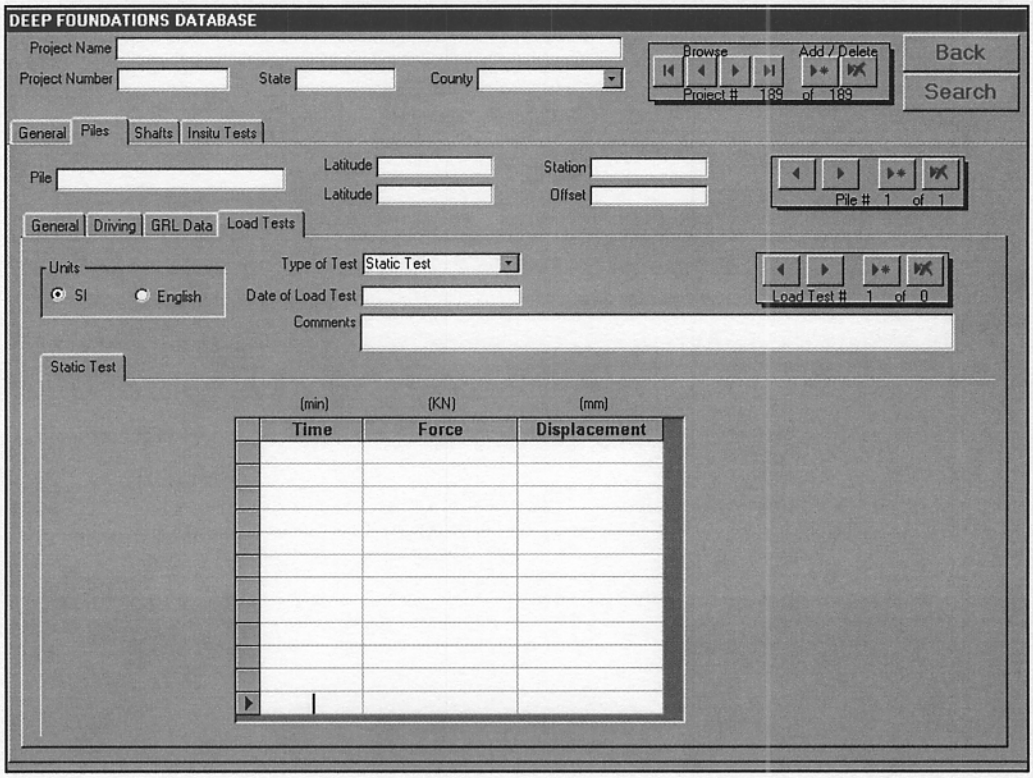


Figure 3.12 Add/Edit Form (Piles, Static Load Test)

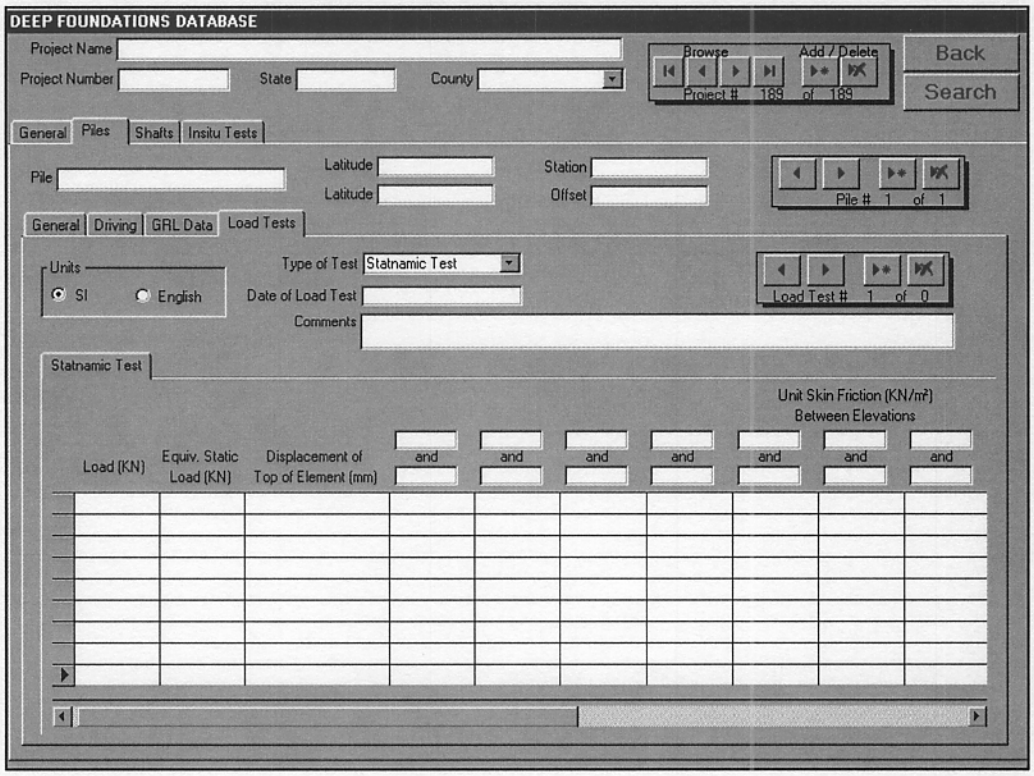


Figure 3.13 Add/Edit Form (Piles, Statnamic Load Test)

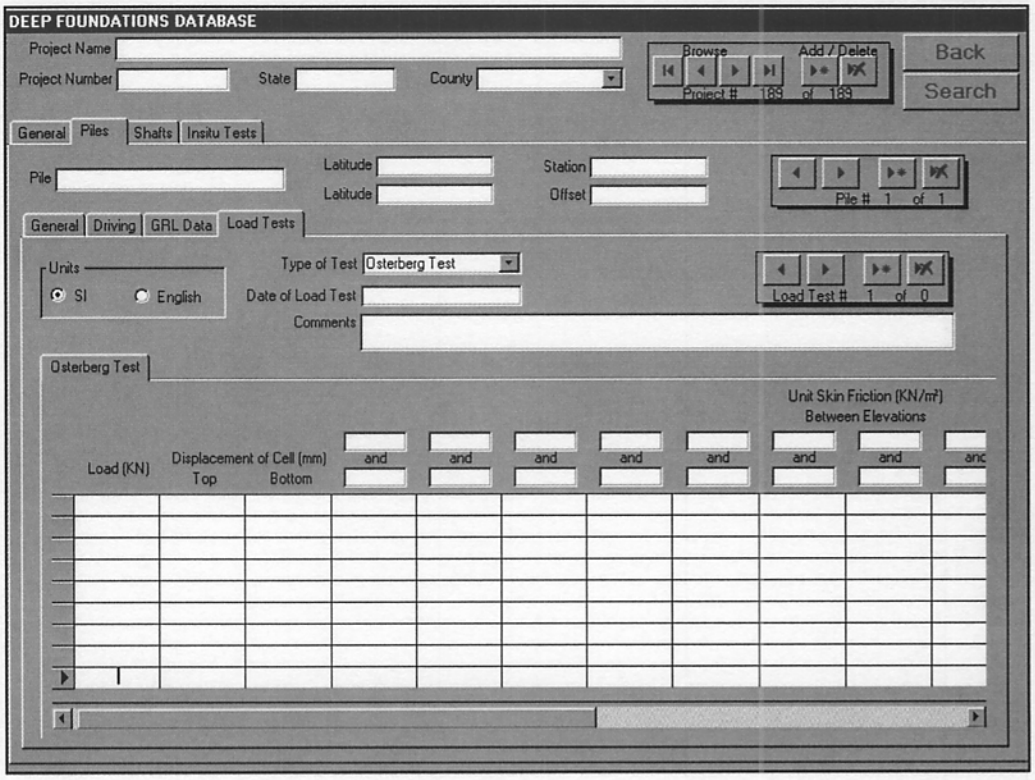


Figure 3.14 Add/Edit Form (Piles, Osterberg Load Test)

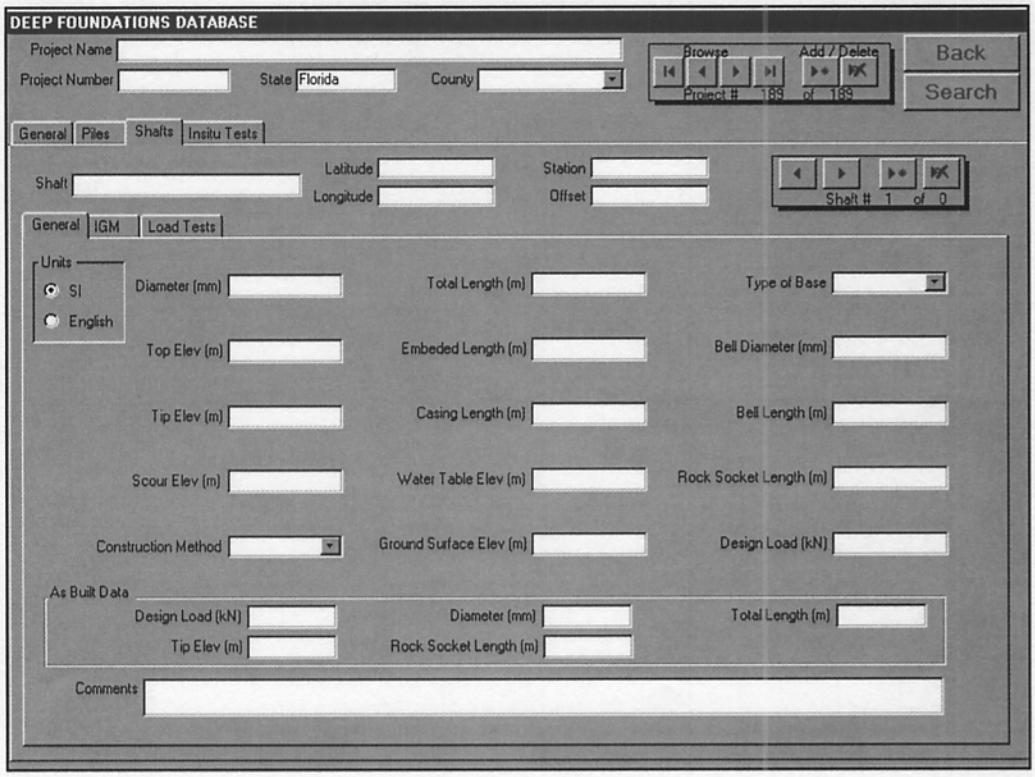


Figure 3.15 Add/Edit Form (Shafts)

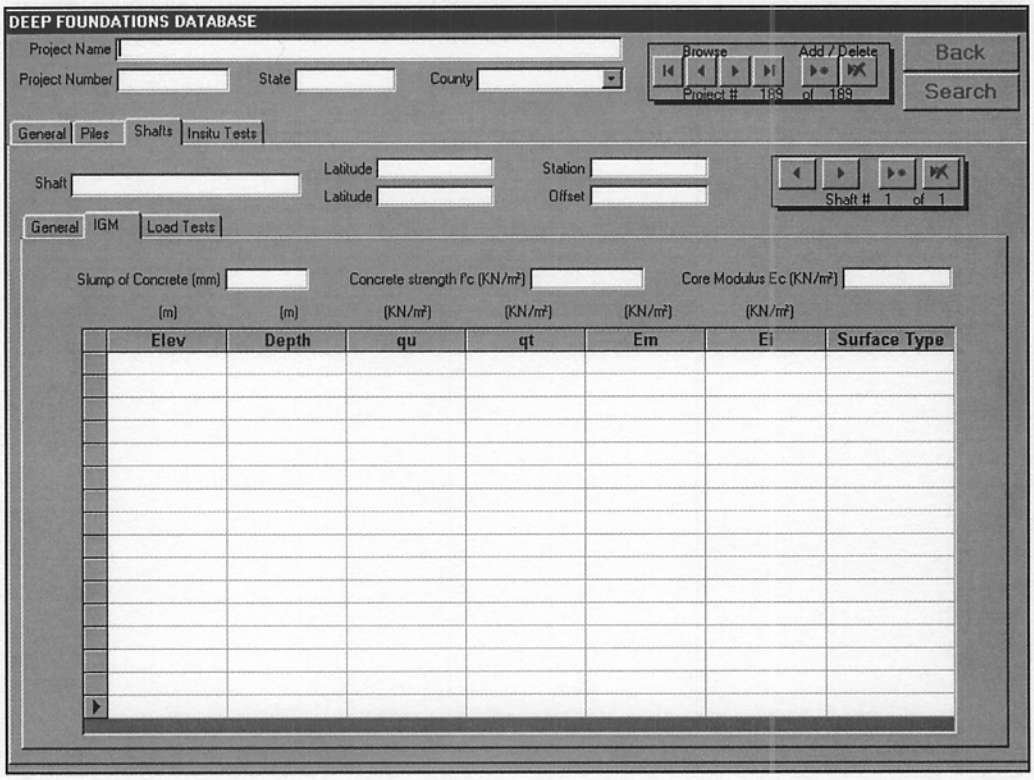


Figure 3.16 Add/Edit Form (Shafts, IGM)

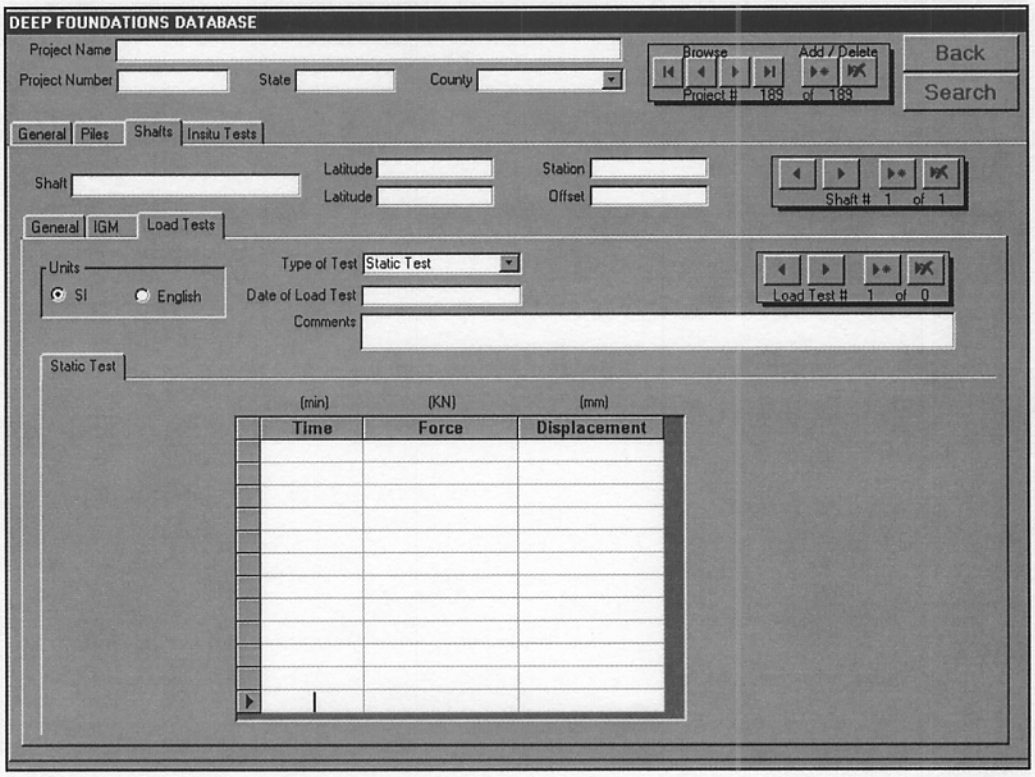


Figure 3.17 Add/Edit Form (Shafts, Static Load Test)

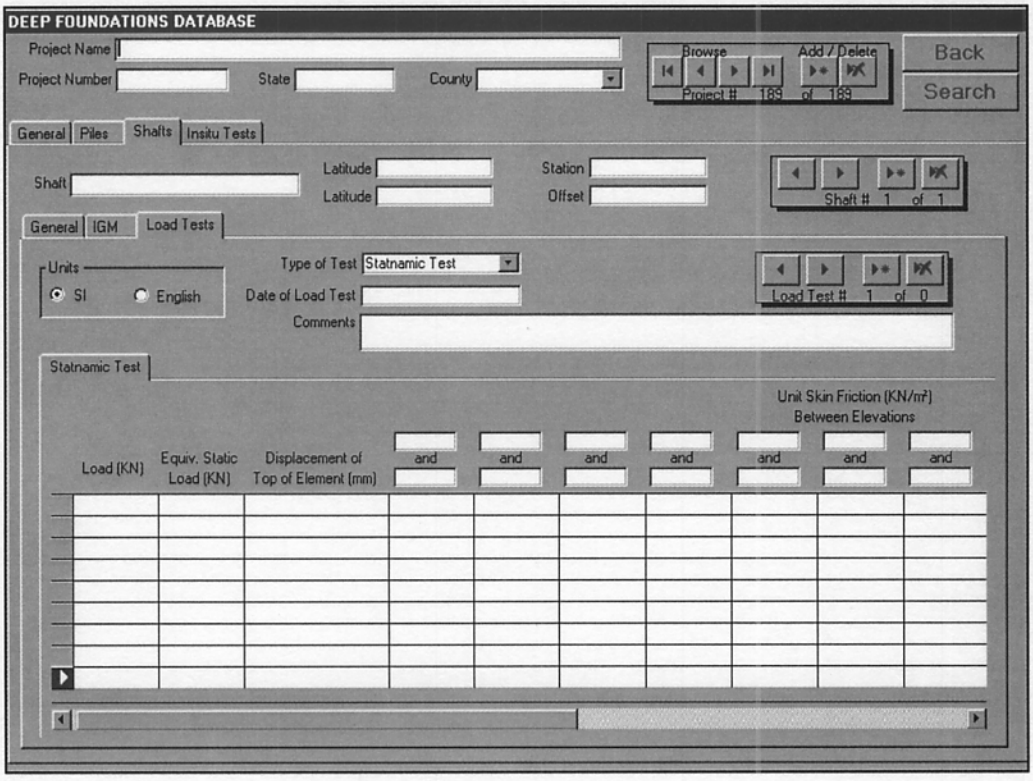


Figure 3.18 Add/Edit Form (Shafts, Statnamic Load Test)

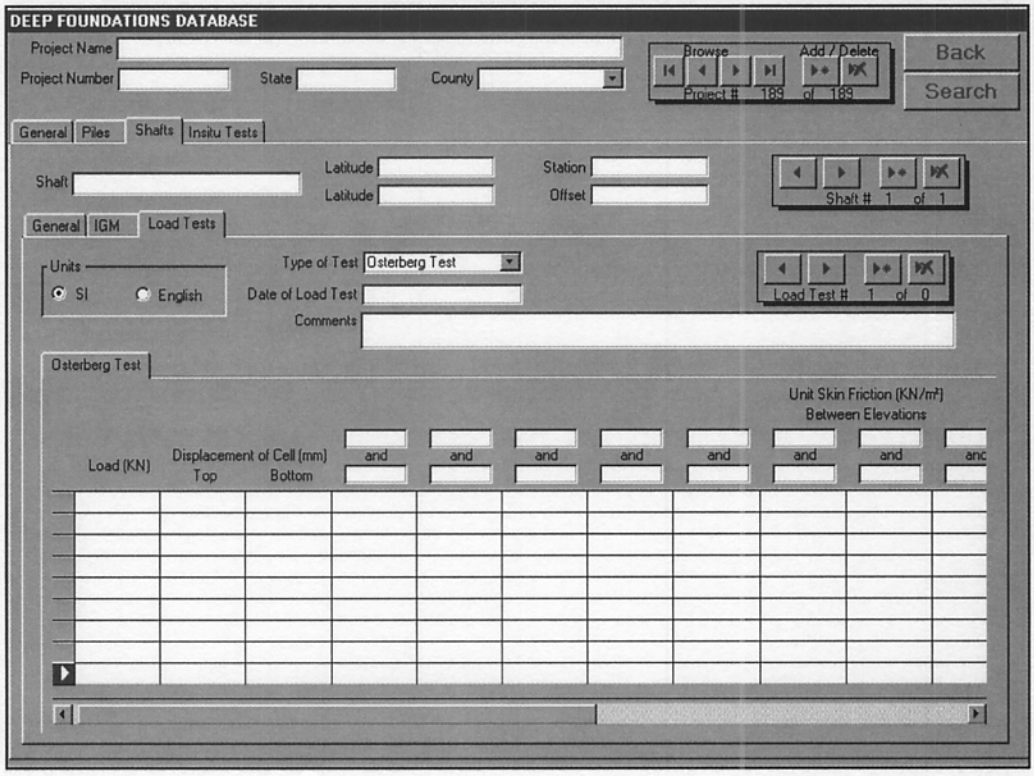


Figure 3.19 Add/Edit Form (Shafts, Osterberg Load Test)

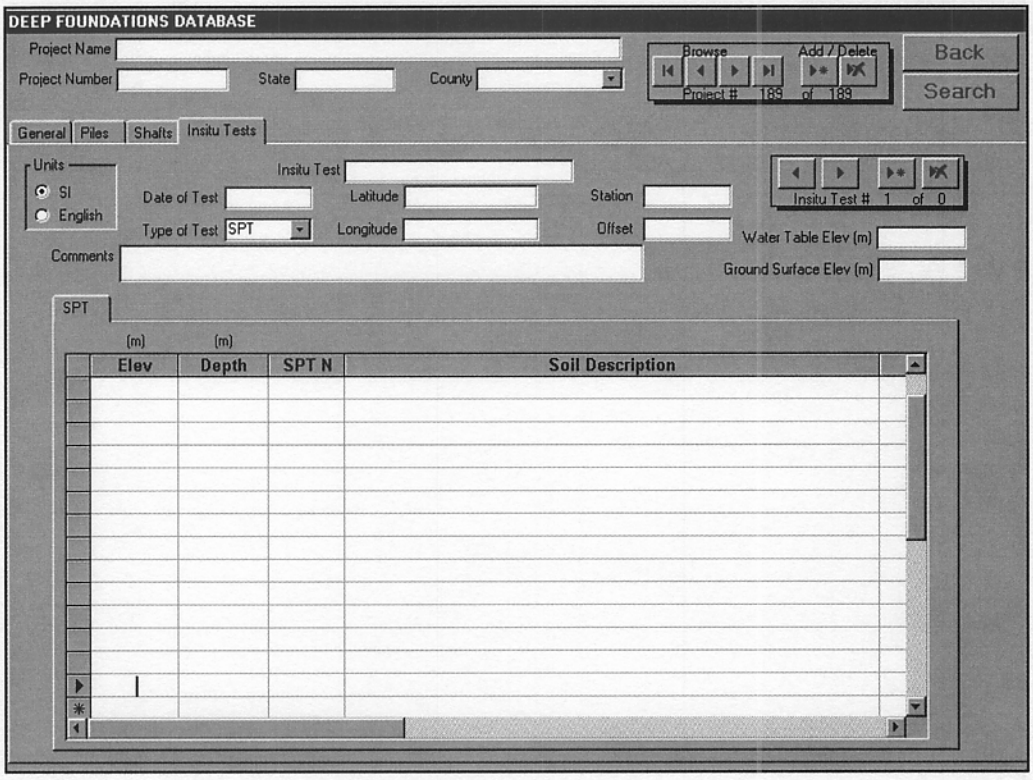


Figure 3.20 Add/Edit Form (Insitu Tests, SPT)

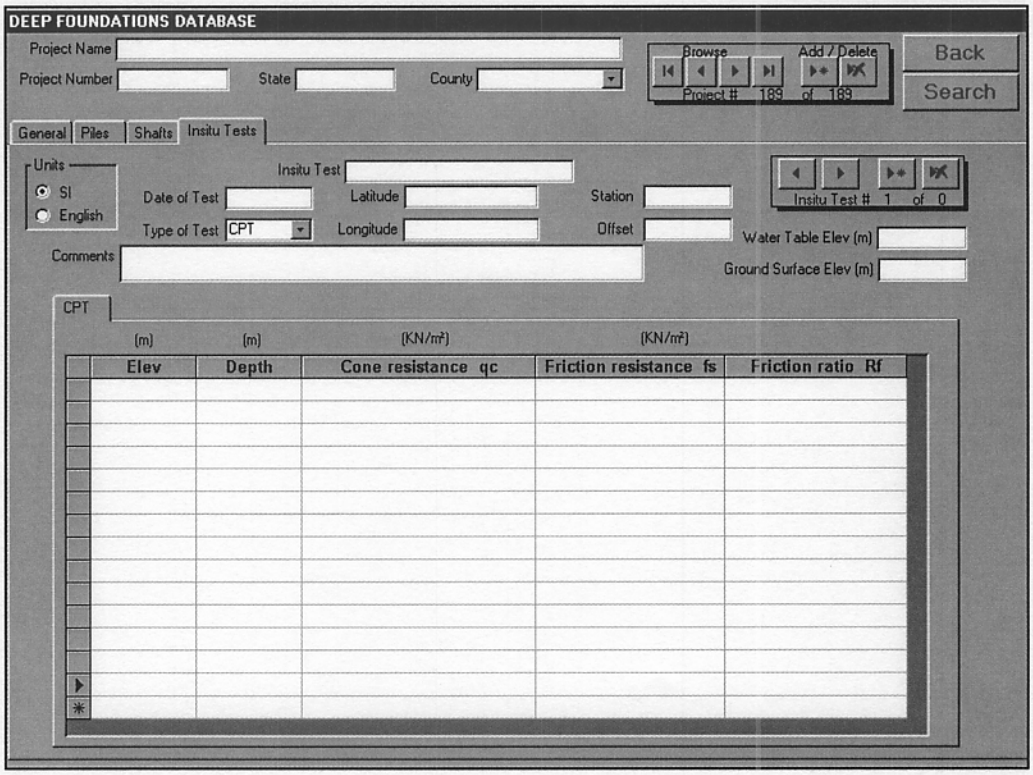


Figure 3.21 Add/Edit Form (Insitu Tests, CPT)

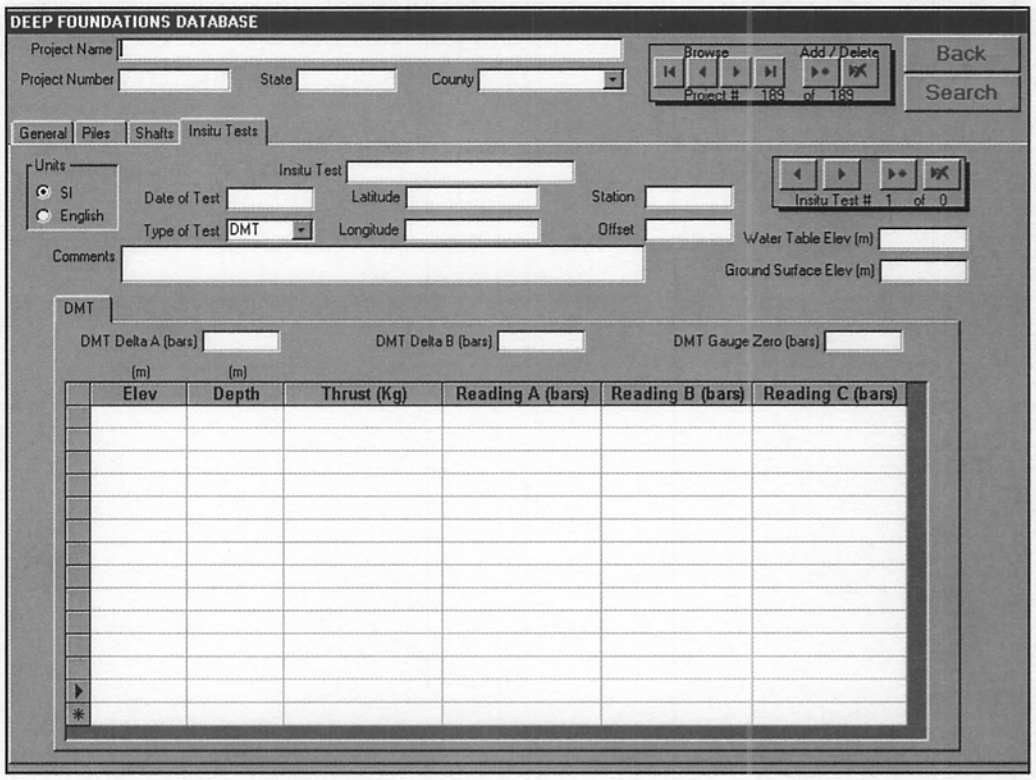


Figure 3.22 Add/Edit (Insitu Tests, DMT)

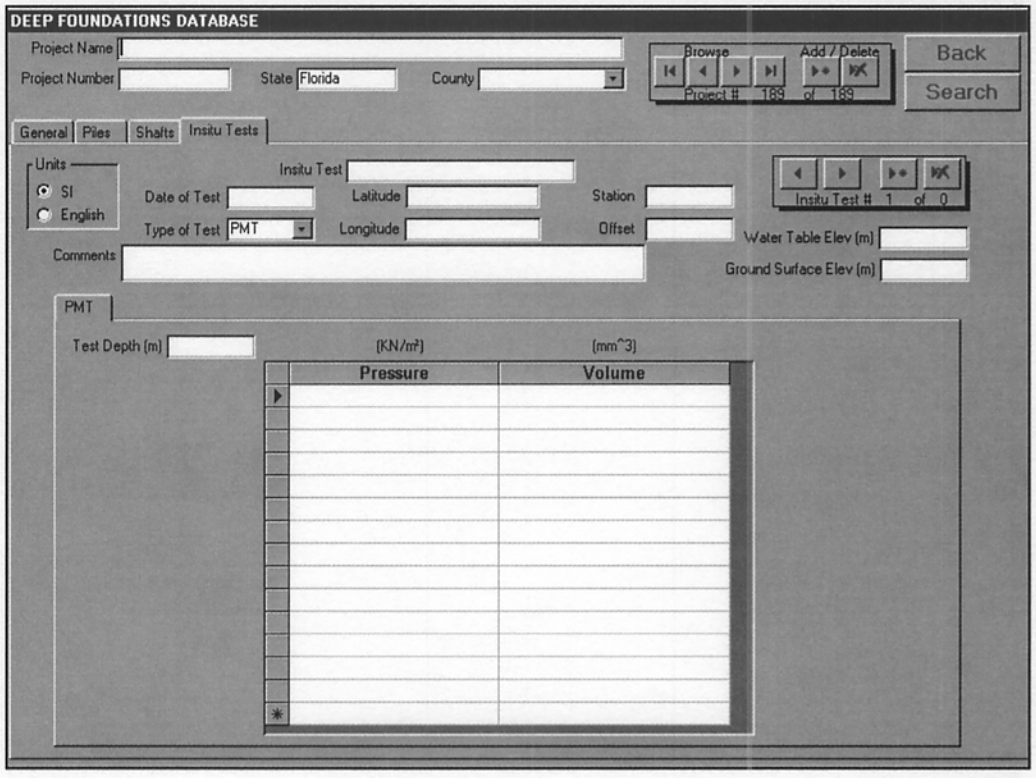


Figure 3.23 Add/Edit (Insitu Tests, PMT)

3.3 Searching and Displaying Records

Click on the "Search" button on the Menu form or on any of the Add/Edit forms. The Search form, Figure 3.24, will appear. If the "Search" button on this form is clicked without making any selections or typing any criteria the search will include all the records in the database. Select the desired criteria and click "Search" to narrow the search to only those records matching the specified criteria. The results from the search are shown by means of the Data Display form. Examples are included as Figures 3.25 and 3.26. Figure 3.25 shows general information and Figure 3.26 data from a static load test, for the third of four piles for the fifth of five projects identified in a search of project names which contained the word *orange*. As noted in Section 2.2.2 the "Add" and "Delete" buttons are not active on this form.

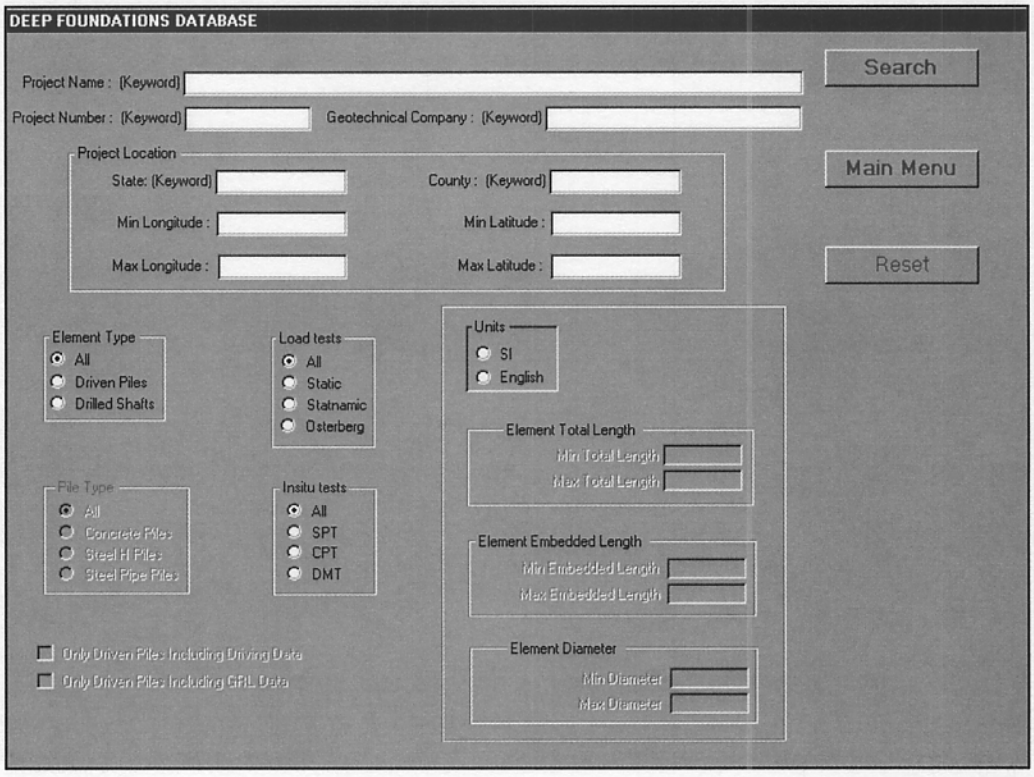


Figure 3.24 Search Form

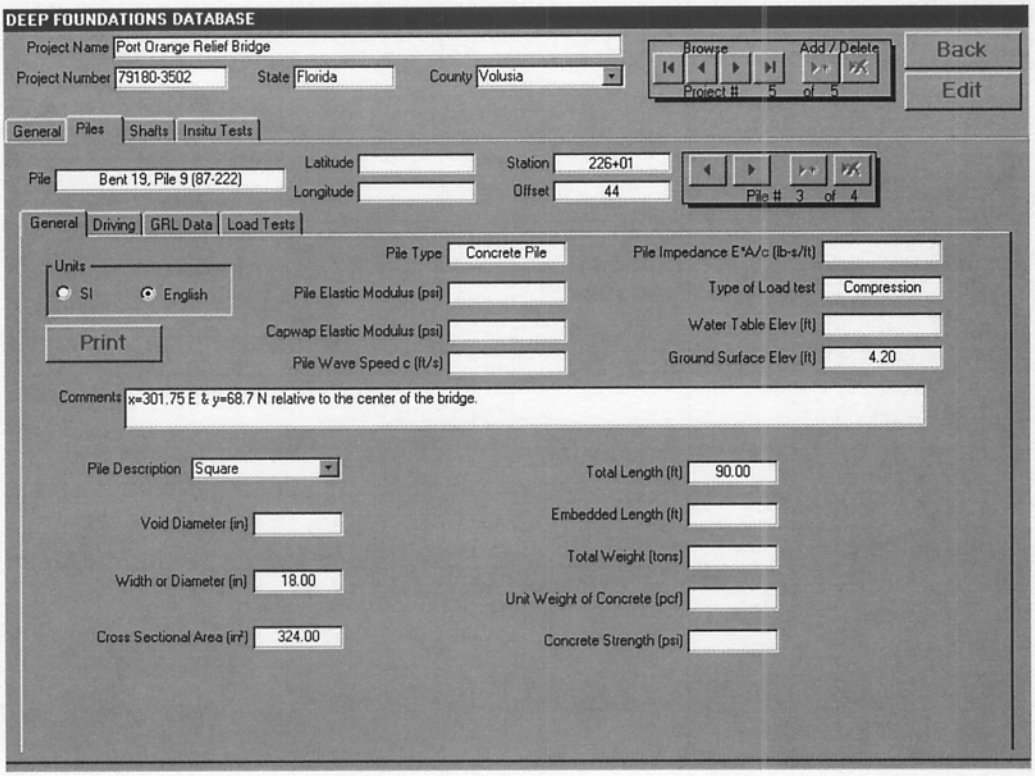


Figure 3.25 Data Display Form

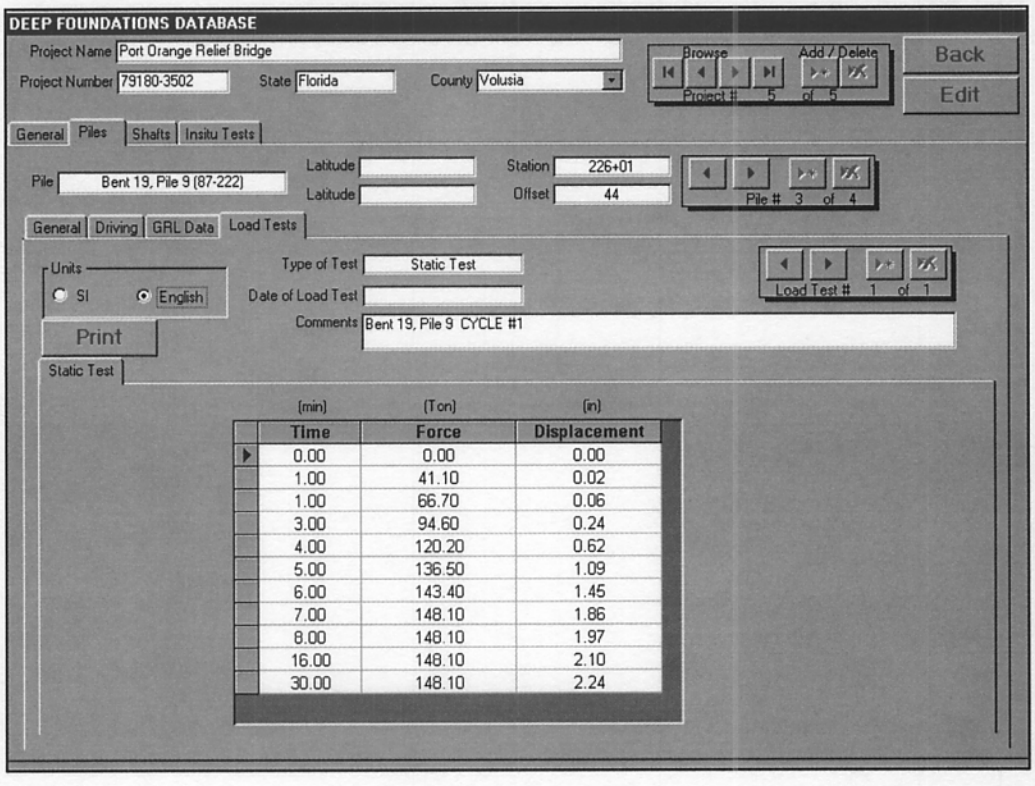


Figure 3.26 Data Display Form

Three command buttons are available on the Data Display form.

The "Back" button:

Closes the Display Data form and returns to the Search form. The user can modify the previous search criteria to perform a new search. Alternatively, the user can click "Reset" to delete the specified criteria and then click "Search" to display all the records or specify new criteria.

The "Edit" button:

Opens the Add/Edit form for the current record. The user is allowed to edit or erase the current record.

Details are provided in Section 3.4.

The "Print" button:

Opens the Print form to allow output of the data related to the current record.

Details are provided in Section 3.5.

3.4 Editing Records.

When displaying the record in the Data Display form, click on the "Edit" button. The Add/Edit form will appear as shown in Figure 3.27. This form only allows editing or deleting the current record. In the record navigation box the project is labeled Project 1 of 1, and only the "Delete" button is active. Click "Back" to close the Add/Edit form and return to the Display Data form.

Data can also be edited by clicking "Input New Data" and locating the records to be edited using the record navigation boxes on the Add/Edit forms. This technique will allow the user to edit several records without leaving the Add/Edit form; however, it is much more difficult to find the records.

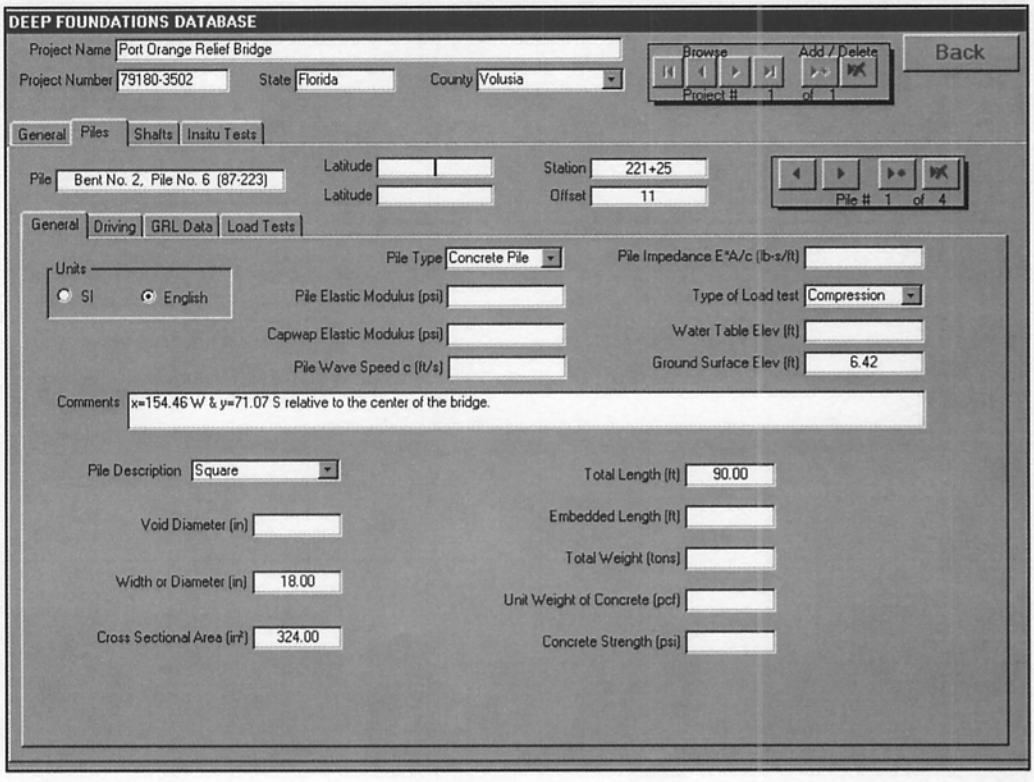


Figure 3.27 Editing on the Add/Edit Form

3.5 Printing Records

When displaying a specific record in the Data Display form, click the "Print" command button. The Print form will appear superimposed as shown in Figure 3.28. Reports can be printed in either SI or English units. In addition, each report can be printed, exported to a Microsoft Excel workbook or exported to a Microsoft Word document. Select the units and the output option and click "OK".

If either "Create Excel Workbook" or "Create Word Document" is selected, the user is prompted to assign a name and a location for each new file created. If the print option is selected, the reports are automatically printed when the "OK" button is clicked. Figure 3.29 shows the report printed if the OK of Figure 3.28 where clicked.

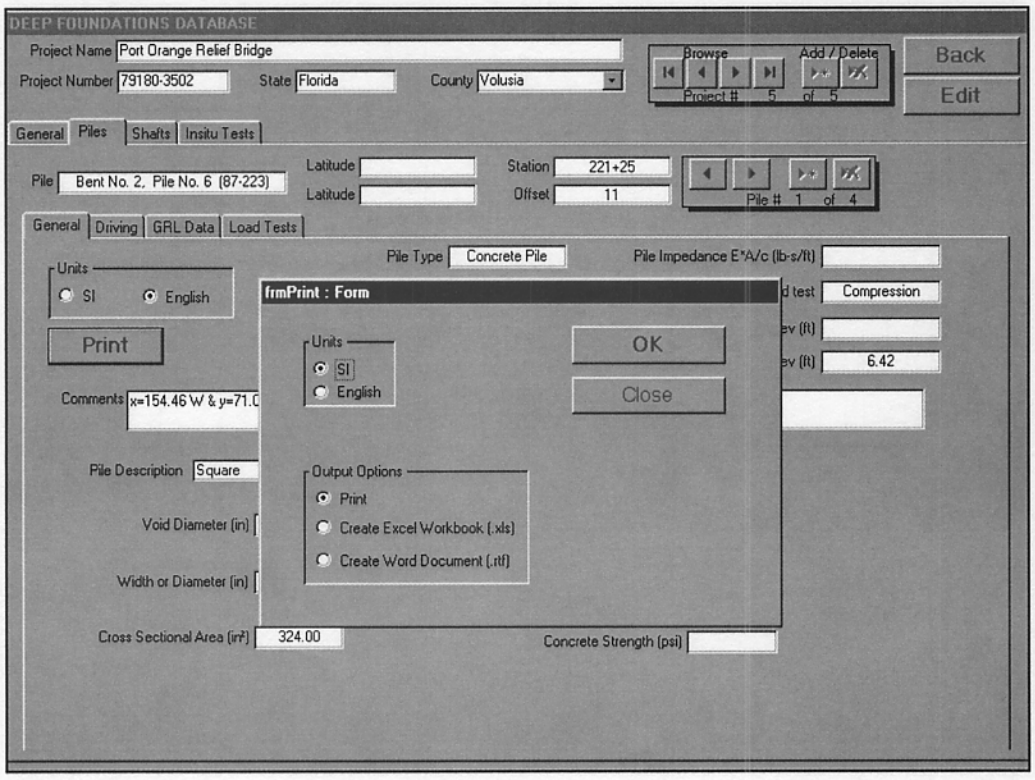


Figure 3.28 Print Form

Project Name: Port Orange Relief Bridge

Project Number: 79180-3502

Pile Information

Pile: Bent No. 2, Pile No. 6 (87-223)

Latitude:

Longitude:

Station: 221+25

Offset: 11

Pile Type: Concrete Pile

Type of Load test: Compression

Ground Surface Elev (ft): 6.43

Water Table Elev (ft):

Pile Elastic Modulus (psi):

Capwap Elastic Modulus (psi):

Pile Wave Speed c (ft/s):

Pile Impedance E*A/c (lb-s/ft):

Comments: x=154.46 W & y=71.07 S relative to the center of the bridge.

Pile Description: Square

Width (in): 18.00

Void Diameter (in):

Cross Sectional Area (in): 324.00

Unit Weight of Concrete (pcf):

Total Length (ft): 89.99

Embedded Length (ft):

Total Weight (ton):

Concrete Strength (psi):

Figure 3.29 Pile Information Report

3.6 Optimizing Database Performance

After numerous additions and changes are made to the objects within a database file, especially additions and deletions of data in tables, the database file can become disorganized. When a record is deleted, the space in the file that the deleted data occupied is not automatically regained. The database must be compacted periodically to optimize its file size and the organization of data within the tables that the file contains. To compact the database the user should go to the menu form, then close it by clicking on the close control (⊠), which is located in the upper right corner of the form. Choose "Tools, Database Utilities, Compact Database" from the Menu toolbar. Access immediately begins compacting the database. After a few seconds, the database compaction is completed and Access automatically returns to the Menu form.

If a message that the database is corrupted is displayed or if the database behaves in an irregular manner, one or more of its objects may be corrupted as the result of a hardware error. For example, databases can become corrupt as the result of a power failure when the computer is writing on the database file. In such cases, close the Menu form and choose "Tools, Database Utilities, Repair Database" from the Menu toolbar to attempt to repair the damage. If Access cannot repair corruption, the user would have to restore the latest database backup.

CHAPTER 4

DATABASE CONTENTS

4.1 Numbers of Records

All prior University of Florida deep foundation databases were based on driven pile or drilled shaft records. Information on that particular pile or shaft included dimensions, installation records, load test data and an associated SPT boring log. A recent Microsoft Access 97 database contained 435 such records, 213 piles and 222 shafts. The current database has instead been organized by project. Any project can then include multiple piles, shafts and insitu tests.

The current University of Florida Deep Foundation Database contains 189 records or projects. Eight-six projects have driven piles records, 99 have drilled shaft records, 3 have both piles and shafts and one has only insitu data. In total, there are 441 piles (374 concrete, 49 H-pile, 18 pipe pile), 419 drilled shafts and 1063 insitu tests (1009 SPT, 42 CPT, 12 DMT) in the database. The 189 projects are listed with corresponding numbers of piles, shafts and insitu tests in Appendix A.

Over the past six months, efforts have been concentrated on the seventeen Florida bridge records listed in Table 4.1. These were projects for which it was hoped geotechnical reports and site plans could be easily obtained. In future research projects, efforts will be continued to complete and add to these projects. Table 4.1 also provides numbers of driving records and numbers of pile and shaft load tests in the database.

No pile PDA, shaft IGM, Statnamic or Osterberg load test records have been input at this time.

Record #	Name	# Piles	# Driving Records	# Pile Load Tests	# Shafts	# Shaft Load Tests	# Insitu Tests			
							SPT	CPT	PMT	DMT
3	Acosta	67	67	3	4	-	72	-	-	2
5	Appalachicola River	28	28	9	-	-	9	-	-	-
153	Port Orange Intrcoastal	13	13	-	3	2	9	14	-	-
175	Port Orange Relief	4	4	2	-	-	2	7	-	-
176	Appalachicola Bay	37	37	5	-	-	28	2	-	-
177	Vilano	31	31	2	-	-	-	19	-	10
179	Blountstown	-	-	-	71		120	-	-	-
180	Dodge Island	8	8	1	-	-	6	-	-	-
181	Julington Creek	14	14	-	-	-	5	-	-	-
182	McArthur Causeway	-	-	-	5	1	45	-	-	-
183	Venetian Causeway	-	-	-	60	-	17	-	-	-
184	17th Street Causeway	-	-	-	16	-	75	-	-	-
185	Christa McAuliffe	-	-	-	17	-	11	-	-	-
186	Fuller Warren Replacement	-	-	-	10	-	72	-	-	-
187	Hillsborough River	-	-	-	5	-	65	-	-	-
188	Gandy Replacement	-	-	-	8	-	117	-	-	-
189	Victory Bridge Replacement	-	-	-	-	-	28	-	-	-
Totals	17 projects	202	202	22	199	3	681	42	-	12

Table 4.1 Florida Bridge Projects

4.2 Location of Pile, Shaft or Insitu Test

For future planned uses of the database it is desired to have each geotechnical member of the bridge (pile, shaft, insitu test) located by latitude and longitude. Since, for example the insitu tests, can no longer be physically located, their position will have to be calculated from station and off-set data included on plans and in reports. It was decided at this time to choose one known and easily recognized point on each bridge, which at some later date could be accurately located by GPS procedures, and locate all other members relative to this point. This section describes how the x- (East) and y- (North) coordinates of a pile, shaft or insitu test are calculated relative to the chosen point (the origin). Two Cone Penetration Tests, CPT-6P and CPT-7P, at the Vilano Bridge will be used as examples.

4.2.1 Choosing the Origin

The origin must be a point of known Station Number. It should also be easy to physically locate (e.g., the center of the channel which the bridge crosses or mid-point between two bents) in order to simplify the subsequent GPS survey. Otherwise some resurveying will be necessary to locate the point. Finally, the calculations are simplest if the point is chosen on a straight, rather than curved, section of the bridge.

4.2.2 Sketch the Bridge

Sketch the bridge center line with North in the +y direction. Line bearings are provided on the plans. Locate the origin on a straight line section. Note Station Numbers for important points such as the origin, points of curvature, points of tangency, beginning and end of the project. Figure 4.1 is a schematic sketch of the Vilano Bridge. CPT-6P is located at Station 172+40.00, right offset 35ft while CPT-7P is at Station 182+32.00, right offset 44ft.

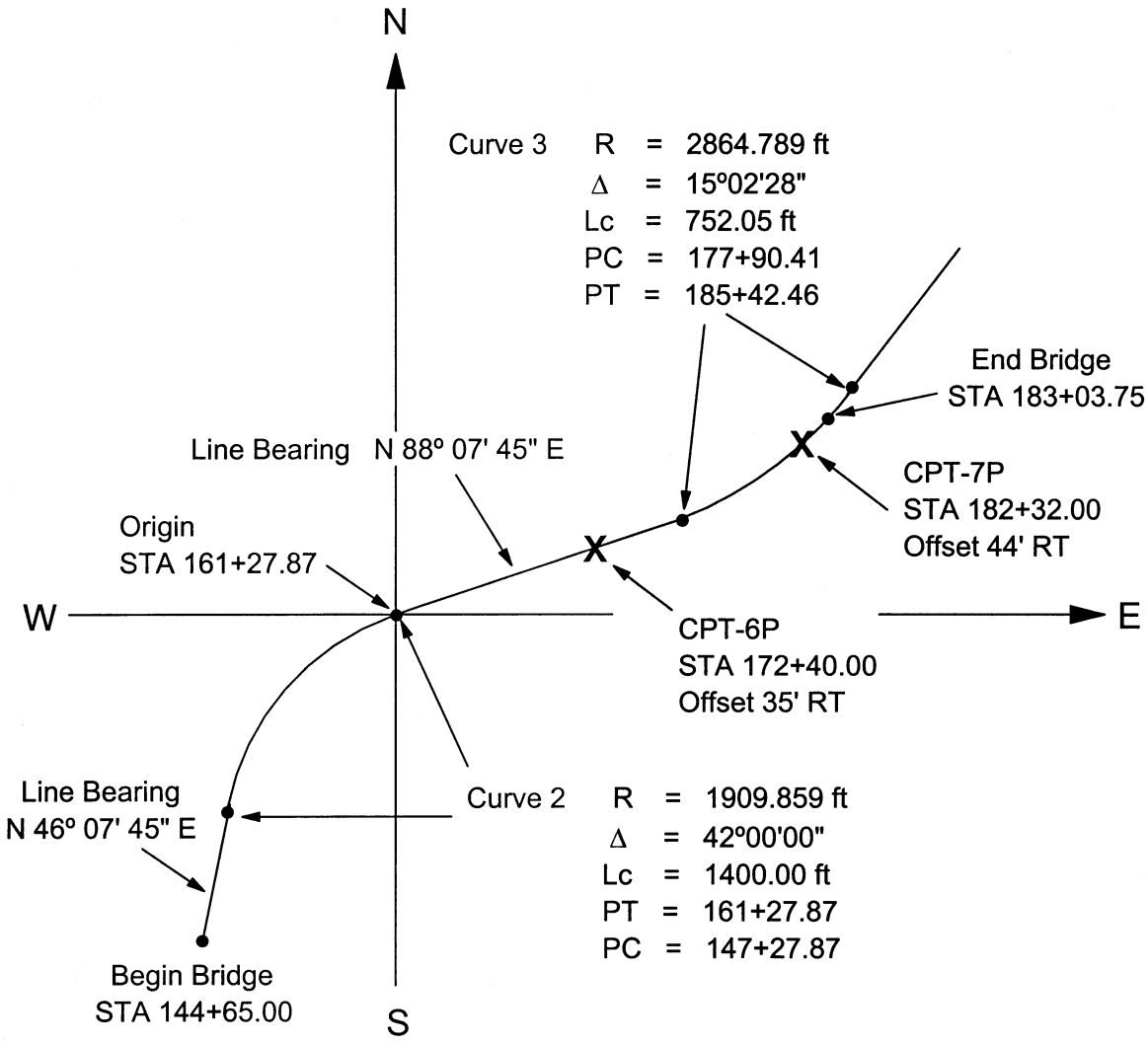


Figure 4.1 Sketch of Vilano Bridge Center Line

4.2.3 Straight Section

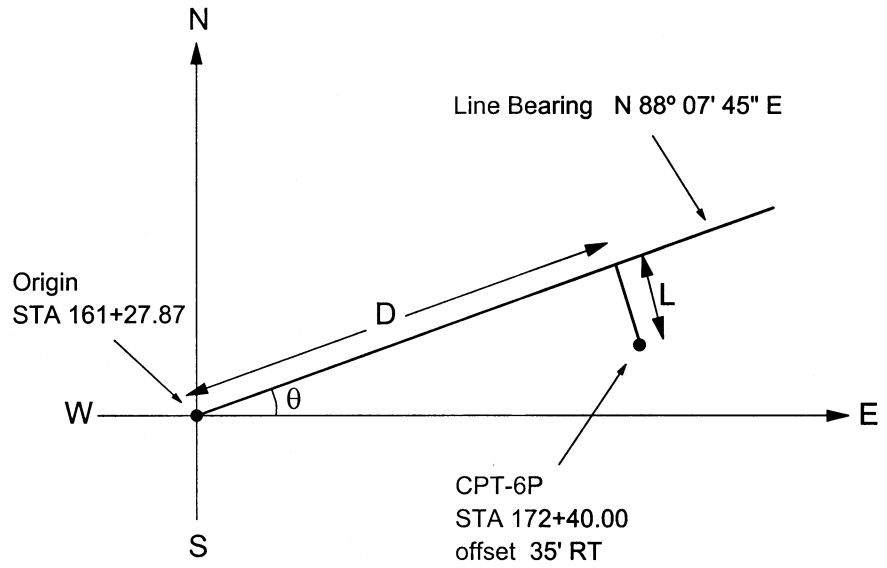
The straight section calculations are much simpler than when the point to be located is on a curved section. Figure 4.2 shows the calculations for Vilano Bridge CPT-6P. The steps of the procedure are as follows:

1. Subtract the point's station from the origin station to obtain the distance, D.
2. Determine the angle θ which is used in the sine and cosine equations from the bearing.
3. Calculate the x- and y-coordinates (X' and Y') of the point's center line location.
4. The offset of the point must then be considered. The offset will be either left or right of the center line. Depending on which way the bridge stationing is designated the offset will make the new coordinates either more negative or more positive than the center line point. Calculate Δx and Δy due to the offset, L.
5. "Add" the distance and offset contributions to obtain the new x- and y-coordinates.

4.2.3 Curved Section

The following information defining a curved segment is usually provided on the plans: the radius, the angle delta (central angle of the circular curve), the point of curvature (the beginning of the curve), the point of tangency (the end of the curve) and the length of the arc, L_C. Figure 4.3 shows the equations for the setup shown. In other cases (e.g., a left offset) some of the signs would be reversed. Figure 4.4 shows the calculations for Vilano Bridge CPT-7P. The steps of the procedure are as follows.

1. Subtract the PC station from the origin station to obtain the distance, D.
2. Determine the angle θ from the straight segment bearing.
3. Calculate the x- and y-components due to D.



Step 1 Distance D = STA_{POINT} - STA_{ORIGIN}

= 17240.00 - 16127.87 = 1112.13 ft

Step 2 $\theta = 90^{\circ} - 88.1291667 = 1.870833^{\circ}$

Step 3 $X' = D \cdot \cos\theta = (1112.13) \cdot (\cos 1.870833) = 1111.54$ ft

$Y' = D \cdot \sin\theta = (1112.13) \cdot (\sin 1.870833) = 36.31$ ft

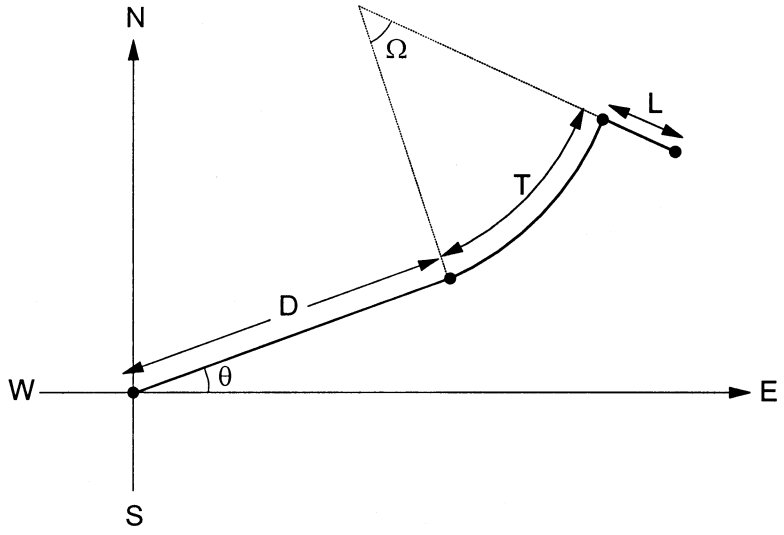
Step 4 $\Delta x = L \cdot \sin\theta = (35) \cdot (\sin 1.870833) = 1.14$ ft

$\Delta y = L \cdot \cos\theta = (35) \cdot (\cos 1.870833) = 34.98$ ft

Step 5 X = 1111.54 + 1.14 = 1112.68 ft East

Y = 36.31 - 34.98 = 1.33 ft North

Figure 4.2 Vilano Bridge CPT-6P Calculations



$$X = X' + X'' + \Delta X$$

$$Y = Y' + Y'' - \Delta Y$$

where: X' and Y' are components of the straight line segment D

X'' and Y'' are components of the curved segment T

ΔX and ΔY are components of the offset L

$$X' = D \cos \theta$$

$$Y' = D \sin \theta$$

$$X'' = 2\ R \sin \frac{\Omega}{2} \cos \left(\frac{\Omega}{2} + \theta\right)$$

$$Y'' = 2\ R \sin \frac{\Omega}{2} \sin \left(\frac{\Omega}{2} + \theta\right)$$

$$\Delta X = L \sin (\Omega + \theta)$$

$$\Delta Y = L \cos (\Omega + \theta)$$

Figure 4.3 Equations for a Point on a Curved Segment

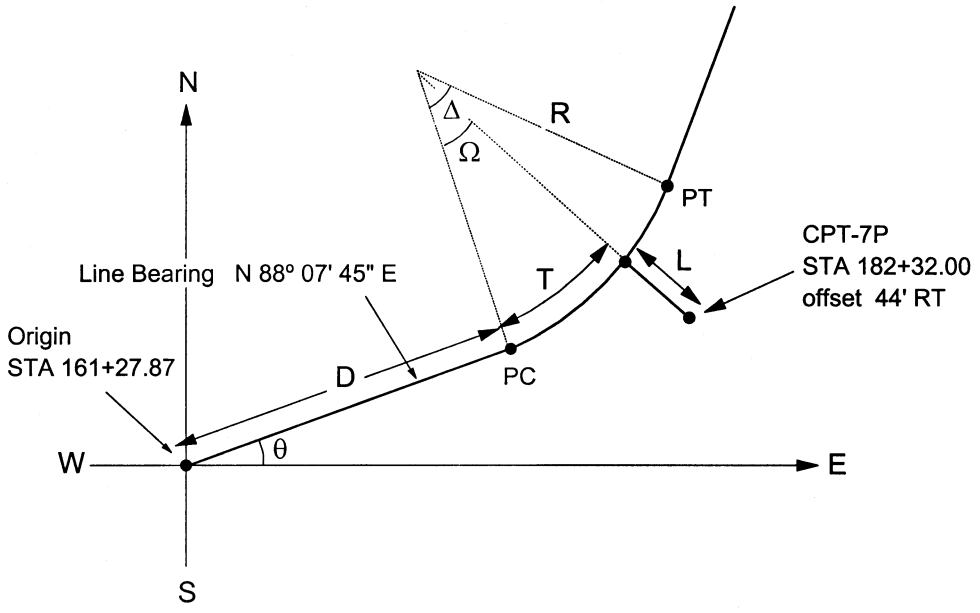
4. Subtract the point's station from the PC station to obtain the distance T.

5. Calculate the angle Ω subtended between PC and the point, from equality of ratios.

6. Calculate the x- and y-components due to T.

7. Calculate Δx and Δy due to the offset, L.

8. "Add" the distance, curve and offset contributions to obtain the new x- and y-coordinates.



Curve data: R = 2864.789 ft PC = 177+90.41

Δ = 15.041° PT = 185+42.46

L_C = 752.05 ft

Figure 4.4 Vilano Bridge CPT-7P Calculations

Step 1 Distance $D = STA_{PC} - STA_{ORIGIN}$

$D = 17790.41 - 16127.87 = 1662.54$ ft

Step 2 $\theta = 90^\circ - 88.1291667 = 1.870833^\circ$

Step 3 $X' = D \cdot \cos\theta = (1662.54) \cdot (\cos 1.870833) = 1661.65$ ft

$Y' = D \cdot \sin\theta = (1662.54) \cdot (\sin 1.870833) = 54.28$ ft

Step 4 Distance $T = STA_{CPT\text{-}TP} - STA_{PC}$

$T = 18232.00 - 17790.41 = 441.59$ ft

Step 5 $$\frac{\Delta}{L_C} = \frac{\Omega}{T}$$

$\Omega = (15.041^\circ)(441.59) / (752.05) = 8.832^\circ$

Step 6 $X'' = (2)(2864.789)(\sin 4.416)(\cos 6.287) = 438.510$ ft

$Y'' = (2)(2864.789)(\sin 4.416)(\sin 6.287) = 48.311$ ft

Step 7 $\Delta X = (44)(\sin 10.703) = 8.172$ ft

$\Delta Y = (44)(\cos 10.703) = 43.235$ ft

Step 8 $X = 1661.65 + 438.510 + 8.172 = 2108.332$ ft

$Y = 54.28 + 48.311 - 43.235 = 59.356$ ft

Figure 4.4 Vilano Bridge CPT-7P Calculations (Continued)

REFERENCES

Ahrens, M. D. (1995) "Driven Pile Database for Dynamic Testing Comparison." Master of Engineering Thesis, University of Florida, Gainesville, Florida.

Avasarala, S. K. V. (1993) "Pile Load Test Database and an Evaluation of the SPILE and UNIPILE Programs." Master of Engineering Thesis, University of Florida, Gainesville, Florida.

Davidson, J. L. and Fernandez, R. (1999). "Deep Foundation Databases." Research Progress Report Prepared for the Florida Department of Transportation, University of Florida, Gainesville, Florida.

Davidson, J. L., Smith, E. R. and Dornic S. D. (1997). "Evaluation of Static Capacity Prediction Between SPT94 and PILEUF Load Test Data." Research Final Report Prepared for the Florida Department of Transportation, University of Florida, Gainesville, Florida.

Davidson J. L. and Townsend, F. C. (1996). "Reliability and Limitations of Dynamic Methods for Predicting Capacity and Settlement of Driven Piles." Research Final Report Prepared for the Florida Department of Transportation, University of Florida, Gainesville, Florida.

Davidson J. L. and Townsend, F. C. (1994). "Maintenance of Load Test Databases." Research Final Report Prepared for the Florida Department of Transportation, University of Florida, Gainesville, Florida.

Dornic, S. D. (1995). "PILEUF Database and Evaluation of End Bearing in Limestone." Master of Engineering Thesis, University of Florida, Gainesville, Florida.

Fernandez, R. (1999). "Deep Foundations Database." Master of Engineering Report, University of Florida, Gainesville, Florida.

Hren, K. F. (1994). "The Effect of Damping Coefficient and Set on the Accuracy of Dynamic Calculations of Driven Concrete Pile Capacity." Master of Engineering Thesis, University of Florida, Gainesville, Florida.

Ruesta, R. F. (1992). "Pile Load Test Database and an Evaluation of the SPT91 Program." Master of Engineering Report, University of Florida, Gainesville, Florida.

Sharp, M. R. (1987). "An Evaluation of Currently Available Computer Programs for Predicting Capacity and Settlement of Axially Loaded Piles." Master of Engineering Thesis, University of Florida, Gainesville, Florida.

Smith, E. R. (1996). "Evaluation of Static Capacity Prediction Between SPT94 and PILEUF Load Test Data." Master of Engineering Thesis, University of Florida, Gainesville, Florida.

Spears, L. D. (1993). "Drilled Shaft Load Test Database and an Evaluation of the SHAFTUF Program." Master of Engineering Thesis, University of Florida, Gainesville, Florida.

APPENDIX A

LISTING OF THE 189 PROJECTS

Record Number	Project Name	Number of Piles	Number of Shafts	Number of Insitu Tests
1	Saint John's (ASCE) - 3A	1	0	1
2	49th Street Bridge	3	0	3
3	Acosta Bridge	67	4	74
4	Allegheny	8	0	8
5	Appalachicola River	28	0	9
6	Arvida Hotel	1	0	1
7	Ballona Creek	4	0	4
8	Beaches of Longboat	1	0	1
9	Blount Island	5	0	5
10	Buckman Bridge	4	0	4
11	Cape Canaveral	4	0	4
12	Choctawhatchee	39	0	10
13	Colorado River RTE8 Sec 96.88	1	0	1
14	Delaware	5	0	5
15	DeSota Condominiums	1	0	1

Record Number	Project Name	Number of Piles	Number of Shafts	Number of Insitu Tests
16	Dodge Island	6	0	6
17	Downtown Orlando Arena	1	0	1
18	Du Charme Residence TP3	1	0	1
19	Dupont Center, Jacksonville	1	0	1
20	Escambia River	2	0	2
21	FDOT	4	0	4
22	Florence / Marion	3	0	3
23	Fort Myers	1	0	1
24	Georgia/Florida Boundary	1	0	1
25	Hamilton Bayfront	3	0	3
26	Houston, Texas	1	0	1
27	Howard Frank	3	0	3
28	Hunter's Point	1	0	1
29	I-275 34th St. Pinellas	1	0	1
30	I-275 SR21 Pier 3R-37	1	0	1

Record Number	Project Name	Number of Piles	Number of Shafts	Number of Insitu Tests
31	I-275 / 103rd St. Pier 1R-P1	1	0	1
32	I-295 / CSX Bent 2R-16	1	0	1
33	I295 / I-10	3	0	3
34	I-295 / Melvin Rd Pier 2E-P1	1	0	1
35	I-295 / Mem. Pk Bent 2W-P12	1	0	1
36	I-295 / Ortega Riv Pier 3R-14	1	0	1
37	I-295 / SR 17 Pier 1L-11	1	0	1
38	I-295 / SR 228	2	0	2
39	I-295 / US 90	4	0	4
40	I-295 / Wilson Pier 2W-P3	1	0	1
41	I-95 West Palm Beach	2	0	2
42	Jacksonville Industrial	2	0	2
43	Jacksonville	2	0	2
44	Jacksonville, Florida	1	0	1
45	Julington Crk	10	0	10

Record Number	Project Name	Number of Piles	Number of Shafts	Number of Insitu Tests
46	Karidas Condominium	3	0	3
47	Longboat, Sarasota	2	0	2
48	Marco Island	2	0	2
49	Marina Bay Club TP7	1	0	1
50	Mayport N.A.S. Jacksonville	1	0	1
51	Northeast Villa Mirada - 6	1	0	1
52	NW Conn. OC Retrofit	2	0	2
53	Orlando D-22	1	0	1
54	Philadelphia	18	0	18
55	Pioneer Freezer Site	2	0	2
56	Players Club Villas Bridge	1	0	1
57	Port Orange	2	0	2
58	Roosevelt Bridge	2	0	2
59	Saint John River Site F	1	0	1
60	San Juan, PR	1	0	1

Record Number	Project Name	Number of Piles	Number of Shafts	Number of Insitu Tests
61	Sarasota Mem. Hospital	1	0	1
62	Seaway Hotels, Sand Key	1	0	1
63	Siesta Key, Sarasota	1	0	1
64	Southern Freeway, CA	1	0	1
65	SR 580 Oldsmar, FL	1	0	1
66	St. Augustine (ASCE) 4A	1	0	1
67	St. John's River (ASCE)	2	0	2
68	St. Laurent Tower 106	1	0	1
69	St. Louis, Missouri	2	0	2
70	St. Marissa Condominium	2	0	2
71	Stoneburner TP-SW-2-14	1	0	1
72	Sunset Resort Hotel	1	0	1
73	Sunshine Skyway Bridge	15	4	19
74	Surfrider Condominium	1	0	1
75	Talmadge Memorial Bridge	1	0	1

Record Number	Project Name	Number of Piles	Number of Shafts	Number of Insitu Tests
76	Tennessee-Tombigbee	3	0	3
77	Tombigbee Waterway	2	0	2
78	Veranda Hotel, Sarasota	1	0	1
79	Vienta Condominium	1	0	1
80	Washington Condominium	1	0	1
81	West Bay Bridge	2	0	2
82	White City Bridge	8	0	8
83	Unknown	0	1	1
84	T-BE Report 7	0	2	2
85	2100 Riveredge Parkway, Atlanta	0	1	1
86	Alamo, Nevada	0	2	0
87	Allegheny County	0	9	9
88	Americana Hotel, Tampa	0	2	2
89	Barnett Center, Jacksonville	0	1	1
90	Barnett Tower, Clearwater	0	1	1

Record Number	Project Name	Number of Piles	Number of Shafts	Number of Insitu Tests
91	Bat Yam, Israel	0	1	1
92	Bay St. Parking Garage, Jax.	0	1	1
93	Bayshore Place, Tampa	0	1	1
94	Brookfield, Nova Scotia	0	3	0
95	Brooksville	0	2	2
96	Brownsville, Texas	0	2	2
97	Burlington, Ontario	0	5	0
98	Cement Storage Silos, Dade Co.	0	1	0
99	Chicago. Illinois	0	3	0
100	Chicago, One Park Place	0	4	0
101	Coast Guard Tower	0	2	2
102	Cocoa Beach, Bevard, FL	0	1	0
103	Coventry, England	0	1	1
104	Dallas, Texas	0	1	0
105	Dallas, Texas Shaft #DT3	0	1	0

Record Number	Project Name	Number of Piles	Number of Shafts	Number of Insitu Tests
106	Denver Int. Airport	0	8	0
107	Dept of Ed. Building, Tallahassee	0	1	1
108	Enterprise Center Jax, FL	0	1	1
109	FHWA , Texas	0	5	4
110	FHWA - File HB & T	0	1	0
111	FHWA, Northside	0	2	2
112	FHWA, Florida	0	2	2
113	FHWA, Houston, Texas	0	3	0
114	FHWA, London	0	12	0
115	FHWA, Orange County	0	1	1
116	FHWA - File - Pile 1	0	1	1
117	FHWA - Site B	0	8	8
118	FHWA - File - South Carolina	0	1	1
119	Fort Collins, Colorado	0	1	1
120	Fort Carson, Colorado	0	2	0

Record Number	Project Name	Number of Piles	Number of Shafts	Number of Insitu Tests
121	France	0	1	1
122	Fruitland, NM	0	2	0
123	FHWA	0	1	1
124	Harlingen, TX - Test Shaft #3	0	1	1
125	Holly Hill, South Carolina	0	2	2
126	Hong Kong	0	1	1
127	Hong Kong	0	5	5
128	Houston, Texas	0	1	1
129	Howard Franklin Site 1	0	5	5
130	I-595 Ft. Lauderdale	0	2	2
131	Irving, Texas	0	2	0
132	Islamorada, FL	0	2	1
133	Isle of Palm Connector, SC	0	3	0
134	Jacksonville Center, FL	0	1	1
135	Johannesburg, S.A.	0	3	3

Record Number	Project Name	Number of Piles	Number of Shafts	Number of Insitu Tests
136	Kanapaha, Alachua Co., FL	0	1	1
137	Kilroot, N. Ireland	0	2	0
138	Lewis State Bank, Tallahassee	0	2	2
139	Littlebrook, UK Power Station	0	1	0
140	Live Oak County, Texas	0	1	1
141	Marriot Hotel, Miami, FL	0	1	0
142	Melbourne, Australia	0	3	0
143	Melbourne, Australia, #428	0	2	0
144	Memorial Hospital, Tampa, FL	0	1	1
145	Miami Center Phase ii Condo	0	1	1
146	Montopolis, Texas	0	3	0
147	North Halawa Valley, Hawaii	0	3	0
148	Oklahoma, USA	0	2	0
149	Orange County, Texas	0	1	1
150	Palm Beach - Hamptons, Fl	0	1	0

Record Number	Project Name	Number of Piles	Number of Shafts	Number of Insitu Tests
151	Phoenix, AZ	0	25	15
152	Port Elizabeth, S.A.	0	3	0
153	Port Orange Intracoastal Bridge	13	3	23
154	Registry Hotel, Naples, FL	0	1	0
155	Roanoke, TX - Westlake IBM	0	2	1
156	S. E. Financial Center, Miami	0	1	0
157	S. E. bank Building, Jax.	0	2	2
158	San Antonio, Texas	0	1	1
159	San Antonio, Lackland A.F.B.	0	3	0
160	Sao Paulo, Brazil	0	1	1
161	Saskatchewan, Canada	0	1	0
162	Singapore	0	4	0
163	Singer Island, Riviera Beach	0	1	1
164	Sohoio Corporate, Clev., Ohio	0	1	0
165	Stainless Electric	0	2	2

Record Number	Project Name	Number of Piles	Number of Shafts	Number of Insitu Tests
166	Suwannee River Bridge, Dixie	0	1	1
167	Talmadge memorial Bridge	0	2	2
168	Tampa International Airport	0	2	2
169	Tampa International Airport	0	1	1
170	Toronto, Ontario	0	3	3
171	University of Florida	0	1	1
172	Vienna, Virginia	0	1	1
173	Walkerton, Ontario, Canada	0	1	1
174	West Palm Beach Admin. Off. #1	0	2	2
175	Port Orange Relief Bridge	4	0	9
176	Appalachicola Bay bridge	37	0	30
177	Vilano Bridge Replace. SR A1A	31	0	29
178	White City Bridge - SR 71	9	0	7
179	SR20 Blountstown Bridge	0	71	120
180	Port of Miami Dodge Island	8	0	6

Record Number	Project Name	Number of Piles	Number of Shafts	Number of Insitu Tests
181	SR 13 over Julington Creek	14	0	5
182	McArthur Causeway Bridge	0	5	45
183	Venetian Causeway Bridge	0	60	17
184	SR A1A 17th St Causeway	0	16	75
185	Christa McAuliffe Bridge SR3	0	17	11
186	Fuller Warren Bridge Replace.	0	10	71
187	SR 600 Hillsborough River	0	5	65
188	Gandy Bridge SR 600	0	8	117
189	Victory Bridge Repl. SR10 US90	0	0	28

Totals	441	419	1063

APPENDIX B

DATABASE MODULES

MODULES

PublicVars Declaration

```
Option Compare Database
Option Explicit

'Defines global varialbes, Used in more than one form
Public CurrentProjectID As Integer
Public PileUnits As Integer
Public ShaftUnits As Integer
Public DrivingUnits As Integer
Public InsituUnits As Integer
Public LoadTestUnits As Integer
Public Report_Name As String
Public Print_Filter As String
Public Print_Units As Integer
Public Print_Excel As Integer
```

CLASS MODULES
(Contained in the forms)

frmGeneral

```
Option Compare Database
Option Explicit

Private Sub Form_Current()
   On Error Resume Next
   'CurrentProjectID is used in the printing filter
   CurrentProjectID = Me.[Project ID]
   Me!Box.Enabled = True
   DoCmd.Maximize
   'Updates the value of the current record on the navigation box
   Me![CurrentRecordNumber].Value = Me.CurrentRecord
   Me![Project Name].SetFocus
   'Denies access to the data tabs to force the user to input
   'a Project Name or Project Number before starting entering data
   If IsNull(Me![Project Name]) And IsNull(Me![Project Number]) Then
      Me!Box.Enabled = False
   End If
End Sub

'Allows access to the data tabs when a Project Name is entered
Private Sub Project_Name_AfterUpdate()
   Me!Box.Enabled = True
   CurrentProjectID = Me.[Project ID]
End Sub

'Allows access to the data tabs when a Project Number is entered
Private Sub Project_Number_AfterUpdate()
   Me!Box.Enabled = True
   CurrentProjectID = Me.[Project ID]
End Sub

'Moves to the previous record
Private Sub GoBack_Click()
On Error Resume Next
      DoCmd.GoToRecord , , acPrevious
End Sub

'Moves to the first record
Private Sub GoFirst_Click()
On Error Resume Next
      DoCmd.GoToRecord , , acFirst
End Sub

'Moves to the next record
Private Sub GoForward_Click()
On Error Resume Next
      DoCmd.GoToRecord , , acNext
End Sub

'Moves to the last record
Private Sub GoLast_Click()
On Error Resume Next
      DoCmd.GoToRecord , , acLast
End Sub
```

```
'Deletes the current record
Private Sub DeleteRecord_Click()
On Error Resume Next
   DoCmd.RunCommand acCmdDeleteRecord
   DoCmd.GoToRecord , , acPrevious
End Sub

'Starts a new record
Private Sub AddRecord_Click()
On Error Resume Next
      DoCmd.GoToRecord , , acNewRec
End Sub

'Closes the form
Private Sub BackButton_Click()
On Error Resume Next
      DoCmd.Close acForm, "frmGeneral"
End Sub

'Closes frmGeneral and opens frmsearch form
Private Sub Search_Click()
On Error Resume Next
      DoCmd.OpenForm "frmSearch"
      DoCmd.Close acForm, "frmGeneral"
End Sub
```

frmShafts

```
Option Compare Database
Option Explicit

Private Sub UnitsOff()
  'Displays a message if no units are selected prompting the user to select a set of units
  If IsNull(Me![Textbox_Units]) Then
    MsgBox "Please specify units"
    Me![Units].SetFocus
    'Erases the value assigned to the control
    DoCmd.RunCommand acCmdUndo
  End If
End Sub


Private Sub Form_Current()
  'Disables IGM and Load Tets Tabs until a shaft record is started
    If IsNull(Me![Textbox_Units]) Then
      Me![IGM Tab].Enabled = False
      Me![Load Tests].Enabled = False
    Else
      Me![IGM Tab].Enabled = True
      Me![Load Tests].Enabled = True
    End If
  'Shows the cursor in Latitude with the General tab selected
  Me!Units.SetFocus
  Me!Latitude.SetFocus
  Me!CurrentRecordNumber.Value = Me.CurrentRecord
  'Checks if a set of units is selected and updates the Units control
  Select Case Me![Textbox_Units]
  Case "SI"
  Me![Units].Value = 1
  Case "English"
  Me![Units].Value = 2
  Case Else
  Me![Units].Value = 0
  End Select
  'Updates the ShaftUnits variable
  ShaftUnits = Me![Units]
  'Calls the UnitsShaft subprocedure
  UnitsShaft
End Sub

Private Sub Load_test_Enter()
'Updates the load test units to avoid problems in data input
LoadTestUnits = Me![Load Test]![Units]
End Sub

Private Sub Units_AfterUpdate()
DoCmd.SetWarnings False
 'Updates the Textbox_Units control if units are selected for the first time
 If ShaftUnits = 0 Then
    Select Case Me![Units]
      Case 1
        Me![Textbox_Units].Value = "SI"
      Case 2
        Me![Textbox_Units].Value = "English"
    End Select
```

```
'Updates the ShaftUnits variable
    ShaftUnits = Me![Units]
    'Calls the UnitsShaft subprocedure
    UnitsShaft
        'Enables the IGM and Load Tests Tags
        Me![IGM Tab].Enabled = True
        Me![Load Tests].Enabled = True
 Else
    'Displays a message warning the deletion of the current data
    'if the user attempts to change the units
    If MsgBox("Changing the Units will Erase the current Shaft Data", vbOKCancel) = vbOK Then
      'Deletes the record if the user clicks OK
      DoCmd.RunCommand acCmdDeleteRecord
    Else
      'Displays the current units if the user clicks Cancel
      Me![Units] = ShaftUnits
    End If
 End If
DoCmd.SetWarnings True
End Sub


Private Sub UnitsShaft()
'Changes all the labels to the specified set of units
Select Case ShaftUnits
  Case 1
    Me![Diameter Label].Caption = "Diameter (mm)"
    Me![Total Length Label].Caption = "Total Length (m)"
    Me![Embeded Length Label].Caption = "Embeded Length (m)"
    Me![Casing Length Label].Caption = "Casing Length (m)"
    Me![Bell Diameter Label].Caption = "Bell Diameter (mm)"
    Me![Bell Length Label].Caption = "Bell Length (m)"
    Me![Slump of Concrete Label].Caption = "Slump of Concrete (mm)"
    Me![Concrete strength Label].Caption = "Concrete strength f'c (KN/m²)"
    Me![Core Modulus Label].Caption = "Core Modulus Ec (KN/m²)"
    Me![Elev Label].Caption = "(m)"
    Me![Depth Label].Caption = "(m)"
    Me![qu Label].Caption = "(KN/m²)"
    Me![qt Label].Caption = "(KN/m²)"
    Me![Em Label].Caption = "(KN/m²)"
    Me![Ei Label].Caption = "(KN/m²)"
    Me![Water Table Elev Label].Caption = "Water Table Elev (m)"
    Me![Ground Surface Elev Label].Caption = "Ground Surface Elev (m)"
    Me![As built Design Load  Label].Caption = "Design Load (kN)"
    Me![As Built Diameter Label].Caption = "Diameter (mm)"
    Me![As Built Tip Elev Label].Caption = "Tip Elev (m)"
    Me![As Built Total Length Label].Caption = "Total Length (m)"
    Me![As Built Rock Socket Length Label].Caption = "Rock Socket Length (m)"
    Me![Top Elev Label].Caption = "Top Elev (m)"
    Me![Tip Elev Label].Caption = "Tip Elev (m)"
    Me![Scour Elev Label].Caption = "Scour Elev (m)"
    Me![Rock Socket Length Label].Caption = "Rock Socket Length (m)"
    Me![Design Load Label].Caption = "Design Load (kN)"
  Case 2
    Me![Diameter Label].Caption = "Diameter (in)"
    Me![Total Length Label].Caption = "Total Length (ft)"
    Me![Embeded Length Label].Caption = "Embeded Length (ft)"
    Me![Casing Length Label].Caption = "Casing Length (ft)"
    Me![Bell Diameter Label].Caption = "Bell Diameter (in)"
    Me![Bell Length Label].Caption = "Bell Length (ft)"
    Me![Slump of Concrete Label].Caption = "Slump of Concrete (in)"
    Me![Concrete strength Label].Caption = "Concrete strength f'c (Ton/ft²)"
```

```
    Me![Core Modulus Label].Caption = "Core Modulus Ec (Ton/ft²)"
    Me![Elev Label].Caption = "(ft)"
    Me![Depth Label].Caption = "(ft)"
    Me![qu Label].Caption = "(Ton/ft²)"
    Me![qt Label].Caption = "(Ton/ft²)"
    Me![Em Label].Caption = "(Ton/ft²)"
    Me![Ei Label].Caption = "(Ton/ft²)"
    Me![Water Table Elev Label].Caption = "Water Table Elev (ft)"
    Me![Ground Surface Elev Label].Caption = "Ground Surface Elev (ft)"
    Me![As built Design Load  Label].Caption = "Design Load (tons)"
    Me![As Built Diameter Label].Caption = "Diameter (in)"
    Me![As Built Tip Elev Label].Caption = "Tip Elev (ft)"
    Me![As Built Total Length Label].Caption = "Total Length (ft)"
    Me![As Built Rock Socket Length Label].Caption = "Rock Socket Length (ft)"
    Me![Top Elev Label].Caption = "Top Elev (ft)"
    Me![Tip Elev Label].Caption = "Tip Elev (ft)"
    Me![Scour Elev Label].Caption = "Scour Elev (ft)"
    Me![Rock Socket Length Label].Caption = "Rock Socket Length (ft)"
    Me![Design Load Label].Caption = "Design load (tons)"
  Case Else
    Me![Diameter Label].Caption = "Diameter"
    Me![Total Length Label].Caption = "Total Length"
    Me![Embeded Length Label].Caption = "Embeded Length"
    Me![Casing Length Label].Caption = "Casing Length"
    Me![Bell Diameter Label].Caption = "Bell Diameter"
    Me![Bell Length Label].Caption = "Bell Length"
    Me![Slump of Concrete Label].Caption = "Slump of Concrete"
    Me![Concrete strength Label].Caption = "Concrete strength f'c"
    Me![Core Modulus Label].Caption = "Core Modulus Ec"
    Me![Elev Label].Caption = ""
    Me![Depth Label].Caption = ""
    Me![qu Label].Caption = ""
    Me![qt Label].Caption = ""
    Me![Em Label].Caption = ""
    Me![Ei Label].Caption = ""
    Me![Water Table Elev Label].Caption = "Water Table Elev"
    Me![Ground Surface Elev Label].Caption = "Ground Surface Elev"
    Me![As built Design Load  Label].Caption = "Design Load"
    Me![As Built Diameter Label].Caption = "Diameter"
    Me![As Built Tip Elev Label].Caption = "Tip Elev"
    Me![As Built Total Length Label].Caption = "Total Length"
    Me![As Built Rock Socket Length Label].Caption = "Rock Socket Length"
    Me![Top Elev Label].Caption = "Top Elev"
    Me![Tip Elev Label].Caption = "Tip Elev"
    Me![Scour Elev Label].Caption = "Scour Elev"
    Me![Rock Socket Length Label].Caption = "Rock Socket Length"
    Me![Design Load Label].Caption = "Design Load"
  End Select
End Sub

'All of the following subprocedures have the same structure, they assign a value
'to the SI units control based on the value input in the Raw data field and on
'the selected set of units

Private Sub Diameter_AfterUpdate()
  'Calls the UnitsOff subprocedure
  UnitsOff
  'Calculates the value of the SI control
  Select Case ShaftUnits
  Case 1
    Me![Diameter (mm)] = Me![Diameter]
```

```
 Case 2
     Me![Diameter (mm)] = Me![Diameter] * 25.4
   End Select
End Sub

Private Sub Total_Length_AfterUpdate()
   UnitsOff
   Select Case ShaftUnits
   Case 1
     Me![Total Length (m)] = Me![Total Length]
   Case 2
     Me![Total Length (m)] = Me![Total Length] * 0.3048
   End Select
End Sub

Private Sub Embedded_Length_AfterUpdate()
   UnitsOff
   Select Case ShaftUnits
   Case 1
     Me![Embedded Length (m)] = Me![Embedded Length]
   Case 2
     Me![Embedded Length (m)] = Me![Embedded Length] * 0.3048
   End Select
End Sub

Private Sub Casing_Length_AfterUpdate()
   UnitsOff
   Select Case ShaftUnits
   Case 1
     Me![Casing Length (m)] = Me![Casing Length]
   Case 2
     Me![Casing Length (m)] = Me![Casing Length] * 0.3048
   End Select
End Sub

Private Sub Bell_Diameter_AfterUpdate()
   UnitsOff
   Select Case ShaftUnits
   Case 1
     Me![Bell Diameter (mm)] = Me![Bell Diameter]
   Case 2
     Me![Bell Diameter (mm)] = Me![Bell Diameter] * 25.4
   End Select
End Sub

Private Sub Bell_Length_AfterUpdate()
   UnitsOff
   Select Case ShaftUnits
   Case 1
     Me![Bell Length (m)] = Me![Bell Length]
   Case 2
     Me![Bell Length (m)] = Me![Bell Length] * 0.3048
   End Select
End Sub

Private Sub Slump_of_Concrete_AfterUpdate()
   UnitsOff
   Select Case ShaftUnits
   Case 1
     Me![Slump of Concrete(mm)] = Me![Slump of Concrete]
   Case 2
     Me![Slump of Concrete(mm)] = Me![Slump of Concrete] * 25.4
```

```
  End Select
End Sub

Private Sub Concrete_strength__f_c_AfterUpdate()
  UnitsOff
  Select Case ShaftUnits
  Case 1
    Me![Concrete strength  f'c (KN/m²)] = Me![Concrete strength  f'c]
  Case 2
    Me![Concrete strength  f'c (KN/m²)] = Me![Concrete strength  f'c] * 95.7605
  End Select
End Sub

Private Sub Core_Modulus__Ec_AfterUpdate()
  UnitsOff
  Select Case ShaftUnits
  Case 1
    Me![Core Modulus  Ec (KN/m²)] = Me![Core Modulus  Ec]
  Case 2
    Me![Core Modulus  Ec (KN/m²)] = Me![Core Modulus  Ec] * 95.7605
  End Select
End Sub

Private Sub Top_Elev_AfterUpdate()
  UnitsOff
  Select Case ShaftUnits
  Case 1
    Me![Top Elev (m)] = Me![Top Elev]
  Case 2
    Me![Top Elev (m)] = Me![Top Elev] * 0.3048
  End Select
End Sub

Private Sub Tip_Elev_AfterUpdate()
  UnitsOff
  Select Case ShaftUnits
  Case 1
    Me![Tip Elev (m)] = Me![Tip Elev]
  Case 2
    Me![Tip Elev (m)] = Me![Tip Elev] * 0.3048
  End Select
End Sub

Private Sub Scour_Elev_AfterUpdate()
  UnitsOff
  Select Case ShaftUnits
  Case 1
    Me![Scour Elev (m)] = Me![Scour Elev]
  Case 2
    Me![Scour Elev (m)] = Me![Scour Elev] * 0.3048
  End Select
End Sub

Private Sub Rock_Socket_Length_AfterUpdate()
  UnitsOff
  Select Case ShaftUnits
  Case 1
    Me![Rock Socket Length (m)] = Me![Rock Socket Length]
  Case 2
    Me![Rock Socket Length (m)] = Me![Rock Socket Length] * 0.3048
  End Select
End Sub
```

```
Private Sub Design_Load_AfterUpdate()
  UnitsOff
  Select Case ShaftUnits
  Case 1
    Me![Design Load (kN)] = Me![Design Load]
  Case 2
    Me![Design Load (kN)] = Me![Design Load] * 8.8964
  End Select
End Sub

Private Sub Water_Table_Elev_AfterUpdate()
UnitsOff
  Select Case ShaftUnits
  Case 1
    Me![Water Table Elev (m)] = Me![Water Table Elev]
  Case 2
    Me![Water Table Elev (m)] = Me![Water Table Elev] * 0.3048
  End Select
End Sub


Private Sub Ground_Surface_Elev_AfterUpdate()
  UnitsOff
  Select Case ShaftUnits
  Case 1
    Me![Ground Surface Elev (m)] = Me![Ground Surface Elev]
  Case 2
    Me![Ground Surface Elev (m)] = Me![Ground Surface Elev] * 0.3048
  End Select
End Sub


Private Sub As_Built_Diameter_AfterUpdate()
  UnitsOff
  Select Case ShaftUnits
  Case 1
    Me![As Built Diameter (mm)] = Me![As Built Diameter]
  Case 2
    Me![As Built Diameter (mm)] = Me![As Built Diameter] * 25.4
  End Select
End Sub

Private Sub As_Built_Tip_Elev_AfterUpdate()
  UnitsOff
  Select Case ShaftUnits
  Case 1
    Me![As Built Tip Elev (m)] = Me![As Built Tip Elev]
  Case 2
    Me![As Built Tip Elev (m)] = Me![As Built Tip Elev] * 0.3048
  End Select
End Sub

Private Sub As_Built_Total_Length_AfterUpdate()
  UnitsOff
  Select Case ShaftUnits
  Case 1
    Me![As Built Total length (m)] = Me![As Built Total Length]
  Case 2
    Me![As Built Total length (m)] = Me![As Built Total Length] * 0.3048
  End Select
End Sub
```

```
Private Sub As_Built_Rock_Socket_Length_AfterUpdate()
  UnitsOff
  Select Case ShaftUnits
  Case 1
    Me![As Built Rock Socket Length (m)] = Me![As Built Rock Socket Length]
  Case 2
    Me![As Built Rock Socket Length (m)] = Me![As Built Rock Socket Length] * 0.3048
  End Select
End Sub

Private Sub As_built_Design_Load_AfterUpdate()
  UnitsOff
  Select Case ShaftUnits
  Case 1
    Me![As built Design Load (kN)] = Me![As built Design Load]
  Case 2
    Me![As built Design Load (kN)] = Me![As built Design Load] * 8.8964
  End Select
End Sub


'Moves to the previous record
Private Sub GoBack_Click()
On Error Resume Next
  DoCmd.GoToRecord , , acPrevious
End Sub

'Moves to the next record
Private Sub GoForward_Click()
On Error Resume Next
  DoCmd.GoToRecord , , acNext
End Sub

'Deletes the current record
Private Sub DeleteRecord_Click()
On Error Resume Next
  DoCmd.RunCommand acCmdDeleteRecord
  DoCmd.GoToRecord , , acPrevious
End Sub

'Starts a new record
Private Sub AddRecord_Click()
On Error Resume Next
  DoCmd.GoToRecord , , acNewRec
End Sub
```

sbfIGM

```
Option Compare Database
Option Explicit

'All of the following subprocedures have the same structure, they assign a value to the SI units control based on the
'value input in the Raw data field and on the selected set of units
Public Sub Elev_AfterUpdate()
   Select Case ShaftUnits
   Case 1
      Me![Elev (m)] = Me![Elev]
   Case 2
      Me![Elev (m)] = Me![Elev] * 0.3048
   End Select
End Sub

Public Sub Depth_AfterUpdate()
   Select Case ShaftUnits
   Case 1
      Me![Depth (m)] = Me![Depth]
   Case 2
      Me![Depth (m)] = Me![Depth] * 0.3048
   End Select
End Sub

Public Sub qu_AfterUpdate()
   Select Case ShaftUnits
   Case 1
      Me![qu (KN/m²)] = Me![qu]
   Case 2
      Me![qu (KN/m²)] = Me![qu] * 95.7605
   End Select
End Sub

Private Sub qt_AfterUpdate()
   Select Case ShaftUnits
   Case 1
      Me![qt (KN/m²)] = Me![qt]
   Case 2
      Me![qt (KN/m²)] = Me![qt] * 95.7605
   End Select
End Sub

Private Sub Em_AfterUpdate()
   Select Case ShaftUnits
   Case 1
      Me![Em (KN/m²)] = Me![Em]
   Case 2
      Me![Em (KN/m²)] = Me![Em] * 95.7605
   End Select
End Sub

Private Sub Ei_AfterUpdate()
   Select Case ShaftUnits
   Case 1
      Me![Ei  (KN/m²)] = Me![Ei]
   Case 2
      Me![Ei  (KN/m²)] = Me![Ei] * 95.7605
   End Select
End Sub
```

frmPiles

```
Option Compare Database
Option Explicit
Dim CurrentPile As String

Private Sub Form_Current()
'Disables the Driving, GRL and Load Test Tabs until a pile record is started
    If IsNull(Me![Textbox_Units]) Then
      Me![Driving].Enabled = False
      Me![GRL Data].Enabled = False
      Me![Load Tests].Enabled = False
    Else
      Me![Driving].Enabled = True
      Me![GRL Data].Enabled = True
      Me![Load Tests].Enabled = True
    End If
  'Calls the ViewP subprocedure
  ViewP
  'Updates the value of the Load test current record in the navigation box
  Me!CurrentRecordNumber.Value = Me.CurrentRecord
  'Checks if a set of units is selected and updates the Units control
  Select Case Me![Textbox_Units]
  Case "SI"
  Me![Units].Value = 1
  Case "English"
  Me![Units].Value = 2
  Case Else
  Me![Units].Value = 0
  End Select
  'Updates the PileUnits variable
  PileUnits = Me![Units]
  'Calls the UnitsPile subprocecure
  UnitsPile
  Me![Latitude].SetFocus
End Sub

Private Sub UnitsOff()
  'Displays a message if no units are selected prompting the user to select a set of units
  If IsNull(Me![Textbox_Units]) Then
    MsgBox "Please specify units"
    Me![Units].SetFocus
    'Erases the value assigned to the control
    DoCmd.RunCommand acCmdUndo
  End If
End Sub

Private Sub Pile_Load_Tests_Enter()
'Updates the load test units to avoid problems in data input
LoadTestUnits = Me![Pile Load Tests]![Units]
End Sub

Private Sub Pile_Type_Change()
DoCmd.SetWarnings False
  'Calls the ViewP subprocedure if the Pile Type is selected for the first time
  If CurrentPile = "No Pile" Then
  UnitsOff
  ViewP
  Else
```

```
 'Displays a message warning the deletion of the current data
 'if the user attempts to change the insitu test type
     If MsgBox("Changing the Pile type will Erase all data related to this pile", vbOKCancel) = vbOK Then
        'Deletes the record if the user clicks OK
        DoCmd.RunCommand acCmdDeleteRecord
     Else
        'Displays the current pile type if the user clicks Cancel
        Me![Pile Type] = CurrentPile
     End If
  End If
DoCmd.SetWarnings True
End Sub

Private Sub Units_AfterUpdate()
DoCmd.SetWarnings False
 'Updates the Textbox_Units control if units are selected for the first time
 If PileUnits = 0 Then
    Select Case Me![Units]
     Case 1
      Me![Textbox_Units].Value = "SI"
     Case 2
      Me![Textbox_Units].Value = "English"
    End Select
    'Updates the PileUnits variable
    PileUnits = Me![Units]
    'Calls the UnitsPile subprocedure
    UnitsPile
     'Enables the Driving, GRL and Load Tests tabs
     Me![Driving].Enabled = True
     Me![GRL Data].Enabled = True
     Me![Load Tests].Enabled = True
 Else
    'Displays a message warning the deletion of the current data
    'if the user attempts to change the units
    If MsgBox("Changing the Units will Erase all data related to this pile", vbOKCancel) = vbOK Then
      'Deletes the record if the user clicks OK
      DoCmd.RunCommand acCmdDeleteRecord
    Else
      'Displays the current units if the user clicks Cancel
      Me![Units] = PileUnits
    End If
 End If
DoCmd.SetWarnings True
End Sub

Private Sub ViewP()
   'Resets the visible properties of the tab control
   Me![Units].SetFocus
   Me![H Steel].Visible = False
   Me![Pipe].Visible = False
   Me![Concrete].Visible = False
  'Displays only the tab for the selected pile type and Updates the CurrentPile variable
  Select Case Me![Pile Type]
        Case "Concrete Pile"
         Me![Concrete].Visible = True
         CurrentPile = "Concrete Pile"
        Case "H Steel Pile"
         Me![H Steel].Visible = True
         CurrentPile = "H Steel Pile"
        Case "Pipe Steel Pile"
         Me![Pipe].Visible = True
         CurrentPile = "Pipe Steel Pile"
```

```
        Case Else
         CurrentPile = "No Pile"
  End Select
End Sub

Private Sub UnitsPile()
   'Changes all the labels to the specified set of units
   Select Case PileUnits
   Case 1
      Me![Pile Elastic Modulus Label].Caption = "Pile Elastic Modulus (KN/m²)"
      Me![Capwap Elastic Modulus Label].Caption = "Capwap Elastic Modulus (KN/m²)"
      Me![Pile Wave Speed Label].Caption = "Pile Wave Speed c (m/s)"
      Me![Pile Impedance Label].Caption = "Pile Impedance E*A/c (KN-s/m)"
      Me![Water Table Elev Label].Caption = "Water Table Elev (m)"
      Me![Ground Surface Elev Label].Caption = "Ground Surface Elev (m)"
      Me![Concrete]![Void Diameter Label].Caption = "Void Diameter (mm)"
      Me![Concrete]![Width Label].Caption = "Width or Diameter (mm)"
      Me![Concrete]![Cross Sectional Area Label].Caption = "Cross Sectional Area (mm²)"
      Me![Concrete]![Total Length Label].Caption = "Total Length (m)"
      Me![Concrete]![Embedded Length Label].Caption = "Embedded Length (m)"
      Me![Concrete]![Total Weight Label].Caption = "Total Weight (KN)"
      Me![Concrete]![Unit Weight of Concrete Label].Caption = "Unit Weight of Concrete (KN/m^3)"
      Me![Concrete]![Concrete strength Label].Caption = "Concrete Strength (KN/m²)"
      Me![H Steel]![Steel Cross Sectional Area Label].Caption = "Steel Cross Sectional Area (mm²)"
      Me![H Steel]![Pile Cross Sectional Area Label].Caption = "Pile Cross Sectional Area (mm²)"
      Me![H Steel]![Steel Section Perimeter Label].Caption = "Steel Section Perimeter (mm)"
      Me![H Steel]![Pile  Perimeter Label].Caption = "Pile  Perimeter (mm)"
      Me![H Steel]![Weight of Steel  Label].Caption = "Weight of Steel (KN/m)"
      Me![H Steel]![Weight of pile Label].Caption = "Weight of pile (KN)"
      Me![H Steel]![Total Length Label].Caption = "Total Length (m)"
      Me![H Steel]![Embedded Length Label].Caption = "Embedded Length (m)"
      Me![H Steel]![Unit Weight of Steel Label].Caption = "Unit Weight of Steel (KN/m^3)"
      Me![H Steel]![Concrete strength Label].Caption = "Concrete Strength (KN/m²)"
      Me![Pipe]![Outside Diameter Label].Caption = "Outside Diameter (mm)"
      Me![Pipe]![Thickness of Steel Label].Caption = "Thickness of Steel (mm)"
      Me![Pipe]![Steel Cross Sectional Area Label].Caption = "Steel Cross Sectional Area (mm²)"
      Me![Pipe]![Pile Cross Sectional Area Label].Caption = "Pile Cross Sectional Area (mm²)"
      Me![Pipe]![Concrete strength Label].Caption = "Concrete Strength (KN/m²)"
      Me![Pipe]![Weight of Steel Label].Caption = "Weight of Steel (KN/m)"
      Me![Pipe]![Weight of pile Label].Caption = "Weight of  Pile (KN)"
      Me![Pipe]![Total Length Label].Caption = "Total Length (m)"
      Me![Pipe]![Embedded Length Label].Caption = "Embedded Length (m)"
      Me![Pipe]![Unit Weight of Steel Label].Caption = "Unit Weight of Steel (KN/m^3)"
   Case 2
      Me![Pile Elastic Modulus Label].Caption = "Pile Elastic Modulus (psi)"
      Me![Capwap Elastic Modulus Label].Caption = "Capwap Elastic Modulus (psi)"
      Me![Pile Wave Speed Label].Caption = "Pile Wave Speed c (ft/s)"
      Me![Pile Impedance Label].Caption = "Pile Impedance E*A/c (lb-s/ft)"
      Me![Water Table Elev Label].Caption = "Water Table Elev (ft)"
      Me![Ground Surface Elev Label].Caption = "Ground Surface Elev (ft)"
      Me![Concrete]![Void Diameter Label].Caption = "Void Diameter (in)"
      Me![Concrete]![Width Label].Caption = "Width or Diameter (in)"
      Me![Concrete]![Cross Sectional Area Label].Caption = "Cross Sectional Area (in²)"
      Me![Concrete]![Total Length Label].Caption = "Total Length (ft)"
      Me![Concrete]![Embedded Length Label].Caption = "Embedded Length (ft)"
      Me![Concrete]![Total Weight Label].Caption = "Total Weight (tons)"
      Me![Concrete]![Unit Weight of Concrete Label].Caption = "Unit Weight of Concrete (pcf)"
      Me![Concrete]![Concrete strength Label].Caption = "Concrete Strength (psi)"
      Me![H Steel]![Steel Cross Sectional Area Label].Caption = "Steel Cross Sectional Area (in²)"
      Me![H Steel]![Pile Cross Sectional Area Label].Caption = "Pile Cross Sectional Area (in²)"
      Me![H Steel]![Steel Section Perimeter Label].Caption = "Steel Section Perimeter (in)"
      Me![H Steel]![Pile  Perimeter Label].Caption = "Pile  Perimeter (in)"
```

```
    Me![H Steel]![Weight of Steel  Label].Caption = "Weight of Steel (tons/ft)"
    Me![H Steel]![Weight of pile Label].Caption = "Weight of pile (tons)"
    Me![H Steel]![Total Length Label].Caption = "Total Length (ft)"
    Me![H Steel]![Embedded Length Label].Caption = "Embedded Length (ft)"
    Me![H Steel]![Unit Weight of Steel Label].Caption = "Unit Weight of Steel (pcf)"
    Me![H Steel]![Concrete strength Label].Caption = "Concrete Strength (psi)"
    Me![Pipe]![Outside Diameter Label].Caption = "Outside Diameter (in)"
    Me![Pipe]![Thickness of Steel Label].Caption = "Thickness of Steel (in)"
    Me![Pipe]![Steel Cross Sectional Area Label].Caption = "Steel Cross Sectional Area (in²)"
    Me![Pipe]![Pile Cross Sectional Area Label].Caption = "Pile Cross Sectional Area (in²)"
    Me![Pipe]![Concrete strength Label].Caption = "Concrete Strength (psi)"
    Me![Pipe]![Weight of Steel Label].Caption = "Weight of Steel (tons/ft)"
    Me![Pipe]![Weight of pile Label].Caption = "Weight of  Pile (tons)"
    Me![Pipe]![Total Length Label].Caption = "Total Length (ft)"
    Me![Pipe]![Embedded Length Label].Caption = "Embedded Length (ft)"
    Me![Pipe]![Unit Weight of Steel Label].Caption = "Unit Weight of Steel (pcf)"
  Case Else
    Me![Pile Elastic Modulus Label].Caption = "Pile Elastic Modulus"
    Me![Capwap Elastic Modulus Label].Caption = "Capwap Elastic Modulus"
    Me![Pile Wave Speed Label].Caption = "Pile Wave Speed ( c )"
    Me![Pile Impedance Label].Caption = "Pile Impedance ( E*A/c )"
    Me![Water Table Elev Label].Caption = "Water Table Elev"
    Me![Ground Surface Elev Label].Caption = "Ground Surface Elev"
    Me![Concrete]![Void Diameter Label].Caption = "Void Diameter"
    Me![Concrete]![Width Label].Caption = "Width or Diameter"
    Me![Concrete]![Cross Sectional Area Label].Caption = "Cross Sectional Area"
    Me![Concrete]![Total Length Label].Caption = "Total Length"
    Me![Concrete]![Embedded Length Label].Caption = "Embedded Length"
    Me![Concrete]![Total Weight Label].Caption = "Total Weight"
    Me![Concrete]![Unit Weight of Concrete Label].Caption = "Unit Weight of Concrete"
    Me![Concrete]![Concrete strength Label].Caption = "Concrete Strength"
    Me![H Steel]![Steel Cross Sectional Area Label].Caption = "Steel Cross Sectional Area"
    Me![H Steel]![Pile Cross Sectional Area Label].Caption = "Pile Cross Sectional Area"
    Me![H Steel]![Steel Section Perimeter Label].Caption = "Steel Section Perimeter"
    Me![H Steel]![Pile  Perimeter Label].Caption = "Pile  Perimeter"
    Me![H Steel]![Weight of Steel  Label].Caption = "Weight of Steel"
    Me![H Steel]![Weight of pile Label].Caption = "Weight of pile"
    Me![H Steel]![Total Length Label].Caption = "Total Length"
    Me![H Steel]![Embedded Length Label].Caption = "Embedded Length"
    Me![H Steel]![Unit Weight of Steel Label].Caption = "Unit Weight of Steel"
    Me![H Steel]![Concrete strength Label].Caption = "Concrete Strength"
    Me![Pipe]![Outside Diameter Label].Caption = "Outside Diameter"
    Me![Pipe]![Thickness of Steel Label].Caption = "Thickness of Steel"
    Me![Pipe]![Steel Cross Sectional Area Label].Caption = "Steel Cross Sectional Area"
    Me![Pipe]![Pile Cross Sectional Area Label].Caption = "Pile Cross Sectional Area"
    Me![Pipe]![Concrete strength Label].Caption = "Concrete Strength"
    Me![Pipe]![Weight of Steel Label].Caption = "Weight of Steel"
    Me![Pipe]![Weight of pile Label].Caption = "Weight of  Pile"
    Me![Pipe]![Total Length Label].Caption = "Total Length"
    Me![Pipe]![Embedded Length Label].Caption = "Embedded Length"
    Me![Pipe]![Unit Weight of Steel Label].Caption = "Unit Weight of Steel"
  End Select
End Sub

'All of the following subprocedures have the same structure, they assign a value
'to the SI units control based on the value input in the Raw data field and on
'the selected set of units
Private Sub Pile_Elastic_Modulus_AfterUpdate()
 'Calls the UnitsOff subprocedure
  UnitsOff
  'Calculates the value of the SI control
  Select Case PileUnits
```

```
    Case 1
      Me![Pile Elastic Modulus (KN/m²)] = Me![Pile Elastic Modulus]
    Case 2
      Me![Pile Elastic Modulus (KN/m²)] = Me![Pile Elastic Modulus] * 6.8948
    End Select
End Sub

Private Sub Capwap_Elastic_Modulus_AfterUpdate()
  UnitsOff
    Select Case PileUnits
    Case 1
      Me![Capwap Elastic Modulus (KN/m²)] = Me![Capwap Elastic Modulus]
    Case 2
      Me![Capwap Elastic Modulus (KN/m²)] = Me![Capwap Elastic Modulus] * 6.8948
    End Select
End Sub

Private Sub Pile_Wave_Speed_AfterUpdate()
    UnitsOff
    Select Case PileUnits
    Case 1
      Me![Pile Wave Speed c (m/s)] = Me![Pile Wave Speed]
    Case 2
      Me![Pile Wave Speed c (m/s)] = Me![Pile Wave Speed] * 0.3048
    End Select
End Sub

Private Sub Pile_Impedance_AfterUpdate()
    UnitsOff
    Select Case PileUnits
    Case 1
      Me![Pile Impedance E*A/c (KN-s/m)] = Me![Pile Impedance]
    Case 2
      Me![Pile Impedance E*A/c (KN-s/m)] = Me![Pile Impedance] * 0.01459
    End Select
End Sub

Private Sub Water_Table_Elev_AfterUpdate()
UnitsOff
    Select Case PileUnits
    Case 1
      Me![Water Table Elev (m)] = Me![Water Table Elev]
    Case 2
      Me![Water Table Elev (m)] = Me![Water Table Elev] * 0.3048
    End Select
End Sub

Private Sub Ground_Surface_Elev_AfterUpdate()
  UnitsOff
    Select Case PileUnits
    Case 1
      Me![Ground Surface Elev (m)] = Me![Ground Surface Elev]
    Case 2
      Me![Ground Surface Elev (m)] = Me![Ground Surface Elev] * 0.3048
    End Select
End Sub

'Moves to the previous record
Private Sub GoBack_Click()
On Error Resume Next
    DoCmd.GoToRecord , , acPrevious
End Sub
```

```
'Moves to the next record
Private Sub GoForward_Click()
On Error Resume Next
   DoCmd.GoToRecord , , acNext
End Sub

'Deletes the current record
Private Sub DeleteRecord_Click()
On Error Resume Next
   DoCmd.RunCommand acCmdDeleteRecord
   DoCmd.GoToRecord , , acPrevious
End Sub

'Starts a new record
Private Sub AddRecord_Click()
On Error Resume Next
   DoCmd.GoToRecord , , acNewRec
End Sub
```

sbfPiles Concrete

```
Option Compare Database
Option Explicit

'All of the following subprocedures have the same structure, they assign a value
'to the SI units control based on the value input in the Raw data field and on
'the selected set of units
Private Sub Void_Diameter_AfterUpdate()
  'Calculates the value of the SI control
  Select Case PileUnits
  Case 1
    Me![Void Diameter (mm)] = Me![Void Diameter]
  Case 2
    Me![Void Diameter (mm)] = Me![Void Diameter] * 25.4
  End Select
End Sub

Public Sub Width_AfterUpdate()
  Select Case PileUnits
  Case 1
    Me![Width (mm)] = Me![Width]
  Case 2
    Me![Width (mm)] = Me![Width] * 25.4
  End Select
  Forms!frmGeneral.[Driven Piles]![Diameter (mm)] = Me![Width (mm)]
End Sub

Public Sub Cross_Sectional_Area_AfterUpdate()
  Select Case PileUnits
  Case 1
    Me![Cross Sectional Area (mm²)] = Me![Cross Sectional Area]
  Case 2
    Me![Cross Sectional Area (mm²)] = Me![Cross Sectional Area] * 645.16
  End Select
End Sub

Public Sub Total_Length_AfterUpdate()
  Select Case PileUnits
  Case 1
    Me![Total Length (m)] = Me![Total Length]
  Case 2
    Me![Total Length (m)] = Me![Total Length] * 0.3048
  End Select
  Forms!frmGeneral.[Driven Piles]![Total Length (m)] = Me![Total Length (m)]
End Sub

Public Sub Embedded_Length_AfterUpdate()
  Select Case PileUnits
  Case 1
    Me![Embedded Length (m)] = Me![Embedded Length]
  Case 2
    Me![Embedded Length (m)] = Me![Embedded Length] * 0.3048
  End Select
  Forms!frmGeneral.[Driven Piles]![Embedded Length (m)] = Me![Embedded Length (m)]
End Sub
```

```
Public Sub Total_Weight_AfterUpdate()
  Select Case PileUnits
  Case 1
    Me![Total Weight (KN)] = Me![Total Weight]
  Case 2
    Me![Total Weight (KN)] = Me![Total Weight] * 8.8964
  End Select
End Sub

Public Sub Unit_Weight_of_Concrete_AfterUpdate()
  Select Case PileUnits
  Case 1
    Me![Unit Weight of Concrete (KN/m^3)] = Me![Unit Weight of Concrete]
  Case 2
    Me![Unit Weight of Concrete (KN/m^3)] = Me![Unit Weight of Concrete] * 0.1571
  End Select
End Sub

Public Sub Concrete_Strength_AfterUpdate()
  Select Case PileUnits
  Case 1
    Me![Concrete Strength (KN/m²)] = Me![Concrete Strength]
  Case 2
    Me![Concrete Strength (KN/m²)] = Me![Concrete Strength] * 6.8948
  End Select
End Sub
```

sbfPiles H

```
Option Compare Database
Option Explicit

'All of the following subprocedures have the same structure, they assign a value
'to the SI units control based on the value input in the Raw data field and on
'the selected set of units
Private Sub Steel_Cross_Sectional_Area_AfterUpdate()
  'Calculates value of the SI control
  Select Case PileUnits
  Case 1
    Me![Steel Cross Sectional Area (mm²)] = Me![Steel Cross Sectional Area]
  Case 2
    Me![Steel Cross Sectional Area (mm²)] = Me![Steel Cross Sectional Area] * 645.16
  End Select
End Sub

Private Sub Pile_Cross_Sectional_Area_AfterUpdate()
  Select Case PileUnits
  Case 1
    Me![Pile Cross Sectional Area (mm²)] = Me![Pile Cross Sectional Area]
  Case 2
    Me![Pile Cross Sectional Area (mm²)] = Me![Pile Cross Sectional Area] * 645.16
  End Select
End Sub

Private Sub Steel_Section_Perimeter_AfterUpdate()
  Select Case PileUnits
  Case 1
    Me![Steel Section Perimeter (mm)] = Me![Steel Section Perimeter]
  Case 2
    Me![Steel Section Perimeter (mm)] = Me![Steel Section Perimeter] * 25.4
  End Select
End Sub

Private Sub Pile__Perimeter_AfterUpdate()
  Select Case PileUnits
  Case 1
    Me![Pile  Perimeter (mm)] = Me![Pile  Perimeter]
  Case 2
    Me![Pile  Perimeter (mm)] = Me![Pile  Perimeter] * 25.4
End Select
End Sub

Private Sub Weight_of_Steel_AfterUpdate()
  Select Case PileUnits
  Case 1
    Me![Weight of Steel (KN/m)] = Me![Weight of Steel]
  Case 2
    Me![Weight of Steel (KN/m)] = Me![Weight of Steel] * 29.1878
  End Select
End Sub

Private Sub Weight_of_pile_AfterUpdate()
  Select Case PileUnits
  Case 1
    Me![Weight of pile (KN)] = Me![Weight of pile]
  Case 2
    Me![Weight of pile (KN)] = Me![Weight of pile] * 8.8964
```

```
    End Select
End Sub

Private Sub Total_Length_AfterUpdate()
    Select Case PileUnits
    Case 1
        Me![Total Length (m)] = Me![Total Length]
    Case 2
        Me![Total Length (m)] = Me![Total Length] * 0.3048
    End Select
    Forms!frmGeneral.[Driven Piles]![Total Length (m)] = Me![Total Length (m)]
End Sub

Private Sub Embedded_Length_AfterUpdate()
    Select Case PileUnits
    Case 1
        Me![Embedded Length (m)] = Me![Embedded Length]
    Case 2
        Me![Embedded Length (m)] = Me![Embedded Length] * 0.3048
    End Select
    Forms!frmGeneral.[Driven Piles]![Embedded Length (m)] = Me![Embedded Length (m)]
End Sub

Private Sub Unit_Weight_of_Steel_AfterUpdate()
    Select Case PileUnits
    Case 1
        Me![Unit Weight of Steel (KN/m^3)] = Me![Unit Weight of Steel]
    Case 2
        Me![Unit Weight of Steel (KN/m^3)] = Me![Unit Weight of Steel] * 0.1571
    End Select
End Sub

Private Sub Concrete_Strength_AfterUpdate()
    Select Case PileUnits
    Case 1
        Me![Concrete Strength (KN/m²)] = Me![Concrete Strength]
    Case 2
        Me![Concrete Strength (KN/m²)] = Me![Concrete Strength] * 6.8948
    End Select
End Sub
```

sbfPiles Pipe

```
Option Compare Database
Option Explicit

'All of the following subprocedures have the same structure, they assign a value
'to the SI units control based on the value input in the Raw data field and on
'the selected set of units
Public Sub Outside_Diameter_AfterUpdate()
  'Calculates value of SI control
  Select Case PileUnits
  Case 1
    Me![Outside Diameter (mm)] = Me![Outside Diameter]
  Case 2
    Me![Outside Diameter (mm)] = Me![Outside Diameter] * 25.4
  End Select
   Forms!frmGeneral.[Driven Piles]![Diameter (mm)] = Me![Outside Diameter (mm)]
End Sub

Public Sub Thickness_of_Steel_AfterUpdate()
  Select Case PileUnits
  Case 1
    Me![Thickness of Steel (mm)] = Me![Thickness of Steel]
  Case 2
    Me![Thickness of Steel (mm)] = Me![Thickness of Steel] * 25.4
  End Select
End Sub

Public Sub Steel_Cross_Sectional_Area_AfterUpdate()
  Select Case PileUnits
  Case 1
    Me![Steel Cross Sectional Area (mm²)] = Me![Steel Cross Sectional Area]
  Case 2
    Me![Steel Cross Sectional Area (mm²)] = Me![Steel Cross Sectional Area] * 645.16
  End Select
End Sub

Public Sub Pile_Cross_Sectional_Area_AfterUpdate()
  Select Case PileUnits
  Case 1
    Me![Pile Cross Sectional Area (mm²)] = Me![Pile Cross Sectional Area]
  Case 2
    Me![Pile Cross Sectional Area (mm²)] = Me![Pile Cross Sectional Area] * 645.16
  End Select
End Sub

Public Sub Concrete_Strength_AfterUpdate()
  Select Case PileUnits
  Case 1
    Me![Concrete Strength (KN/m²)] = Me![Concrete Strength]
  Case 2
    Me![Concrete Strength (KN/m²)] = Me![Concrete Strength] * 6.8948
  End Select
End Sub

Public Sub Weight_of_Steel_AfterUpdate()
  Select Case PileUnits
  Case 1
    Me![Weight of Steel (KN/m)] = Me![Weight of Steel]
```

```
Case 2
    Me![Weight of Steel (KN/m)] = Me![Weight of Steel] * 29.1878
  End Select
End Sub

Public Sub Weight_of_pile_AfterUpdate()
  Select Case PileUnits
  Case 1
    Me![Weight of pile (KN)] = Me![Weight of pile]
  Case 2
    Me![Weight of pile (KN)] = Me![Weight of pile] * 8.8964
  End Select
End Sub

Public Sub Total_Length_AfterUpdate()
  Select Case PileUnits
  Case 1
    Me![Total Length (m)] = Me![Total Length]
  Case 2
    Me![Total Length (m)] = Me![Total Length] * 0.3048
  End Select
  Forms!frmGeneral.[Driven Piles]![Total Length (m)] = Me![Total Length (m)]
End Sub

Public Sub Embedded_Length_AfterUpdate()
  Select Case PileUnits
  Case 1
    Me![Embedded Length (m)] = Me![Embedded Length]
  Case 2
    Me![Embedded Length (m)] = Me![Embedded Length] * 0.3048
  End Select
  Forms!frmGeneral.[Driven Piles]![Embedded Length (m)] = Me![Embedded Length (m)]
End Sub

Public Sub Unit_Weight_of_Steel_AfterUpdate()
  Select Case PileUnits
  Case 1
    Me![Unit Weight of Steel (KN/m^3)] = Me![Unit Weight of Steel]
  Case 2
    Me![Unit Weight of Steel (KN/m^3)] = Me![Unit Weight of Steel] * 0.1571
  End Select
End Sub
```

frmPiles Driving

```
Option Compare Database
Option Explicit
Dim Current_Pile As Integer

Private Sub UnitsOff()
   'Displays a message if no units are selected prompting the user to select a set of units
   If IsNull(Me![Textbox_Units]) Then
      MsgBox "Please specify units"
      Me![Units].SetFocus
      'Erases the value assigned to the control
      DoCmd.RunCommand acCmdUndo
   End If
End Sub

Private Sub Form_Current()
   'Checks if a set of units is selected and updates the Units control
   Select Case Me![Textbox_Units]
   Case "SI"
   Me![Units].Value = 1
   Case "English"
   Me![Units].Value = 2
   Case Else
   Me![Units].Value = 0
   End Select
   'Updates the DrivingUnits variable
   DrivingUnits = Me![Units]
   'Calls the UnitsDriving subprocecure
   UnitsDriving
 Me!Units.SetFocus
End Sub

Private Sub Units_AfterUpdate()
DoCmd.SetWarnings False
 'Updates the Textbox_Units control if units are selected for the first time
 If DrivingUnits = 0 Then
    Select Case Me![Units]
      Case 1
        Me![Textbox_Units].Value = "SI"
      Case 2
        Me![Textbox_Units].Value = "English"
    End Select
    'Updates the DrivingUnits variable
    DrivingUnits = Me![Units]
    'Calls the UnitsDriving subprocedure
    UnitsDriving
    Forms.frmGeneral![Driven Piles]![Driving Data] = "Yes"
 Else
    'Displays a message warning the deletion of the current data if the user attempts to change the units
    If MsgBox("Changing the Units will Erase the current Driving Data", vbOKCancel) = vbOK Then
      'Deletes the record if the user clicks OK
      DoCmd.RunCommand acCmdDeleteRecord
    Else
      'Displays the current units if the user clicks Cancel
      Me![Units] = DrivingUnits
    End If
 End If
DoCmd.SetWarnings True
End Sub
```

```
Private Sub UnitsDriving()
'Changes all the labels to the specified set of units
Select Case DrivingUnits
   Case 1
      Me![Weigh Label].Caption = "Weigh (N)"
      Me![Energy Label].Caption = "Energy (KN-m)"
      Me![Pre Bored Depth Label].Caption = "Pre Bored Depth (m)"
      Me![Last Blow Label].Caption = "Last Blow (mm)"
      Me![End of Driving Label].Caption = "End of Driving (mm)"
      Me![Start of Restrike Label].Caption = "Start of Restrike (mm)"
      Me![Penet Depth Label].Caption = "(m)"
      Me![Penetration Label].Caption = "(Blows/304.8mm)"
   Case 2
      Me![Weigh Label].Caption = "Weigh (lb)"
      Me![Energy Label].Caption = "Energy (lb-ft)"
      Me![Pre Bored Depth Label].Caption = "Pre Bored Depth (ft)"
      Me![Last Blow Label].Caption = "Last Blow (in)"
      Me![End of Driving Label].Caption = "End of Driving (in)"
      Me![Start of Restrike Label].Caption = "Start of Restrike (in)"
      Me![Penet Depth Label].Caption = "(ft)"
      Me![Penetration Label].Caption = "Blows/ft"
   Case Else
      Me![Weigh Label].Caption = "Weigh"
      Me![Energy Label].Caption = "Energy"
      Me![Pre Bored Depth Label].Caption = "Pre Bored Depth"
      Me![Last Blow Label].Caption = "Last Blow"
      Me![End of Driving Label].Caption = "End of Driving"
      Me![Start of Restrike Label].Caption = "Start of Restrike"
      Me![Penet Depth Label].Caption = ""
      Me![Penetration Label].Caption = ""
   End Select
End Sub

'All of the following subprocedures have the same structure, they assign a value
'to the SI units control based on the value input in the Raw data field and on
'the selected set of units
Private Sub Weight_AfterUpdate()
   'Calls the UnitsOff subprocedure
   UnitsOff
   'Calculates the value of the SI control
   Select Case DrivingUnits
   Case 1
      Me![Weight (N)] = Me![Weight]
   Case 2
      Me![Weight (N)] = Me![Weight] * 4.4482
   End Select
End Sub

Private Sub Energy_AfterUpdate()
   UnitsOff
   Select Case DrivingUnits
   Case 1
      Me![Energy (KN-m)] = Me![Energy]
   Case 2
      Me![Energy (KN-m)] = Me![Energy] * 0.0013558
   End Select
End Sub

Private Sub Pre_Bored_Depth_AfterUpdate()
   UnitsOff
   Select Case DrivingUnits
```

```
    Case 1
      Me![Pre Bored Depth (m)] = Me![Pre Bored Depth]
    Case 2
      Me![Pre Bored Depth (m)] = Me![Pre Bored Depth] * 0.3048
    End Select
End Sub

Private Sub Last_Blow_AfterUpdate()
    UnitsOff
    Select Case DrivingUnits
    Case 1
      Me![Last Blow (mm)] = Me![Last Blow]
    Case 2
      Me![Last Blow (mm)] = Me![Last Blow] * 25.4
    End Select
End Sub

Private Sub End_of_Driving_AfterUpdate()
    UnitsOff
    Select Case DrivingUnits
    Case 1
      Me![End of Driving (mm)] = Me![End of Driving]
    Case 2
      Me![End of Driving (mm)] = Me![End of Driving] * 25.4
    End Select
End Sub

Private Sub Start_of_Restrike_AfterUpdate()
    UnitsOff
    Select Case DrivingUnits
    Case 1
      Me![Start of Restrike (mm)] = Me![Start of Restrike]
    Case 2
      Me![Start of Restrike (mm)] = Me![Start of Restrike] * 25.4
    End Select
End Sub

'Prints the pile information
Private Sub PrintButton_Click()
    Print_Units = 0
    Print_Excel = 0
    Current_Pile = Me![Pile ID]
    Report_Name = "Pile Penetration"
    Print_Filter = "[Pile ID] =" & (Current_Pile)
    DoCmd.OpenForm "frmPrint", acNormal
End Sub
```

sbfDriving Penetration

```
Option Compare Database
Option Explicit

'Assign a value to the SI units control based on the value input in the Raw data
'field and on the selected set of units
Private Sub Depth_AfterUpdate()
   'Calculates the value of the SI control
   Select Case DrivingUnits
   Case 1
      Me![Depth (m)] = Me![Depth]
   Case 2
      Me![Depth (m)] = Me![Depth] * 0.3048
   End Select
End Sub
```

frmGRL

```
Option Compare Database
Option Explicit
Dim GRLUnits As Integer
Dim Current_GRL As Integer


Private Sub Form_Current()
   'Updates the value of the Load test current record in the navigation box
   Me!CurrentRecordNumber.Value = Me.CurrentRecord
   'Checks if a set of units is selected and updates the Units control
   Select Case Me![Textbox_Units]
   Case "SI"
   Me![Units].Value = 1
   Case "English"
   Me![Units].Value = 2
   Case Else
   Me![Units].Value = 0
   End Select
   'Updates the GRLUnits variable
   GRLUnits = Me![Units]
   'Calls the UnitsGRL subprocecure
   UnitsGRL
Me!Units.SetFocus
End Sub

Private Sub UnitsOff()
   'Displays a message if no units are selected prompting the user
   'to select a set of units
   If IsNull(Me![Textbox_Units]) Then
      MsgBox "Please specify units"
      Me![Units].SetFocus
      'Erases the value assigned to the control
      DoCmd.RunCommand acCmdUndo
   End If
End Sub

Private Sub Units_AfterUpdate()
DoCmd.SetWarnings False
 'Updates the Textbox_Units control if units are selected for the first time
 If GRLUnits = 0 Then
    Select Case Me![Units]
      Case 1
        Me![Textbox_Units].Value = "SI"
      Case 2
        Me![Textbox_Units].Value = "English"
    End Select
    'Updates the GRLUnits variable
    GRLUnits = Me![Units]
    'Calls the UnitsGRL subprocedure
    UnitsGRL
    Forms.frmGeneral![Driven Piles]![GRLData] = "Yes"
 Else
    'Displays a message warning the deletion of the current data
    'if the user attempts to change the units
    If MsgBox("Changing the Units will Erase the current GRL Data", vbOKCancel) = vbOK Then
      'Deletes the record if the user clicks OK
      DoCmd.RunCommand acCmdDeleteRecord
    Else
```

```
      'Displays the current units if the user clicks Cancel
      Me![Units] = GRLUnits
    End If
  End If
DoCmd.SetWarnings True
End Sub

Private Sub UnitsGRL()
  'Changes all the labels to the specified set of units
  Select Case GRLUnits
  Case 1
    Me![L Label].Caption = "L (m) "
    Me![Le Label].Caption = "Le (m) "
    Me![Area Label].Caption = "Area (m²)"
    Me![c Label].Caption = "c (m/s) "
    Me![Ec Label].Caption = "Ec (GN/m²) "
    Me![EA/c Label].Caption = "EA/c (KN-s/m) "
  Case 2
    Me![L Label].Caption = "L (ft) "
    Me![Le Label].Caption = "Le (ft) "
    Me![Area Label].Caption = "Area (ft²) "
    Me![c Label].Caption = "c (ft/s) "
    Me![Ec Label].Caption = "Ec (Tons/ft²) "
    Me![EA/c Label].Caption = "EA/c (Ton-s/ft) "
  Case Else
    Me![L Label].Caption = "L "
    Me![Le Label].Caption = "Le "
    Me![Area Label].Caption = "Area "
    Me![c Label].Caption = "c "
    Me![Ec Label].Caption = "Ec "
    Me![EA/c Label].Caption = "EA/c "
  End Select
End Sub

'All of the following subprocedures have the same structure, they assign a value
'to the SI units control based on the value input in the Raw data field and on
'the selected set of units
Private Sub L_AfterUpdate()
'Calls the UnitsOff subprocedure
UnitsOff
  'Calculates the value of the SI control
  Select Case GRLUnits
  Case 1
    Me![L (m)] = Me![L]
  Case 2
    Me![L (m)] = Me![L] * 0.3048
  End Select
End Sub

Private Sub Le_AfterUpdate()
UnitsOff
  Select Case GRLUnits
  Case 1
    Me![Le (m)] = Me![Le]
  Case 2
    Me![Le (m)] = Me![Le] * 0.3048
  End Select
End Sub

Private Sub Area_AfterUpdate()
UnitsOff
  Select Case GRLUnits
```

```
  Case 1
    Me![Area (m²)] = Me![Area]
  Case 2
    Me![Area (m²)] = Me![Area] * 0.0929
  End Select
End Sub

Private Sub c_AfterUpdate()
UnitsOff
  Select Case GRLUnits
  Case 1
    Me![c (m/s)] = Me![c]
  Case 2
    Me![c (m/s)] = Me![c] * 0.3048
  End Select
End Sub

Private Sub Ec_AfterUpdate()
UnitsOff
  Select Case GRLUnits
  Case 1
    Me![Ec (GN/m²)] = Me![Ec]
  Case 2
    Me![Ec (GN/m²)] = Me![Ec] * 0.00009576
  End Select
End Sub

Private Sub EA_c_AfterUpdate()
UnitsOff
  Select Case GRLUnits
  Case 1
    Me![EA/c  (KN-s/m)] = Me![EA/c]
  Case 2
    Me![EA/c  (KN-s/m)] = Me![EA/c] * 29.1878
  End Select
End Sub


'Moves to the previous record
Private Sub GoBack_Click()
On Error Resume Next
  DoCmd.GoToRecord , , acPrevious
End Sub

'Moves to the next record
Private Sub GoForward_Click()
On Error Resume Next
  DoCmd.GoToRecord , , acNext
End Sub

'Deletes the current record
Private Sub DeleteRecord_Click()
On Error Resume Next
  DoCmd.RunCommand acCmdDeleteRecord
  DoCmd.GoToRecord , , acPrevious
End Sub

'Starts a new record
Private Sub AddRecord_Click()
On Error Resume Next
  DoCmd.GoToRecord , , acNewRec
End Sub
```

```
'Prints the GRL Data
Private Sub PrintButton_Click()
   Print_Units = 0
   Print_Excel = 0
   Current_GRL = Me![GRL ID]
   Report_Name = "GRL PDA Trace"
   Print_Filter = "[GRL ID] =" & (Current_GRL)
   DoCmd.OpenForm "frmPrint", acNormal
End Sub
```

frmInsitu Tests

```
Option Compare Database
Option Explicit
Dim CurrentTest As String

Private Sub UnitsOff()
   'Displays a message if no units are selected prompting the user
   'to select a set of units
   If IsNull(Me![Textbox_Units]) Then
      MsgBox "Please specify units"
      Me![Units].SetFocus
      'Erases the value assigned to the control
      DoCmd.RunCommand acCmdUndo
   End If
End Sub

Private Sub Form_Current()
   'Calls the ViewInsitu subprocedure
   ViewInsitu
   'Updates the value of the insitu test current record in the navigation box
   Me!CurrentRecordNumber.Value = Me.CurrentRecord
   'Updates the Units control
   Select Case Me![Textbox_Units]
   Case "SI"
   Me![Units].Value = 1
   Case "English"
   Me![Units].Value = 2
   Case Else
   Me![Units].Value = 0
   End Select
   'Updates the InsituUnits variable
   InsituUnits = Me![Units]
   'Calls the UnitsInsitu subprocedure
   UnitsInsitu
End Sub

Private Sub Units_AfterUpdate()
DoCmd.SetWarnings False
 'Updates the Textbox_Units control if units are selected for the first time
 If InsituUnits = 0 Then
    Select Case Me![Units]
      Case 1
        Me![Textbox_Units].Value = "SI"
      Case 2
        Me![Textbox_Units].Value = "English"
    End Select
    'Updates the InsituUnits variable
    InsituUnits = Me![Units]
    'Calls the UnitsInsitu subprocedure
    UnitsInsitu
 Else
    'Displays a message warning the deletion of the current data
    'if the user attempts to change the units
    If MsgBox("Changing the Units will Erase the current Insitu Data Data", vbOKCancel) = vbOK Then
      'Deletes the record if the user clicks OK
      DoCmd.RunCommand acCmdDeleteRecord
    Else
      'Displays the current units if the user clicks Cancel
      Me![Units] = InsituUnits
```

```
    End If
  End If
  DoCmd.SetWarnings True
End Sub

Private Sub Type_of_Test_Change()
DoCmd.SetWarnings False
   'Calls the UnitsOff subprocedure
   UnitsOff
   'Calls the ViewInsitu subprocedure if the Insitu test type is selected for the first time
   If CurrentTest = "No Test" Then
      ViewInsitu
   Else
    'Displays a message warning the deletion of the current data
    'if the user attempts to change the insitu test type
      If MsgBox("Changing the Test type will Erase the current Test Data", vbOKCancel) = vbOK Then
         'Deletes the record if the user clicks OK
         DoCmd.RunCommand acCmdDeleteRecord
      Else
         'Displays the current type of test if the user clicks Cancel
         Me![Type of Test] = CurrentTest
      End If
   End If
DoCmd.SetWarnings True
End Sub

Private Sub ViewInsitu()
   'Resets the visible properties of the tab control
   Me![Box].Visible = True
   Me![CPT].Visible = False
   Me![DMT].Visible = False
   Me![SPT].Visible = False
   Me![PRESSUREMETER].Visible = False
  'Displays only the selected Insitu test tab and Updates the CurrentTest variable
  Select Case Me![Type of Test]
        Case "SPT"
        Me![SPT].Visible = True
        CurrentTest = "SPT"
        Case "CPT"
        Me![CPT].Visible = True
        CurrentTest = "CPT"
        Case "DMT"
        Me![DMT].Visible = True
        CurrentTest = "DMT"
        Case "PMT"
        Me![PRESSUREMETER].Visible = True
        CurrentTest = "PMT"
        Case Else
        Me![Box].Visible = False
        CurrentTest = "No Test"
  End Select
End Sub

Private Sub UnitsInsitu()
   'Changes all the labels to the specified set of units
   Select Case Me![Units]
   Case 1
      Me![Water Table Elev Label].Caption = "Water Table Elev (m)"
      Me![Ground Surface Elev Label].Caption = "Ground Surface Elev (m)"
      Me![Test Depth Label].Caption = "Test Depth (m)"
      Me![SPT Elev Label].Caption = "(m)"
      Me![SPT Depth Label].Caption = "(m)"
```

```
        Me![CPT Elev Label].Caption = "(m)"
        Me![CPT Depth Label].Caption = "(m)"
        Me![Cone Resistance Label].Caption = "(KN/m²)"
        Me![Friction Resistance Label].Caption = "(KN/m²)"
        Me![DMT Elev Label].Caption = "(m)"
        Me![DMT Depth Label].Caption = "(m)"
        Me![PMT Pressure Label].Caption = "(KN/m²)"
        Me![PMT Volume Label].Caption = "(mm^3)"
    Case 2
        Me![Water Table Elev Label].Caption = "Water Table Elev (ft)"
        Me![Ground Surface Elev Label].Caption = "Ground Surface Elev (ft)"
        Me![Test Depth Label].Caption = "Test Depth (ft)"
        Me![SPT Elev Label].Caption = "(ft)"
        Me![SPT Depth Label].Caption = "(ft)"
        Me![CPT Elev Label].Caption = "(ft)"
        Me![CPT Depth Label].Caption = "(ft)"
        Me![Cone Resistance Label].Caption = "(Ton/ft²)"
        Me![Friction Resistance Label].Caption = "(Ton/ft²)"
        Me![DMT Elev Label].Caption = "(ft)"
        Me![DMT Depth Label].Caption = "(ft)"
        Me![PMT Pressure Label].Caption = "(Ton/ft²)"
        Me![PMT Volume Label].Caption = "(in^3)"
    Case Else
        Me![Water Table Elev Label].Caption = "Water Table Elev"
        Me![Ground Surface Elev Label].Caption = "Ground Surface Elev"
        Me![Test Depth Label].Caption = "Test Depth"
        Me![SPT Elev Label].Caption = ""
        Me![SPT Depth Label].Caption = ""
        Me![CPT Elev Label].Caption = ""
        Me![CPT Depth Label].Caption = ""
        Me![Cone Resistance Label].Caption = ""
        Me![Friction Resistance Label].Caption = ""
        Me![DMT Elev Label].Caption = ""
        Me![DMT Depth Label].Caption = ""
        Me![PMT Pressure Label].Caption = ""
        Me![PMT Volume Label].Caption = ""
    End Select
End Sub

'Moves to the previous record
Private Sub GoBack_Click()
On Error Resume Next
    DoCmd.GoToRecord , , acPrevious
End Sub

'Moves to the next record
Private Sub GoForward_Click()
On Error Resume Next
    DoCmd.GoToRecord , , acNext
End Sub

'Deletes the current record
Private Sub DeleteRecord_Click()
On Error Resume Next
    DoCmd.RunCommand acCmdDeleteRecord
    DoCmd.GoToRecord , , acPrevious
End Sub

'Starts a new record
Private Sub AddRecord_Click()
On Error Resume Next
    DoCmd.GoToRecord , , acNewRec
```

```
End Sub

'All of the following subprocedures have the same structure, they assign a value
'to the SI units control based on the value input in the Raw data field and on
'the selected set of units
Private Sub Water_Table_Elev_AfterUpdate()
'Calls the UnitsOff subprocedure
UnitsOff
   'Calculates the value of the SI control
   Select Case InsituUnits
   Case 1
      Me![Water Table Elev (m)] = Me![Water Table Elev]
   Case 2
      Me![Water Table Elev (m)] = Me![Water Table Elev] * 0.3048
   End Select
End Sub


Private Sub Ground_Surface_Elev_AfterUpdate()
   UnitsOff
   Select Case InsituUnits
   Case 1
      Me![Ground Surface Elev (m)] = Me![Ground Surface Elev]
   Case 2
      Me![Ground Surface Elev (m)] = Me![Ground Surface Elev] * 0.3048
   End Select
End Sub

Private Sub Test_Depth_AfterUpdate()
 UnitsOff
   Select Case InsituUnits
   Case 1
      Me![Test Depth (m)] = Me![Test Depth]
   Case 2
      Me![Test Depth (m)] = Me![Test Depth] * 0.3048
   End Select
End Sub
```

sbfSPT

```
Option Compare Database
Option Explicit

'All of the following subprocedures have the same structure, they assign a value
'to the SI units control based on the value input in the Raw data field and on
'the selected set of units
Private Sub Elev_AfterUpdate()
   'Calculates value of SI control
   Select Case InsituUnits
   Case 1
      Me![Elev (m)] = Me![Elev]
   Case 2
      Me![Elev (m)] = Me![Elev] * 0.3048
   End Select
End Sub

Private Sub Depth_AfterUpdate()
 Select Case InsituUnits
   Case 1
      Me![Depth (m)] = Me![Depth]
   Case 2
      Me![Depth (m)] = Me![Depth] * 0.3048
   End Select
End Sub
```

sbfCPT

```
Option Compare Database
Option Explicit

'All of the following subprocedures have the same structure, they assign a value
'to the SI units control based on the value input in the Raw data field and on
'the selected set of units
Private Sub Elev_AfterUpdate()
   'Calculates the value of the SI control
   Select Case InsituUnits
   Case 1
      Me![Elev (m)] = Me![Elev]
   Case 2
      Me![Elev (m)] = Me![Elev] * 0.3048
   End Select
End Sub

Private Sub Depth_AfterUpdate()
 Select Case InsituUnits
   Case 1
      Me![Depth (m)] = Me![Depth]
   Case 2
      Me![Depth (m)] = Me![Depth] * 0.3048
   End Select
End Sub

Private Sub Cone_resistance__qc_AfterUpdate()
   Select Case InsituUnits
   Case 1
      Me![Cone resistance  qc (KN/m²)] = Me![Cone resistance  qc]
   Case 2
      Me![Cone resistance  qc (KN/m²)] = Me![Cone resistance  qc] * 95.7605
   End Select
End Sub

Private Sub Friction_resistance__fs_AfterUpdate()
   Select Case InsituUnits
   Case 1
      Me![Friction resistance  fs (KN/m²)] = Me![Friction resistance  fs]
   Case 2
      Me![Friction resistance  fs (KN/m²)] = Me![Friction resistance  fs] * 95.7605
   End Select
End Sub
```

sbfDMT

```
Option Compare Database
Option Explicit

'All of the following subprocedures have the same structure, they assign a value
'to the SI units control based on the value input in the Raw data field and on
'the selected set of units
Private Sub Elev_AfterUpdate()
   'Calculates the value of the SI control
   Select Case InsituUnits
   Case 1
      Me![Elev (m)] = Me![Elev]
   Case 2
      Me![Elev (m)] = Me![Elev] * 0.3048
   End Select
End Sub

Private Sub Depth_AfterUpdate()
 Select Case InsituUnits
   Case 1
      Me![Depth (m)] = Me![Depth]
   Case 2
      Me![Depth (m)] = Me![Depth] * 0.3048
   End Select
End Sub
```

sbfPMT

```
Option Compare Database
Option Explicit


'All of the following subprocedures have the same structure, they assign a value
'to the SI units control based on the value input in the Raw data field and on
'the selected set of units

Private Sub Pressure_AfterUpdate()
   Select Case InsituUnits
   Case 1
      Me![Pressure (KN/m²)] = Me![Pressure]
   Case 2
      Me![Pressure (KN/m²)] = Me![Pressure] * 95.7605
   End Select
End Sub

Private Sub Volume_AfterUpdate()
 Select Case InsituUnits
   Case 1
      Me![Volume (mm^3)] = Me![Volume]
   Case 2
      Me![Volume (mm^3)] = Me![Volume] * 16387.064
   End Select
End Sub
```

frmLoad Tests

```
Option Compare Database
Option Explicit
Dim CurrentTest As String

Private Sub UnitsOff()
   'Displays a message if no units are selected prompting the user to select a set of units
   If IsNull(Me![Textbox_Units]) Then
      MsgBox "Please specify units"
      Me![Units].SetFocus
      'Erases the value assigned to the control
      DoCmd.RunCommand acCmdUndo
      ViewLT
   End If
End Sub

Private Sub Form_Current()
   'Calls the ViewLT subprocedure
   ViewLT
   'Updates the value of the Load test current record in the navigation box
   Me!CurrentRecordNumber.Value = Me.CurrentRecord
   'Updates the Units control
   Select Case Me![Textbox_Units]
   Case "SI"
   Me![Units].Value = 1
   Case "English"
   Me![Units].Value = 2
   Case Else
   Me![Units].Value = 0
   End Select
   'Updates the LoadTestUnits variable
   LoadTestUnits = Me![Units]
   'Calls the UnitsLT subprocedure
   UnitsLT
 Me!Units.SetFocus
End Sub

Private Sub Units_AfterUpdate()
DoCmd.SetWarnings False
 'Updates the Textbox_Units control if units are selected for the first time
 If LoadTestUnits = 0 Then
    Select Case Me![Units]
      Case 1
        Me![Textbox_Units].Value = "SI"
      Case 2
        Me![Textbox_Units].Value = "English"
    End Select
    'Updates the LoadtestUnits variable
    LoadTestUnits = Me![Units]
    'Calls the UnitsLT subprocedure
    UnitsLT
    'Saves the Project ID to be used in the load test query
   Me![Project ID] = CurrentProjectID
 Else
    'Displays a message warning the deletion of the current data
    'if the user attempts to change the units
    If MsgBox("Changing the Units will Erase the current Load Test Data", vbOKCancel) = vbOK Then
      'Deletes the record if the user clicks OK
      DoCmd.RunCommand acCmdDeleteRecord
```

```
    Else
      'Displays the current units if the user clicks Cancel
      Me![Units] = LoadTestUnits
    End If
  End If
DoCmd.SetWarnings True
End Sub

Private Sub Type_of_Test_Change()
DoCmd.SetWarnings False
  'Calls the UnitsOff subprocedure
  UnitsOff
    'Calls the ViewLT subprocedure if the Insitu test type is selected for the first time
  If CurrentTest = "No Test" Then
    ViewLT
    If CurrentTest = "Osterberg Test" Then
    Me!Osterberg![Load test ID] = Me![Load test ID]
    ElseIf CurrentTest = "Statnamic Test" Then
    Me!Statnamic![Load test ID] = Me![Load test ID]
    End If
  Else
    'Displays a message warning the deletion of the current data
    'if the user attempts to change the insitu test type
    If MsgBox("Changing the Test type will Erase the current Test Data", vbOKCancel) = vbOK Then
      'Deletes the record if the user clicks OK
      DoCmd.RunCommand acCmdDeleteRecord
    Else
      'Displays the current type of test if the user clicks Cancel
      Me![Type of Test] = CurrentTest
    End If
  End If
DoCmd.SetWarnings True
Me.[Date of Load Test].SetFocus
End Sub

Private Sub ViewLT()
  'Resets the visible properties of the tab control
  Me![Box].Visible = True
  Me![Statnamic test].Visible = False
  Me![Static test].Visible = False
  Me![Osterberg test].Visible = False
 'Displays only the selected Load test tab and Updates the CurrentTest variable
  Select Case Me![Type of Test]
      Case "Static Test"
      Me![Static test].Visible = True
      CurrentTest = "Static Test"
      Case "Statnamic Test"
      Me![Statnamic test].Visible = True
      CurrentTest = "Statnamic Test"
      Case "Osterberg Test"
      Me![Osterberg test].Visible = True
      CurrentTest = "Osterberg Test"
      Case Else
      Me![Box].Visible = False
      CurrentTest = "No Test"
  End Select
End Sub

Private Sub UnitsLT()
  'Changes all the labels to the specified set of units
  Select Case LoadTestUnits
  Case 1
```

```
    Me![Static Time Label].Caption = "(min)"
    Me![Static Force Label].Caption = "(KN)"
    Me![Static Displacement Label].Caption = "(mm)"
    Me![Osterberg]![Load Label].Caption = "Load (KN)"
    Me![Osterberg]![Displacement Label].Caption = "Displacement of Cell (mm)"
    Me![Osterberg]![Skin Friction Label].Caption = "Unit Skin Friction (KN/m²)"
    Me![Statnamic]![Load Label].Caption = "Load (KN)"
    Me![Statnamic]![Equiv Load Label].Caption = "Load (KN)"
    Me![Statnamic]![Displacement Label].Caption = "Top of Element (mm)"
    Me![Statnamic]![Skin Friction Label].Caption = "Unit Skin Friction (KN/m²)"
  Case 2
    Me![Static Time Label].Caption = "(min)"
    Me![Static Force Label].Caption = "(Ton)"
    Me![Static Displacement Label].Caption = "(in)"
    Me![Osterberg]![Load Label].Caption = "Load (tons)"
    Me![Osterberg]![Displacement Label].Caption = "Displacement of Cell (in)"
    Me![Osterberg]![Skin Friction Label].Caption = "Unit Skin Friction (tsf)"
    Me![Statnamic]![Load Label].Caption = "Load (tons)"
    Me![Statnamic]![Equiv Load Label].Caption = "Load (tons)"
    Me![Statnamic]![Displacement Label].Caption = "Top of Element (in)"
    Me![Statnamic]![Skin Friction Label].Caption = "Unit Skin Friction (tsf)"
  Case Else
    Me![Static Time Label].Caption = ""
    Me![Static Force Label].Caption = ""
    Me![Static Displacement Label].Caption = ""
    Me![Osterberg]![Load Label].Caption = "Load"
    Me![Osterberg]![Displacement Label].Caption = "Displacement of Cell"
    Me![Osterberg]![Skin Friction Label].Caption = "Unit Skin Friction"
    Me![Statnamic]![Load Label].Caption = "Load"
    Me![Statnamic]![Equiv Load Label].Caption = "Load"
    Me![Statnamic]![Displacement Label].Caption = "Top of Element"
    Me![Statnamic]![Skin Friction Label].Caption = "Unit Skin Friction"
  End Select
End Sub

'Moves to the previous record
Private Sub GoBack_Click()
On Error Resume Next
  DoCmd.GoToRecord , , acPrevious
End Sub

'Moves to the next record
Private Sub GoForward_Click()
On Error Resume Next
  DoCmd.GoToRecord , , acNext
End Sub

'Deletes the current record
Private Sub DeleteRecord_Click()
On Error Resume Next
  DoCmd.RunCommand acCmdDeleteRecord
  DoCmd.GoToRecord , , acPrevious
End Sub

'Starts a new record
Private Sub AddRecord_Click()
On Error Resume Next
  DoCmd.GoToRecord , , acNewRec
End Sub
```

sbfStatic

```
Option Compare Database
Option Explicit

'All of the following subprocedures have the same structure, they assign a value
'to the SI units control based on the value input in the Raw data field and on
'the selected set of units
Private Sub Force_AfterUpdate()
   'Calculates the value of the SI control
   Select Case LoadTestUnits
   Case 1
      Me![Force (KN)] = Me![Force]
   Case 2
      Me![Force (KN)] = Me![Force] * 8.8964
   End Select
End Sub

Private Sub Displacement_AfterUpdate()
   Select Case LoadTestUnits
   Case 1
      Me![Displacement (mm)] = Me![Displacement]
   Case 2
      Me![Displacement (mm)] = Me![Displacement] * 25.4
   End Select
End Sub
```

sbfStatnamic

```
Option Compare Database

'All of the following subprocedures have the same structure, they assign a value
'to the SI units control based on the value input in the Raw data field and on
'the selected set of units
Private Sub Bottom_1_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Bottom 1 (m)] = Me![Bottom 1]
  Case 2
    Me![Bottom 1 (m)] = Me![Bottom 1] * 0.3048
  End Select
End Sub

Private Sub Bottom_10_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Bottom 10 (m)] = Me![Bottom 10]
  Case 2
    Me![Bottom 10 (m)] = Me![Bottom 10] * 0.3048
  End Select
End Sub

Private Sub Bottom_11_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Bottom 11 (m)] = Me![Bottom 11]
  Case 2
    Me![Bottom 11 (m)] = Me![Bottom 11] * 0.3048
  End Select
End Sub

Private Sub Bottom_12_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Bottom 12 (m)] = Me![Bottom 12]
  Case 2
    Me![Bottom 12 (m)] = Me![Bottom 12] * 0.3048
  End Select
End Sub

Private Sub Bottom_13_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Bottom 13 (m)] = Me![Bottom 13]
  Case 2
    Me![Bottom 13 (m)] = Me![Bottom 13] * 0.3048
  End Select
End Sub

Private Sub Bottom_14_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Bottom 14 (m)] = Me![Bottom 14]
  Case 2
    Me![Bottom 14 (m)] = Me![Bottom 14] * 0.3048
  End Select
End Sub
```

```
Private Sub Bottom_15_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Bottom 15 (m)] = Me![Bottom 15]
  Case 2
    Me![Bottom 15 (m)] = Me![Bottom 15] * 0.3048
  End Select
End Sub

Private Sub Bottom_2_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Bottom 2 (m)] = Me![Bottom 2]
  Case 2
    Me![Bottom 2 (m)] = Me![Bottom 2] * 0.3048
  End Select
End Sub

Private Sub Bottom_3_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Bottom 3 (m)] = Me![Bottom 3]
  Case 2
    Me![Bottom 3 (m)] = Me![Bottom 3] * 0.3048
  End Select
End Sub

Private Sub Bottom_4_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Bottom 4 (m)] = Me![Bottom 4]
  Case 2
    Me![Bottom 4 (m)] = Me![Bottom 4] * 0.3048
  End Select
End Sub

Private Sub Bottom_5_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Bottom 5 (m)] = Me![Bottom 5]
  Case 2
    Me![Bottom 5 (m)] = Me![Bottom 5] * 0.3048
  End Select
End Sub

Private Sub Bottom_6_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Bottom 6 (m)] = Me![Bottom 6]
  Case 2
    Me![Bottom 6 (m)] = Me![Bottom 6] * 0.3048
  End Select
End Sub

Private Sub Bottom_7_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Bottom 7 (m)] = Me![Bottom 7]
  Case 2
    Me![Bottom 7 (m)] = Me![Bottom 7] * 0.3048
  End Select
```

```
End Sub

Private Sub Bottom_8_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Bottom 8 (m)] = Me![Bottom 8]
  Case 2
    Me![Bottom 8 (m)] = Me![Bottom 8] * 0.3048
  End Select
End Sub

Private Sub Bottom_9_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Bottom 9 (m)] = Me![Bottom 9]
  Case 2
    Me![Bottom 9 (m)] = Me![Bottom 9] * 0.3048
  End Select
End Sub

Private Sub Form_GotFocus()
DoCmd.DoMenuItem acFormBar, acRecordsMenu, acSaveRecord, , acMenuVer70
End Sub

Private Sub Top_1_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Top 1 (m)] = Me![Top 1]
  Case 2
    Me![Top 1 (m)] = Me![Top 1] * 0.3048
  End Select
End Sub

Private Sub Top_10_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Top 10 (m)] = Me![Top 10]
  Case 2
    Me![Top 10 (m)] = Me![Top 10] * 0.3048
  End Select
End Sub

Private Sub Top_11_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Top 11 (m)] = Me![Top 11]
  Case 2
    Me![Top 11 (m)] = Me![Top 11] * 0.3048
  End Select
End Sub

Private Sub Top_12_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Top 12 (m)] = Me![Top 12]
  Case 2
    Me![Top 12 (m)] = Me![Top 12] * 0.3048
  End Select
End Sub

Private Sub Top_13_AfterUpdate()
Select Case LoadTestUnits
```

```
    Case 1
      Me![Top 13 (m)] = Me![Top 13]
    Case 2
      Me![Top 13 (m)] = Me![Top 13] * 0.3048
    End Select
End Sub

Private Sub Top_14_AfterUpdate()
Select Case LoadTestUnits
    Case 1
      Me![Top 14 (m)] = Me![Top 14]
    Case 2
      Me![Top 14 (m)] = Me![Top 14] * 0.3048
    End Select
End Sub

Private Sub Top_15_AfterUpdate()
Select Case LoadTestUnits
    Case 1
      Me![Top 15 (m)] = Me![Top 15]
    Case 2
      Me![Top 15 (m)] = Me![Top 15] * 0.3048
    End Select
End Sub

Private Sub Top_2_AfterUpdate()
Select Case LoadTestUnits
    Case 1
      Me![Top 2 (m)] = Me![Top 2]
    Case 2
      Me![Top 2 (m)] = Me![Top 2] * 0.3048
    End Select
End Sub

Private Sub Top_3_AfterUpdate()
Select Case LoadTestUnits
    Case 1
      Me![Top 3 (m)] = Me![Top 3]
    Case 2
      Me![Top 3 (m)] = Me![Top 3] * 0.3048
    End Select
End Sub

Private Sub Top_4_AfterUpdate()
Select Case LoadTestUnits
    Case 1
      Me![Top 4 (m)] = Me![Top 4]
    Case 2
      Me![Top 4 (m)] = Me![Top 4] * 0.3048
    End Select
End Sub

Private Sub Top_5_AfterUpdate()
Select Case LoadTestUnits
    Case 1
      Me![Top 5 (m)] = Me![Top 5]
    Case 2
      Me![Top 5 (m)] = Me![Top 5] * 0.3048
    End Select
End Sub

Private Sub Top_6_AfterUpdate()
```

```
Select Case LoadTestUnits
   Case 1
      Me![Top 6 (m)] = Me![Top 6]
   Case 2
      Me![Top 6 (m)] = Me![Top 6] * 0.3048
   End Select
End Sub

Private Sub Top_7_AfterUpdate()
Select Case LoadTestUnits
   Case 1
      Me![Top 7 (m)] = Me![Top 7]
   Case 2
      Me![Top 7 (m)] = Me![Top 7] * 0.3048
   End Select
End Sub

Private Sub Top_8_AfterUpdate()
Select Case LoadTestUnits
   Case 1
      Me![Top 8 (m)] = Me![Top 8]
   Case 2
      Me![Top 8 (m)] = Me![Top 8] * 0.3048
   End Select
End Sub

Private Sub Top_9_AfterUpdate()
Select Case LoadTestUnits
   Case 1
      Me![Top 9 (m)] = Me![Top 9]
   Case 2
      Me![Top 9 (m)] = Me![Top 9] * 0.3048
   End Select
End Sub
```

sbfStatnamic Data

```
Option Compare Database

'All of the following subprocedures have the same structure, they assign a value
'to the SI units control based on the value input in the Raw data field and on
'the selected set of units
Private Sub Displacement_of_top_of_Element_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Displacement of top of Element (mm) ] = Me![Displacement of top of Element]
  Case 2
    Me![Displacement of top of Element (mm) ] = Me![Displacement of top of Element] * 25.4
  End Select
End Sub

Private Sub Equivalent_Static_Load_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Equivalent Static Load (KN)] = Me![Equivalent Static Load]
  Case 2
    Me![Equivalent Static Load (KN)] = Me![Equivalent Static Load] * 8.8964
  End Select
End Sub

Private Sub Load_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Load (KN)] = Me![Load]
  Case 2
    Me![Load (KN)] = Me![Load] * 8.8964
  End Select
End Sub

Private Sub Unit_Skin_Friction_1_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Unit Skin Friction 1 (KN/m²)] = Me![Unit Skin Friction 1]
  Case 2
    Me![Unit Skin Friction 1 (KN/m²)] = Me![Unit Skin Friction 1] * 95.7605
  End Select
End Sub

Private Sub Unit_Skin_Friction_10_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Unit Skin Friction 10 (KN/m²)] = Me![Unit Skin Friction 10]
  Case 2
    Me![Unit Skin Friction 10 (KN/m²)] = Me![Unit Skin Friction 10] * 95.7605
  End Select
End Sub

Private Sub Unit_Skin_Friction_11_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Unit Skin Friction 11 (KN/m²)] = Me![Unit Skin Friction 11]
  Case 2
    Me![Unit Skin Friction 11 (KN/m²)] = Me![Unit Skin Friction 11] * 95.7605
  End Select
End Sub
```

```
Private Sub Unit_Skin_Friction_12_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Unit Skin Friction 12 (KN/m²)] = Me![Unit Skin Friction 12]
  Case 2
    Me![Unit Skin Friction 12 (KN/m²)] = Me![Unit Skin Friction 12] * 95.7605
  End Select
End Sub

Private Sub Unit_Skin_Friction_13_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Unit Skin Friction 13 (KN/m²)] = Me![Unit Skin Friction 13]
  Case 2
    Me![Unit Skin Friction 13 (KN/m²)] = Me![Unit Skin Friction 13] * 95.7605
  End Select
End Sub

Private Sub Unit_Skin_Friction_14_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Unit Skin Friction 14 (KN/m²)] = Me![Unit Skin Friction 14]
  Case 2
    Me![Unit Skin Friction 14 (KN/m²)] = Me![Unit Skin Friction 14] * 95.7605
  End Select
End Sub

Private Sub Unit_Skin_Friction_15_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Unit Skin Friction 15 (KN/m²)] = Me![Unit Skin Friction 15]
  Case 2
    Me![Unit Skin Friction 15 (KN/m²)] = Me![Unit Skin Friction 15] * 95.7605
  End Select
End Sub

Private Sub Unit_Skin_Friction_2_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Unit Skin Friction 2 (KN/m²)] = Me![Unit Skin Friction 2]
  Case 2
    Me![Unit Skin Friction 2 (KN/m²)] = Me![Unit Skin Friction 2] * 95.7605
  End Select
End Sub

Private Sub Unit_Skin_Friction_3_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Unit Skin Friction 3 (KN/m²)] = Me![Unit Skin Friction 3]
  Case 2
    Me![Unit Skin Friction 3 (KN/m²)] = Me![Unit Skin Friction 3] * 95.7605
  End Select
End Sub

Private Sub Unit_Skin_Friction_4_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Unit Skin Friction 4 (KN/m²)] = Me![Unit Skin Friction 4]
  Case 2
    Me![Unit Skin Friction 4 (KN/m²)] = Me![Unit Skin Friction 4] * 95.7605
  End Select
```

```
End Sub

Private Sub Unit_Skin_Friction_5_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Unit Skin Friction 5 (KN/m²)] = Me![Unit Skin Friction 5]
  Case 2
    Me![Unit Skin Friction 5 (KN/m²)] = Me![Unit Skin Friction 5] * 95.7605
  End Select
End Sub

Private Sub Unit_Skin_Friction_6_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Unit Skin Friction 6 (KN/m²)] = Me![Unit Skin Friction 6]
  Case 2
    Me![Unit Skin Friction 6 (KN/m²)] = Me![Unit Skin Friction 6] * 95.7605
  End Select
End Sub

Private Sub Unit_Skin_Friction_7_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Unit Skin Friction 7 (KN/m²)] = Me![Unit Skin Friction 7]
  Case 2
    Me![Unit Skin Friction 7 (KN/m²)] = Me![Unit Skin Friction 7] * 95.7605
  End Select
End Sub

Private Sub Unit_Skin_Friction_8_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Unit Skin Friction 8 (KN/m²)] = Me![Unit Skin Friction 8]
  Case 2
    Me![Unit Skin Friction 8 (KN/m²)] = Me![Unit Skin Friction 8] * 95.7605
  End Select
End Sub

Private Sub Unit_Skin_Friction_9_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Unit Skin Friction 9 (KN/m²)] = Me![Unit Skin Friction 9]
  Case 2
    Me![Unit Skin Friction 9 (KN/m²)] = Me![Unit Skin Friction 9] * 95.7605
  End Select
End Sub
```

sbfOsterberg

```
Option Compare Database

'All of the following subprocedures have the same structure, they assign a value
'to the SI units control based on the value input in the Raw data field and on
'the selected set of units
Private Sub Bottom_1_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Bottom 1 (m)] = Me![Bottom 1]
  Case 2
    Me![Bottom 1 (m)] = Me![Bottom 1] * 0.3048
  End Select
End Sub

Private Sub Bottom_10_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Bottom 10 (m)] = Me![Bottom 10]
  Case 2
    Me![Bottom 10 (m)] = Me![Bottom 10] * 0.3048
  End Select
End Sub

Private Sub Bottom_11_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Bottom 11 (m)] = Me![Bottom 11]
  Case 2
    Me![Bottom 11 (m)] = Me![Bottom 11] * 0.3048
  End Select
End Sub

Private Sub Bottom_12_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Bottom 12 (m)] = Me![Bottom 12]
  Case 2
    Me![Bottom 12 (m)] = Me![Bottom 12] * 0.3048
  End Select
End Sub

Private Sub Bottom_13_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Bottom 13 (m)] = Me![Bottom 13]
  Case 2
    Me![Bottom 13 (m)] = Me![Bottom 13] * 0.3048
  End Select
End Sub

Private Sub Bottom_14_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Bottom 14 (m)] = Me![Bottom 14]
  Case 2
    Me![Bottom 14 (m)] = Me![Bottom 14] * 0.3048
  End Select
End Sub
```

```
Private Sub Bottom_15_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Bottom 15 (m)] = Me![Bottom 15]
  Case 2
    Me![Bottom 15 (m)] = Me![Bottom 15] * 0.3048
  End Select
End Sub

Private Sub Bottom_2_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Bottom 2 (m)] = Me![Bottom 2]
  Case 2
    Me![Bottom 2 (m)] = Me![Bottom 2] * 0.3048
  End Select
End Sub

Private Sub Bottom_3_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Bottom 3 (m)] = Me![Bottom 3]
  Case 2
    Me![Bottom 3 (m)] = Me![Bottom 3] * 0.3048
  End Select
End Sub

Private Sub Bottom_4_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Bottom 4 (m)] = Me![Bottom 4]
  Case 2
    Me![Bottom 4 (m)] = Me![Bottom 4] * 0.3048
  End Select
End Sub

Private Sub Bottom_5_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Bottom 5 (m)] = Me![Bottom 5]
  Case 2
    Me![Bottom 5 (m)] = Me![Bottom 5] * 0.3048
  End Select
End Sub

Private Sub Bottom_6_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Bottom 6 (m)] = Me![Bottom 6]
  Case 2
    Me![Bottom 6 (m)] = Me![Bottom 6] * 0.3048
  End Select
End Sub

Private Sub Bottom_7_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Bottom 7 (m)] = Me![Bottom 7]
  Case 2
    Me![Bottom 7 (m)] = Me![Bottom 7] * 0.3048
  End Select
End Sub
```

```
Private Sub Bottom_8_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Bottom 8 (m)] = Me![Bottom 8]
  Case 2
    Me![Bottom 8 (m)] = Me![Bottom 8] * 0.3048
  End Select
End Sub

Private Sub Bottom_9_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Bottom 9 (m)] = Me![Bottom 9]
  Case 2
    Me![Bottom 9 (m)] = Me![Bottom 9] * 0.3048
  End Select
End Sub

Private Sub Form_GotFocus()
DoCmd.DoMenuItem acFormBar, acRecordsMenu, acSaveRecord, , acMenuVer70
End Sub

Private Sub Top_1_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Top 1 (m)] = Me![Top 1]
  Case 2
    Me![Top 1 (m)] = Me![Top 1] * 0.3048
  End Select
End Sub

Private Sub Top_10_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Top 10 (m)] = Me![Top 10]
  Case 2
    Me![Top 10 (m)] = Me![Top 10] * 0.3048
  End Select
End Sub

Private Sub Top_11_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Top 11 (m)] = Me![Top 11]
  Case 2
    Me![Top 11 (m)] = Me![Top 11] * 0.3048
  End Select
End Sub

Private Sub Top_12_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Top 12 (m)] = Me![Top 12]
  Case 2
    Me![Top 12 (m)] = Me![Top 12] * 0.3048
  End Select
End Sub

Private Sub Top_13_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Top 13 (m)] = Me![Top 13]
```

```
    Case 2
      Me![Top 13 (m)] = Me![Top 13] * 0.3048
    End Select
End Sub

Private Sub Top_14_AfterUpdate()
Select Case LoadTestUnits
    Case 1
      Me![Top 14 (m)] = Me![Top 14]
    Case 2
      Me![Top 14 (m)] = Me![Top 14] * 0.3048
    End Select
End Sub

Private Sub Top_15_AfterUpdate()
Select Case LoadTestUnits
    Case 1
      Me![Top 15 (m)] = Me![Top 15]
    Case 2
      Me![Top 15 (m)] = Me![Top 15] * 0.3048
    End Select
End Sub

Private Sub Top_2_AfterUpdate()
Select Case LoadTestUnits
    Case 1
      Me![Top 2 (m)] = Me![Top 2]
    Case 2
      Me![Top 2 (m)] = Me![Top 2] * 0.3048
    End Select
End Sub

Private Sub Top_3_AfterUpdate()
Select Case LoadTestUnits
    Case 1
      Me![Top 3 (m)] = Me![Top 3]
    Case 2
      Me![Top 3 (m)] = Me![Top 3] * 0.3048
    End Select
End Sub

Private Sub Top_4_AfterUpdate()
Select Case LoadTestUnits
    Case 1
      Me![Top 4 (m)] = Me![Top 4]
    Case 2
      Me![Top 4 (m)] = Me![Top 4] * 0.3048
    End Select
End Sub

Private Sub Top_5_AfterUpdate()
Select Case LoadTestUnits
    Case 1
      Me![Top 5 (m)] = Me![Top 5]
    Case 2
      Me![Top 5 (m)] = Me![Top 5] * 0.3048
    End Select
End Sub

Private Sub Top_6_AfterUpdate()
Select Case LoadTestUnits
    Case 1
```

```
    Me![Top 6 (m)] = Me![Top 6]
  Case 2
    Me![Top 6 (m)] = Me![Top 6] * 0.3048
  End Select
End Sub

Private Sub Top_7_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Top 7 (m)] = Me![Top 7]
  Case 2
    Me![Top 7 (m)] = Me![Top 7] * 0.3048
  End Select
End Sub

Private Sub Top_8_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Top 8 (m)] = Me![Top 8]
  Case 2
    Me![Top 8 (m)] = Me![Top 8] * 0.3048
  End Select
End Sub

Private Sub Top_9_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Top 9 (m)] = Me![Top 9]
  Case 2
    Me![Top 9 (m)] = Me![Top 9] * 0.3048
  End Select
End Sub
```

sbfOsterberg Data

```
Option Compare Database

'All of the following subprocedures have the same structure, they assign a value
'to the SI units control based on the value input in the Raw data field and on
'the selected set of units
Private Sub Displacement_of_bottom_of_O_C_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Displacement of bottom of O-C (mm) ] = Me![Displacement of bottom of O-C]
  Case 2
    Me![Displacement of bottom of O-C (mm) ] = Me![Displacement of bottom of O-C] * 25.4
  End Select
End Sub

Private Sub Displacement_of_top_of_O_C_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Displacement of top of O-C (mm) ] = Me![Displacement of top of O-C]
  Case 2
    Me![Displacement of top of O-C (mm) ] = Me![Displacement of top of O-C] * 25.4
  End Select
End Sub

Private Sub Load_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Load (KN)] = Me![Load]
  Case 2
    Me![Load (KN)] = Me![Load] * 8.8964
  End Select
End Sub

Private Sub Unit_Skin_Friction_1_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Unit Skin Friction 1 (KN/m²)] = Me![Unit Skin Friction 1]
  Case 2
    Me![Unit Skin Friction 1 (KN/m²)] = Me![Unit Skin Friction 1] * 95.7605
  End Select
End Sub

Private Sub Unit_Skin_Friction_10_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Unit Skin Friction 10 (KN/m²)] = Me![Unit Skin Friction 10]
  Case 2
    Me![Unit Skin Friction 10 (KN/m²)] = Me![Unit Skin Friction 10] * 95.7605
  End Select
End Sub

Private Sub Unit_Skin_Friction_11_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Unit Skin Friction 11 (KN/m²)] = Me![Unit Skin Friction 11]
  Case 2
    Me![Unit Skin Friction 11 (KN/m²)] = Me![Unit Skin Friction 11] * 95.7605
  End Select
End Sub
```

```
Private Sub Unit_Skin_Friction_12_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Unit Skin Friction 12 (KN/m²)] = Me![Unit Skin Friction 12]
  Case 2
    Me![Unit Skin Friction 12 (KN/m²)] = Me![Unit Skin Friction 12] * 95.7605
  End Select
End Sub

Private Sub Unit_Skin_Friction_13_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Unit Skin Friction 13 (KN/m²)] = Me![Unit Skin Friction 13]
  Case 2
    Me![Unit Skin Friction 13 (KN/m²)] = Me![Unit Skin Friction 13] * 95.7605
  End Select
End Sub

Private Sub Unit_Skin_Friction_14_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Unit Skin Friction 14 (KN/m²)] = Me![Unit Skin Friction 14]
  Case 2
    Me![Unit Skin Friction 14 (KN/m²)] = Me![Unit Skin Friction 14] * 95.7605
  End Select
End Sub

Private Sub Unit_Skin_Friction_15_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Unit Skin Friction 15 (KN/m²)] = Me![Unit Skin Friction 15]
  Case 2
    Me![Unit Skin Friction 15 (KN/m²)] = Me![Unit Skin Friction 15] * 95.7605
  End Select
End Sub

Private Sub Unit_Skin_Friction_2_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Unit Skin Friction 2 (KN/m²)] = Me![Unit Skin Friction 2]
  Case 2
    Me![Unit Skin Friction 2 (KN/m²)] = Me![Unit Skin Friction 2] * 95.7605
  End Select
End Sub

Private Sub Unit_Skin_Friction_3_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Unit Skin Friction 3 (KN/m²)] = Me![Unit Skin Friction 3]
  Case 2
    Me![Unit Skin Friction 3 (KN/m²)] = Me![Unit Skin Friction 3] * 95.7605
  End Select
End Sub

Private Sub Unit_Skin_Friction_4_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Unit Skin Friction 4 (KN/m²)] = Me![Unit Skin Friction 4]
  Case 2
    Me![Unit Skin Friction 4 (KN/m²)] = Me![Unit Skin Friction 4] * 95.7605
  End Select
End Sub
```

```
Private Sub Unit_Skin_Friction_5_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Unit Skin Friction 5 (KN/m²)] = Me![Unit Skin Friction 5]
  Case 2
    Me![Unit Skin Friction 5 (KN/m²)] = Me![Unit Skin Friction 5] * 95.7605
  End Select
End Sub

Private Sub Unit_Skin_Friction_6_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Unit Skin Friction 6 (KN/m²)] = Me![Unit Skin Friction 6]
  Case 2
    Me![Unit Skin Friction 6 (KN/m²)] = Me![Unit Skin Friction 6] * 95.7605
  End Select
End Sub

Private Sub Unit_Skin_Friction_7_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Unit Skin Friction 7 (KN/m²)] = Me![Unit Skin Friction 7]
  Case 2
    Me![Unit Skin Friction 7 (KN/m²)] = Me![Unit Skin Friction 7] * 95.7605
  End Select
End Sub

Private Sub Unit_Skin_Friction_8_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Unit Skin Friction 8 (KN/m²)] = Me![Unit Skin Friction 8]
  Case 2
    Me![Unit Skin Friction 8 (KN/m²)] = Me![Unit Skin Friction 8] * 95.7605
  End Select
End Sub

Private Sub Unit_Skin_Friction_9_AfterUpdate()
Select Case LoadTestUnits
  Case 1
    Me![Unit Skin Friction 9 (KN/m²)] = Me![Unit Skin Friction 9]
  Case 2
    Me![Unit Skin Friction 9 (KN/m²)] = Me![Unit Skin Friction 9] * 95.7605
  End Select
End Sub
```

View_General

```
Option Compare Database
Option Explicit
Dim Current_Project As Integer


Private Sub Form_Current()
   Me!Box.Enabled = True
   DoCmd.Maximize
   'Updates the value of the current record on the navigation box
   Me![CurrentRecordNumber].Value = Me.CurrentRecord
   Me![Project Name].SetFocus
  On Error Resume Next
'Disables all editing controls - Editing is only allow while at frmGeneral
Me!DeleteRecord.Enabled = False
Me!AddRecord.Enabled = False
Me![Driven Piles]!DeleteRecord.Enabled = False
Me![Driven Piles]!AddRecord.Enabled = False
Me![Driven Piles]![GRL]!DeleteRecord.Enabled = False
Me![Driven Piles]![GRL]!AddRecord.Enabled = False
Me![Driven Piles]![Pile Load Tests]!DeleteRecord.Enabled = False
Me![Driven Piles]![Pile Load Tests]!AddRecord.Enabled = False
Me![Drilled Shafts]!DeleteRecord.Enabled = False
Me![Drilled Shafts]!AddRecord.Enabled = False
Me![Drilled Shafts]![Load Test]!DeleteRecord.Enabled = False
Me![Drilled Shafts]![Load Test]!AddRecord.Enabled = False
Me![Insitu]!DeleteRecord.Enabled = False
Me![Insitu]!AddRecord.Enabled = False
End Sub

'Moves to the previous record
Private Sub GoBack_Click()
On Error Resume Next
     DoCmd.GoToRecord , , acPrevious
End Sub

'Moves to the first record
Private Sub GoFirst_Click()
On Error Resume Next
     DoCmd.GoToRecord , , acFirst
End Sub

'Moves to the next record
Private Sub GoForward_Click()
On Error Resume Next
     DoCmd.GoToRecord , , acNext
End Sub

'Moves to the last record
Private Sub GoLast_Click()
On Error Resume Next
     DoCmd.GoToRecord , , acLast
End Sub

'Closes the form
Private Sub BackButton_Click()
On Error Resume Next
     DoCmd.Close acForm, "View_General"
End Sub
```

```
'Prints the general project information
Private Sub PrintButton_Click()
   Print_Units = 1
   Print_Excel = 1
   Current_Project = Me![Project ID]
   Report_Name = "General"
   Print_Filter = "[Project ID] =" & (Current_Project)
   DoCmd.OpenForm "frmPrint", acNormal
   'DoCmd.OpenReport "General", acViewPreview, , "[Project ID] =" & (Current_Project)
End Sub

Private Sub Edit_Click()
   'Opens frmGeneral
   DoCmd.OpenForm "frmGeneral", , , "[Project ID]=" & Me![Project ID]
   'Hides all the controls in the navigation box, except the delete record button
   Forms![frmGeneral]![GoFirst].Enabled = False
   Forms![frmGeneral]![GoBack].Enabled = False
   Forms![frmGeneral]![GoForward].Enabled = False
   Forms![frmGeneral]![GoLast].Enabled = False
   Forms![frmGeneral]![AddRecord].Enabled = False
   Forms![frmGeneral]![Search].Visible = False
End Sub
```

View_Piles

```
Option Compare Database
Option Explicit
Dim CurrentPile As String
Dim Current_Pile As Integer


Private Sub Form_Current()
'Disables the Driving, GRL and Load Test Tabs until a pile record is started
    If IsNull(Me![Textbox_Units]) Then
      Me![Driving].Enabled = False
      Me![GRL Data].Enabled = False
      Me![Load Tests].Enabled = False
    Else
      Me![Driving].Enabled = True
      Me![GRL Data].Enabled = True
      Me![Load Tests].Enabled = True
    End If
  'Calls the ViewP subprocedure
  ViewP
  'Updates the value of the Load test current record in the navigation box
  Me!CurrentRecordNumber.Value = Me.CurrentRecord
  'Checks if a set of units is selected and updates the Units control
  Select Case Me![Textbox_Units]
  Case "SI"
  Me![Units].Value = 1
  Case "English"
  Me![Units].Value = 2
  Case Else
  Me![Units].Value = 0
  End Select
  'Updates the PileUnits variable
  PileUnits = Me![Units]
  'Calls the UnitsPile subprocecure
  UnitsPile
  Me![Latitude].SetFocus
End Sub

Private Sub ViewP()
  'Resets the visible properties of the tab control
  Me![Units].SetFocus
  Me![H Steel].Visible = False
  Me![Pipe].Visible = False
  Me![Concrete].Visible = False
 'Displays only the tab for the selected pile type and Updates the CurrentPile variable
  Select Case Me![Pile Type]
       Case "Concrete Pile"
        Me![Concrete].Visible = True
        CurrentPile = "Concrete Pile"
       Case "H Steel Pile"
        Me![H Steel].Visible = True
        CurrentPile = "H Steel Pile"
       Case "Pipe Steel Pile"
        Me![Pipe].Visible = True
        CurrentPile = "Pipe Steel Pile"
       Case Else
        CurrentPile = "No Pile"
 End Select
End Sub
```

```
Private Sub UnitsPile()
  'Changes all the labels to the specified set of units
  Select Case PileUnits
  Case 1
    Me![Pile Elastic Modulus Label].Caption = "Pile Elastic Modulus (KN/m²)"
    Me![Capwap Elastic Modulus Label].Caption = "Capwap Elastic Modulus (KN/m²)"
    Me![Pile Wave Speed Label].Caption = "Pile Wave Speed c (m/s)"
    Me![Pile Impedance Label].Caption = "Pile Impedance E*A/c (KN-s/m)"
    Me![Water Table Elev Label].Caption = "Water Table Elev (m)"
    Me![Ground Surface Elev Label].Caption = "Ground Surface Elev (m)"
    Me![Concrete]![Void Diameter Label].Caption = "Void Diameter (mm)"
    Me![Concrete]![Width Label].Caption = "Width or Diameter (mm)"
    Me![Concrete]![Cross Sectional Area Label].Caption = "Cross Sectional Area (mm²)"
    Me![Concrete]![Total Length Label].Caption = "Total Length (m)"
    Me![Concrete]![Embedded Length Label].Caption = "Embedded Length (m)"
    Me![Concrete]![Total Weight Label].Caption = "Total Weight (KN)"
    Me![Concrete]![Unit Weight of Concrete Label].Caption = "Unit Weight of Concrete (KN/m^3)"
    Me![Concrete]![Concrete strength Label].Caption = "Concrete Strength (KN/m²)"
    Me![H Steel]![Steel Cross Sectional Area Label].Caption = "Steel Cross Sectional Area (mm²)"
    Me![H Steel]![Pile Cross Sectional Area Label].Caption = "Pile Cross Sectional Area (mm²)"
    Me![H Steel]![Steel Section Perimeter Label].Caption = "Steel Section Perimeter (mm)"
    Me![H Steel]![Pile  Perimeter Label].Caption = "Pile  Perimeter (mm)"
    Me![H Steel]![Weight of Steel  Label].Caption = "Weight of Steel (KN/m)"
    Me![H Steel]![Weight of pile Label].Caption = "Weight of pile (KN)"
    Me![H Steel]![Total Length Label].Caption = "Total Length (m)"
    Me![H Steel]![Embedded Length Label].Caption = "Embedded Length (m)"
    Me![H Steel]![Unit Weight of Steel Label].Caption = "Unit Weight of Steel (KN/m^3)"
    Me![H Steel]![Concrete strength Label].Caption = "Concrete Strength (KN/m²)"
    Me![Pipe]![Outside Diameter Label].Caption = "Outside Diameter (mm)"
    Me![Pipe]![Thickness of Steel Label].Caption = "Thickness of Steel (mm)"
    Me![Pipe]![Steel Cross Sectional Area Label].Caption = "Steel Cross Sectional Area (mm²)"
    Me![Pipe]![Pile Cross Sectional Area Label].Caption = "Pile Cross Sectional Area (mm²)"
    Me![Pipe]![Concrete strength Label].Caption = "Concrete Strength (KN/m²)"
    Me![Pipe]![Weight of Steel Label].Caption = "Weight of Steel (KN/m)"
    Me![Pipe]![Weight of pile Label].Caption = "Weight of  Pile (KN)"
    Me![Pipe]![Total Length Label].Caption = "Total Length (m)"
    Me![Pipe]![Embedded Length Label].Caption = "Embedded Length (m)"
    Me![Pipe]![Unit Weight of Steel Label].Caption = "Unit Weight of Steel (KN/m^3)"
  Case 2
    Me![Pile Elastic Modulus Label].Caption = "Pile Elastic Modulus (psi)"
    Me![Capwap Elastic Modulus Label].Caption = "Capwap Elastic Modulus (psi)"
    Me![Pile Wave Speed Label].Caption = "Pile Wave Speed c (ft/s)"
    Me![Pile Impedance Label].Caption = "Pile Impedance E*A/c (lb-s/ft)"
    Me![Water Table Elev Label].Caption = "Water Table Elev (ft)"
    Me![Ground Surface Elev Label].Caption = "Ground Surface Elev (ft)"
    Me![Concrete]![Void Diameter Label].Caption = "Void Diameter (in)"
    Me![Concrete]![Width Label].Caption = "Width or Diameter (in)"
    Me![Concrete]![Cross Sectional Area Label].Caption = "Cross Sectional Area (in²)"
    Me![Concrete]![Total Length Label].Caption = "Total Length (ft)"
    Me![Concrete]![Embedded Length Label].Caption = "Embedded Length (ft)"
    Me![Concrete]![Total Weight Label].Caption = "Total Weight (tons)"
    Me![Concrete]![Unit Weight of Concrete Label].Caption = "Unit Weight of Concrete (pcf)"
    Me![Concrete]![Concrete strength Label].Caption = "Concrete Strength (psi)"
    Me![H Steel]![Steel Cross Sectional Area Label].Caption = "Steel Cross Sectional Area (in²)"
    Me![H Steel]![Pile Cross Sectional Area Label].Caption = "Pile Cross Sectional Area (in²)"
    Me![H Steel]![Steel Section Perimeter Label].Caption = "Steel Section Perimeter (in)"
    Me![H Steel]![Pile  Perimeter Label].Caption = "Pile  Perimeter (in)"
    Me![H Steel]![Weight of Steel  Label].Caption = "Weight of Steel (tons/ft)"
    Me![H Steel]![Weight of pile Label].Caption = "Weight of pile (tons)"
    Me![H Steel]![Total Length Label].Caption = "Total Length (ft)"
    Me![H Steel]![Embedded Length Label].Caption = "Embedded Length (ft)"
```

```
        Me![H Steel]![Unit Weight of Steel Label].Caption = "Unit Weight of Steel (pcf)"
        Me![H Steel]![Concrete strength Label].Caption = "Concrete Strength (psi)"
        Me![Pipe]![Outside Diameter Label].Caption = "Outside Diameter (in)"
        Me![Pipe]![Thickness of Steel Label].Caption = "Thickness of Steel (in)"
        Me![Pipe]![Steel Cross Sectional Area Label].Caption = "Steel Cross Sectional Area (in²)"
        Me![Pipe]![Pile Cross Sectional Area Label].Caption = "Pile Cross Sectional Area (in²)"
        Me![Pipe]![Concrete strength Label].Caption = "Concrete Strength (psi)"
        Me![Pipe]![Weight of Steel Label].Caption = "Weight of Steel (tons/ft)"
        Me![Pipe]![Weight of pile Label].Caption = "Weight of  Pile (tons)"
        Me![Pipe]![Total Length Label].Caption = "Total Length (ft)"
        Me![Pipe]![Embedded Length Label].Caption = "Embedded Length (ft)"
        Me![Pipe]![Unit Weight of Steel Label].Caption = "Unit Weight of Steel (pcf)"
    Case Else
        Me![Pile Elastic Modulus Label].Caption = "Pile Elastic Modulus"
        Me![Capwap Elastic Modulus Label].Caption = "Capwap Elastic Modulus"
        Me![Pile Wave Speed Label].Caption = "Pile Wave Speed ( c )"
        Me![Pile Impedance Label].Caption = "Pile Impedance ( E*A/c )"
        Me![Water Table Elev Label].Caption = "Water Table Elev"
        Me![Ground Surface Elev Label].Caption = "Ground Surface Elev"
        Me![Concrete]![Void Diameter Label].Caption = "Void Diameter"
        Me![Concrete]![Width Label].Caption = "Width or Diameter"
        Me![Concrete]![Cross Sectional Area Label].Caption = "Cross Sectional Area"
        Me![Concrete]![Total Length Label].Caption = "Total Length"
        Me![Concrete]![Embedded Length Label].Caption = "Embedded Length"
        Me![Concrete]![Total Weight Label].Caption = "Total Weight"
        Me![Concrete]![Unit Weight of Concrete Label].Caption = "Unit Weight of Concrete"
        Me![Concrete]![Concrete strength Label].Caption = "Concrete Strength"
        Me![H Steel]![Steel Cross Sectional Area Label].Caption = "Steel Cross Sectional Area"
        Me![H Steel]![Pile Cross Sectional Area Label].Caption = "Pile Cross Sectional Area"
        Me![H Steel]![Steel Section Perimeter Label].Caption = "Steel Section Perimeter"
        Me![H Steel]![Pile  Perimeter Label].Caption = "Pile  Perimeter"
        Me![H Steel]![Weight of Steel  Label].Caption = "Weight of Steel"
        Me![H Steel]![Weight of pile Label].Caption = "Weight of pile"
        Me![H Steel]![Total Length Label].Caption = "Total Length"
        Me![H Steel]![Embedded Length Label].Caption = "Embedded Length"
        Me![H Steel]![Unit Weight of Steel Label].Caption = "Unit Weight of Steel"
        Me![H Steel]![Concrete strength Label].Caption = "Concrete Strength"
        Me![Pipe]![Outside Diameter Label].Caption = "Outside Diameter"
        Me![Pipe]![Thickness of Steel Label].Caption = "Thickness of Steel"
        Me![Pipe]![Steel Cross Sectional Area Label].Caption = "Steel Cross Sectional Area"
        Me![Pipe]![Pile Cross Sectional Area Label].Caption = "Pile Cross Sectional Area"
        Me![Pipe]![Concrete strength Label].Caption = "Concrete Strength"
        Me![Pipe]![Weight of Steel Label].Caption = "Weight of Steel"
        Me![Pipe]![Weight of pile Label].Caption = "Weight of  Pile"
        Me![Pipe]![Total Length Label].Caption = "Total Length"
        Me![Pipe]![Embedded Length Label].Caption = "Embedded Length"
        Me![Pipe]![Unit Weight of Steel Label].Caption = "Unit Weight of Steel"
    End Select
End Sub

'Moves to the previous record
Private Sub GoBack_Click()
On Error Resume Next
    DoCmd.GoToRecord , , acPrevious
End Sub

'Moves to the next record
Private Sub GoForward_Click()
On Error Resume Next
    DoCmd.GoToRecord , , acNext
End Sub
```

```
'Prints the pile information
Private Sub PrintButton_Click()
   Print_Units = 0
   Print_Excel = 0
   Current_Pile = Me![Pile ID]
   Report_Name = "Pile"
   Print_Filter = "[Pile ID] =" & (Current_Pile)
   DoCmd.OpenForm "frmPrint", acNormal
End Sub
```

View_Shafts

```
Option Compare Database
Option Explicit
Dim Current_Shaft As Integer

Private Sub Form_Current()
  'Disables IGM and Load Tets Tabs until a shaft record is started
    If IsNull(Me![Textbox_Units]) Then
      Me![IGM Tab].Enabled = False
      Me![Load Tests].Enabled = False
    Else
      Me![IGM Tab].Enabled = True
      Me![Load Tests].Enabled = True
    End If
  'Shows the cursor in Latitude with the General tab selected
  Me!Units.SetFocus
  Me!Latitude.SetFocus
  Me!CurrentRecordNumber.Value = Me.CurrentRecord
  'Checks if a set of units is selected and updates the Units control
  Select Case Me![Textbox_Units]
  Case "SI"
  Me![Units].Value = 1
  Case "English"
  Me![Units].Value = 2
  Case Else
  Me![Units].Value = 0
  End Select
  'Updates the ShaftUnits variable
  ShaftUnits = Me![Units]
  'Calls the UnitsShaft subprocedure
  UnitsShaft
End Sub

Private Sub UnitsShaft()
'Changes all the labels to the specified set of units
Select Case ShaftUnits
  Case 1
    Me![Diameter Label].Caption = "Diameter (mm)"
    Me![Total Length Label].Caption = "Total Length (m)"
    Me![Embeded Length Label].Caption = "Embeded Length (m)"
    Me![Casing Length Label].Caption = "Casing Length (m)"
    Me![Bell Diameter Label].Caption = "Bell Diameter (mm)"
    Me![Bell Length Label].Caption = "Bell Length (m)"
    Me![Slump of Concrete Label].Caption = "Slump of Concrete (mm)"
    Me![Concrete strength Label].Caption = "Concrete strength f'c (KN/m²)"
    Me![Core Modulus Label].Caption = "Core Modulus Ec (KN/m²)"
    Me![Elev Label].Caption = "(m)"
    Me![Depth Label].Caption = "(m)"
    Me![qu Label].Caption = "(KN/m²)"
    Me![qt Label].Caption = "(KN/m²)"
    Me![Em Label].Caption = "(KN/m²)"
    Me![Ei Label].Caption = "(KN/m²)"
    Me![Water Table Elev Label].Caption = "Water Table Elev (m)"
    Me![Ground Surface Elev Label].Caption = "Ground Surface Elev (m)"
    Me![As built Design Load  Label].Caption = "Design Load (kN)"
    Me![As Built Diameter Label].Caption = "Diameter (mm)"
    Me![As Built Tip Elev Label].Caption = "Tip Elev (m)"
    Me![As Built Total Length Label].Caption = "Total Length (m)"
    Me![As Built Rock Socket Length Label].Caption = "Rock Socket Length (m)"
```

```
    Me![Top Elev Label].Caption = "Top Elev (m)"
    Me![Tip Elev Label].Caption = "Tip Elev (m)"
    Me![Scour Elev Label].Caption = "Scour Elev (m)"
    Me![Rock Socket Length Label].Caption = "Rock Socket Length (m)"
    Me![Design Load Label].Caption = "Design Load (kN)"
  Case 2
    Me![Diameter Label].Caption = "Diameter (in)"
    Me![Total Length Label].Caption = "Total Length (ft)"
    Me![Embeded Length Label].Caption = "Embeded Length (ft)"
    Me![Casing Length Label].Caption = "Casing Length (ft)"
    Me![Bell Diameter Label].Caption = "Bell Diameter (in)"
    Me![Bell Length Label].Caption = "Bell Length (ft)"
    Me![Slump of Concrete Label].Caption = "Slump of Concrete (in)"
    Me![Concrete strength Label].Caption = "Concrete strength f'c (Ton/ft²)"
    Me![Core Modulus Label].Caption = "Core Modulus Ec (Ton/ft²)"
    Me![Elev Label].Caption = "(ft)"
    Me![Depth Label].Caption = "(ft)"
    Me![qu Label].Caption = "(Ton/ft²)"
    Me![qt Label].Caption = "(Ton/ft²)"
    Me![Em Label].Caption = "(Ton/ft²)"
    Me![Ei Label].Caption = "(Ton/ft²)"
    Me![Water Table Elev Label].Caption = "Water Table Elev (ft)"
    Me![Ground Surface Elev Label].Caption = "Ground Surface Elev (ft)"
    Me![As built Design Load  Label].Caption = "Design Load (tons)"
    Me![As Built Diameter Label].Caption = "Diameter (in)"
    Me![As Built Tip Elev Label].Caption = "Tip Elev (ft)"
    Me![As Built Total Length Label].Caption = "Total Length (ft)"
    Me![As Built Rock Socket Length Label].Caption = "Rock Socket Length (ft)"
    Me![Top Elev Label].Caption = "Top Elev (ft)"
    Me![Tip Elev Label].Caption = "Tip Elev (ft)"
    Me![Scour Elev Label].Caption = "Scour Elev (ft)"
    Me![Rock Socket Length Label].Caption = "Rock Socket Length (ft)"
    Me![Design Load Label].Caption = "Design load (tons)"
  Case Else
    Me![Diameter Label].Caption = "Diameter"
    Me![Total Length Label].Caption = "Total Length"
    Me![Embeded Length Label].Caption = "Embeded Length"
    Me![Casing Length Label].Caption = "Casing Length"
    Me![Bell Diameter Label].Caption = "Bell Diameter"
    Me![Bell Length Label].Caption = "Bell Length"
    Me![Slump of Concrete Label].Caption = "Slump of Concrete"
    Me![Concrete strength Label].Caption = "Concrete strength f'c"
    Me![Core Modulus Label].Caption = "Core Modulus Ec"
    Me![Elev Label].Caption = ""
    Me![Depth Label].Caption = ""
    Me![qu Label].Caption = ""
    Me![qt Label].Caption = ""
    Me![Em Label].Caption = ""
    Me![Ei Label].Caption = ""
    Me![Water Table Elev Label].Caption = "Water Table Elev"
    Me![Ground Surface Elev Label].Caption = "Ground Surface Elev"
    Me![As built Design Load  Label].Caption = "Design Load"
    Me![As Built Diameter Label].Caption = "Diameter"
    Me![As Built Tip Elev Label].Caption = "Tip Elev"
    Me![As Built Total Length Label].Caption = "Total Length"
    Me![As Built Rock Socket Length Label].Caption = "Rock Socket Length"
    Me![Top Elev Label].Caption = "Top Elev"
    Me![Tip Elev Label].Caption = "Tip Elev"
    Me![Scour Elev Label].Caption = "Scour Elev"
    Me![Rock Socket Length Label].Caption = "Rock Socket Length"
    Me![Design Load Label].Caption = "Design Load"
  End Select
```

```
End Sub

'Moves to the previous record
Private Sub GoBack_Click()
On Error Resume Next
   DoCmd.GoToRecord , , acPrevious
End Sub

'Moves to the next record
Private Sub GoForward_Click()
On Error Resume Next
   DoCmd.GoToRecord , , acNext
End Sub

'Prints the shaft information
Private Sub PrintButton_Click()
   Print_Units = 0
   Print_Excel = 0
   Current_Shaft = Me![Shaft ID]
   Report_Name = "Shaft"
   Print_Filter = "[Shaft ID] =" & (Current_Shaft)
   DoCmd.OpenForm "frmPrint", acNormal
End Sub
```

View_Insitu Tests

```
Option Compare Database
Option Explicit
Dim CurrentTest As String
Dim Current_Insitu As Integer

Private Sub Form_Current()
   'Calls the ViewInsitu subprocedure
   ViewInsitu
   'Updates the value of the insitu test current record in the navigation box
   Me!CurrentRecordNumber.Value = Me.CurrentRecord
   'Updates the Units control
   Select Case Me![Textbox_Units]
   Case "SI"
   Me![Units].Value = 1
   Case "English"
   Me![Units].Value = 2
   Case Else
   Me![Units].Value = 0
   End Select
   'Updates the InsituUnits variable
   InsituUnits = Me![Units]
   'Calls the UnitsInsitu subprocedure
   UnitsInsitu
End Sub

Private Sub ViewInsitu()
   'Resets the visible properties of the tab control
   Me![Box].Visible = True
   Me![CPT].Visible = False
   Me![DMT].Visible = False
   Me![PRESSUREMETER].Visible = False
   Me![SPT].Visible = False
  'Displays only the selected Insitu test tab and Updates the CurrentTest variable
  Select Case Me![Type of Test]
        Case "SPT"
        Me![SPT].Visible = True
        CurrentTest = "SPT"
        Case "CPT"
        Me![CPT].Visible = True
        CurrentTest = "CPT"
        Case "DMT"
        Me![DMT].Visible = True
        CurrentTest = "DMT"
        Case "PMT"
        Me![PRESSUREMETER].Visible = True
        CurrentTest = "DMT"
        Case Else
        Me![Box].Visible = False
        CurrentTest = "No Test"
  End Select
End Sub

Private Sub UnitsInsitu()
   'Changes all the labels to the specified set of units
   Select Case Me![Units]
   Case 1
      Me![Water Table Elev Label].Caption = "Water Table Elev (m)"
      Me![Ground Surface Elev Label].Caption = "Ground Surface Elev (m)"
```

```
        Me![Test Depth Label].Caption = "Test Depth (m)"
        Me![SPT Elev Label].Caption = "(m)"
        Me![SPT Depth Label].Caption = "(m)"
        Me![CPT Elev Label].Caption = "(m)"
        Me![CPT Depth Label].Caption = "(m)"
        Me![Cone Resistance Label].Caption = "(KN/m²)"
        Me![Friction Resistance Label].Caption = "(KN/m²)"
        Me![DMT Elev Label].Caption = "(m)"
        Me![DMT Depth Label].Caption = "(m)"
        Me![PMT Pressure Label].Caption = "(KN/m²)"
        Me![PMT Volume Label].Caption = "(mm^3)"
    Case 2
        Me![Water Table Elev Label].Caption = "Water Table Elev (ft)"
        Me![Ground Surface Elev Label].Caption = "Ground Surface Elev (ft)"
        Me![Test Depth Label].Caption = "Test Depth (ft)"
        Me![SPT Elev Label].Caption = "(ft)"
        Me![SPT Depth Label].Caption = "(ft)"
        Me![CPT Elev Label].Caption = "(ft)"
        Me![CPT Depth Label].Caption = "(ft)"
        Me![Cone Resistance Label].Caption = "(Ton/ft²)"
        Me![Friction Resistance Label].Caption = "(Ton/ft²)"
        Me![DMT Elev Label].Caption = "(ft)"
        Me![DMT Depth Label].Caption = "(ft)"
        Me![PMT Pressure Label].Caption = "(Ton/ft²)"
        Me![PMT Volume Label].Caption = "(in^3)"
    Case Else
        Me![Water Table Elev Label].Caption = "Water Table Elev"
        Me![Ground Surface Elev Label].Caption = "Ground Surface Elev"
        Me![Test Depth Label].Caption = "Test Depth"
        Me![SPT Elev Label].Caption = ""
        Me![SPT Depth Label].Caption = ""
        Me![CPT Elev Label].Caption = ""
        Me![CPT Depth Label].Caption = ""
        Me![Cone Resistance Label].Caption = ""
        Me![Friction Resistance Label].Caption = ""
        Me![DMT Elev Label].Caption = ""
        Me![DMT Depth Label].Caption = ""
        Me![PMT Pressure Label].Caption = ""
        Me![PMT Volume Label].Caption = ""
    End Select
End Sub

'Moves to the previous record
Private Sub GoBack_Click()
On Error Resume Next
    DoCmd.GoToRecord , , acPrevious
End Sub

'Moves to the next record
Private Sub GoForward_Click()
On Error Resume Next
    DoCmd.GoToRecord , , acNext
End Sub

'Prints the Insitu Test information
Private Sub PrintButton_Click()
    Print_Units = 0
    Print_Excel = 0
    Current_Insitu = Me![Insitu test ID]
      Select Case Me![Type of Test]
        Case "SPT"
        Report_Name = "Insitu SPT"
```

```
        Case "CPT"
        Report_Name = "Insitu CPT"
        Case "DMT"
        Report_Name = "Insitu DMT"
        Case "PMT"
        Report_Name = "Insitu PMT"
  End Select
    Print_Filter = "[Insitu test ID] =" & (Current_Insitu)
    DoCmd.OpenForm "frmPrint", acNormal
End Sub
```

View_Load Tests

```
Option Compare Database
Option Explicit
Dim CurrentTest As String
Dim Current_Load As Integer

Private Sub Form_Current()
   'Calls the ViewLT subprocedure
   ViewLT
   'Updates the value of the Load test current record in the navigation box
   Me!CurrentRecordNumber.Value = Me.CurrentRecord
   'Updates the Units control
   Select Case Me![Textbox_Units]
   Case "SI"
   Me![Units].Value = 1
   Case "English"
   Me![Units].Value = 2
   Case Else
   Me![Units].Value = 0
   End Select
   'Updates the LoadTestUnits variable
   LoadTestUnits = Me![Units]
   'Calls the UnitsLT subprocedure
   UnitsLT
 Me!Units.SetFocus
End Sub

Private Sub ViewLT()
  'Resets the visible properties of the tab control
   Me![Box].Visible = True
   Me![Statnamic test].Visible = False
   Me![Static test].Visible = False
   Me![Osterberg test].Visible = False
  'Displays only the selected Load test tab and Updates the CurrentTest variable
   Select Case Me![Type of Test]
         Case "Static Test"
         Me![Static test].Visible = True
         CurrentTest = "Static Test"
         Case "Statnamic Test"
         Me![Statnamic test].Visible = True
         CurrentTest = "Statnamic Test"
         Case "Osterberg Test"
         Me![Osterberg test].Visible = True
         CurrentTest = "Osterberg Test"
         Case Else
         Me![Box].Visible = False
         CurrentTest = "No Test"
   End Select
End Sub

Private Sub UnitsLT()
    'Changes all the labels to the specified set of units
   Select Case LoadTestUnits
   Case 1
      Me![Static Time Label].Caption = "(min)"
      Me![Static Force Label].Caption = "(KN)"
      Me![Static Displacement Label].Caption = "(mm)"
      Me![Osterberg]![Load Label].Caption = "Load (KN)"
      Me![Osterberg]![Displacement Label].Caption = "Displacement of Cell (mm)"
```

```
        Me![Osterberg]![Skin Friction Label].Caption = "Unit Skin Friction (KN/m²)"
        Me![Statnamic]![Load Label].Caption = "Load (KN)"
        Me![Statnamic]![Equiv Load Label].Caption = "Load (KN)"
        Me![Statnamic]![Displacement Label].Caption = "Top of Element (mm)"
        Me![Statnamic]![Skin Friction Label].Caption = "Unit Skin Friction (KN/m²)"
    Case 2
        Me![Static Time Label].Caption = "(min)"
        Me![Static Force Label].Caption = "(Ton)"
        Me![Static Displacement Label].Caption = "(in)"
        Me![Osterberg]![Load Label].Caption = "Load (tons)"
        Me![Osterberg]![Displacement Label].Caption = "Displacement of Cell (in)"
        Me![Osterberg]![Skin Friction Label].Caption = "Unit Skin Friction (tsf)"
        Me![Statnamic]![Load Label].Caption = "Load (tons)"
        Me![Statnamic]![Equiv Load Label].Caption = "Load (tons)"
        Me![Statnamic]![Displacement Label].Caption = "Top of Element (in)"
        Me![Statnamic]![Skin Friction Label].Caption = "Unit Skin Friction (tsf)"
    Case Else
        Me![Static Time Label].Caption = ""
        Me![Static Force Label].Caption = ""
        Me![Static Displacement Label].Caption = ""
        Me![Osterberg]![Load Label].Caption = "Load"
        Me![Osterberg]![Displacement Label].Caption = "Displacement of Cell"
        Me![Osterberg]![Skin Friction Label].Caption = "Unit Skin Friction"
        Me![Statnamic]![Load Label].Caption = "Load"
        Me![Statnamic]![Equiv Load Label].Caption = "Load"
        Me![Statnamic]![Displacement Label].Caption = "Top of Element"
        Me![Statnamic]![Skin Friction Label].Caption = "Unit Skin Friction"
    End Select
End Sub

'Moves to the previous record
Private Sub GoBack_Click()
On Error Resume Next
    DoCmd.GoToRecord , , acPrevious
End Sub

'Moves to the next record
Private Sub GoForward_Click()
On Error Resume Next
    DoCmd.GoToRecord , , acNext
End Sub

'Prints the load test information
Private Sub PrintButton_Click()
    Print_Units = 0
    Print_Excel = 0
    Current_Load = Me![Load test ID]
        Select Case Me![Type of Test]
            Case "Static Test"
            Report_Name = "Load Test Static"
            Case "Statnamic Test"
            Report_Name = "Load Test Statnamic"
            Case "Osterberg Test"
            Report_Name = "Load Test Osterberg"
        End Select
    Print_Filter = "[Load test ID] =" & (Current_Load)
    DoCmd.OpenForm "frmPrint", acNormal
End Sub
```

frmMenu

```
Option Compare Database
Option Explicit

Private Sub Form_Current()
DoCmd.Maximize
End Sub

'Opens frmGeneral and starts a new record
Private Sub InputButton_Click()
On Error Resume Next
   DoCmd.OpenForm "frmGeneral", acNormal, , , , acWindowNormal
   DoCmd.GoToRecord acActiveDataObject, , acNewRec
End Sub

'Opens frmSearch
Private Sub SearchButton_Click()
On Error Resume Next
   DoCmd.OpenForm "frmSearch", acNormal, , , , acDialog
End Sub

'Quits Access
Private Sub QuitButton_Click()
On Error Resume Next
   DoCmd.quit
End Sub
```

frmSearch

```
Option Compare Database
Option Explicit

Dim db As Database
Dim qdfGeneral As QueryDef
Dim qdfPiles As QueryDef
Dim qdfShafts As QueryDef
Dim qdfInsitu As QueryDef
Dim qdfLoad As QueryDef
Dim General_SELECT As String
Dim Piles_SELECT As String
Dim Shafts_SELECT As String
Dim Insitu_SELECT As String
Dim Load_SELECT As String
Dim General_WHERE As String
Dim Piles_WHERE As String
Dim Shafts_WHERE As String
Dim Insitu_WHERE As String
Dim Load_WHERE As String
Dim General_SQL As String
Dim Piles_SQL As String
Dim Shafts_SQL As String
Dim Insitu_SQL As String
Dim Load_SQL As String
Dim ProjectName_WHERE As String
Dim ProjectNumber_WHERE As String
Dim State_WHERE As String
Dim County_WHERE As String
Dim GeotechnicalCompany_WHERE As String
Dim MaxLatitude_WHERE As String
Dim MinLatitude_WHERE As String
Dim MaxLongitude_WHERE As String
Dim MinLongitude_WHERE As String
Dim ElemWHERE_Piles As String
Dim General_Element_WHERE As String
Dim ElemWHERE_Shafts As String
Dim PileType_WHERE As String
Dim InsituTestType_WHERE As String
Dim General_Insitu_WHERE As String
Dim LoadTestType_WHERE As String
Dim Piles_LoadTest_WHERE As String
Dim Shafts_LoadTest_WHERE As String
Dim General_LoadTest_WHERE As String
Dim MaxTotalLength_WHERE_Piles As String
Dim MaxTotalLength_WHERE_Shafts As String
Dim MinTotalLength_WHERE_Piles As String
Dim MinTotalLength_WHERE_Shafts As String
Dim MaxEmbLength_WHERE_Piles As String
Dim MaxEmbLength_WHERE_Shafts As String
Dim MinEmbLength_WHERE_Piles As String
Dim MinEmbLength_WHERE_Shafts As String
Dim MaxDiameter_WHERE_Piles As String
Dim MaxDiameter_WHERE_Shafts As String
Dim MinDiameter_WHERE_Piles As String
Dim MinDiameter_WHERE_Shafts As String
Dim Driving_WHERE As String
Dim GRL_WHERE As String
Dim QueryCount As Long
```

```
Private Sub Clear()
 'Disables the Pile control untill "Driven Pile" is selected from the Element control
  Me!Pile.Enabled = False
  Me!Driving.Enabled = False
  Me!GRL.Enabled = False
  'Resets the Units control
  Me!Units.Value = 0
  'Resets all labels
  Me![Max Total Length Label].Caption = "Max Total Length"
  Me![Min Total Length Label].Caption = "Min Total Length"
  Me![Max Embedded Length Label].Caption = "Max Embedded Length"
  Me![Min Embedded Length Label].Caption = "Min Embedded Length"
  Me![Max Diameter Label].Caption = "Max Diameter"
  Me![Min Diameter label].Caption = "Min Diameter"
  'Resets all criteria
  ProjectName_WHERE = ""
  ProjectNumber_WHERE = ""
  State_WHERE = ""
  County_WHERE = ""
  GeotechnicalCompany_WHERE = ""
  MaxLatitude_WHERE = ""
  MinLatitude_WHERE = ""
  MaxLongitude_WHERE = ""
  MinLongitude_WHERE = ""
  ElemWHERE_Piles = ""
  ElemWHERE_Shafts = ""
  General_Element_WHERE = ""
  PileType_WHERE = ""
  InsituTestType_WHERE = ""
  General_Insitu_WHERE = ""
  LoadTestType_WHERE = ""
  Piles_LoadTest_WHERE = ""
  Shafts_LoadTest_WHERE = ""
  General_LoadTest_WHERE = ""
  MaxTotalLength_WHERE_Piles = ""
  MaxTotalLength_WHERE_Shafts = ""
  MinTotalLength_WHERE_Piles = ""
  MinTotalLength_WHERE_Shafts = ""
  MaxEmbLength_WHERE_Piles = ""
  MaxEmbLength_WHERE_Shafts = ""
  MinEmbLength_WHERE_Piles = ""
  MinEmbLength_WHERE_Shafts = ""
  MaxDiameter_WHERE_Piles = ""
  MaxDiameter_WHERE_Shafts = ""
  MinDiameter_WHERE_Piles = ""
  MinDiameter_WHERE_Piles = ""
  Driving_WHERE = ""
  GRL_WHERE = ""
  'Resets all controls
  Me!Element.Value = 1
  Me!Pile.Value = 1
  Me!Insitu.Value = 1
  Me!Load.Value = 1
  Me![Max Total Length].Value = Null
  Me![Min Total Length].Value = Null
  Me![Max Embedded Length].Value = Null
  Me![Min Embedded Length].Value = Null
  Me![Max Diameter].Value = Null
  Me![Min Diameter].Value = Null
  Me![Project Name] = ""
```

```
    Me![Project Number] = ""
    Me![State] = ""
    Me![County] = ""
    Me![Geotechnical Company] = ""
    Me![Max Latitude] = ""
    Me![Min Latitude] = ""
    Me![Max Longitude] = ""
    Me![Min Longitude] = ""
    Me!Driving = False
    Me!GRL = False
    'Disables the controls involving units to force user to select units
    Me![Min Total Length].Enabled = False
    Me![Max Total Length].Enabled = False
    Me![Min Embedded Length].Enabled = False
    Me![Max Embedded Length].Enabled = False
    Me![Min Diameter].Enabled = False
    Me![Max Diameter].Enabled = False
End Sub

Private Sub Query()
On Error Resume Next
    Set db = CurrentDb
    'Deletes the existing Search Queries
    DoCmd.DeleteObject acQuery, "Query_General"
    DoCmd.DeleteObject acQuery, "Query_Piles"
    DoCmd.DeleteObject acQuery, "Query_Shafts"
    DoCmd.DeleteObject acQuery, "Query_Insitu"
    DoCmd.DeleteObject acQuery, "Query_Load"
    On Error GoTo 0
    'Creates the new Search Queries
    Set qdfGeneral = db.CreateQueryDef("Query_General", General_SQL)
    Set qdfPiles = db.CreateQueryDef("Query_Piles", Piles_SQL)
    Set qdfShafts = db.CreateQueryDef("Query_Shafts", Shafts_SQL)
    Set qdfInsitu = db.CreateQueryDef("Query_Insitu", Insitu_SQL)
    Set qdfLoad = db.CreateQueryDef("Query_Load", Load_SQL)
End Sub

Private Sub Form_Current()
  DoCmd.Maximize
    'Creates the basic Criteria for Query_General (All projects are selected)
    General_SELECT = "SELECT DISTINCT General.[Project ID], General.[Project Name]," & _
      "General.[Project Number], General.Latitude, General.Longitude, General.State," & _
      "General.County, General.[Geotechnical Company], General.Comments " & _
      "FROM (Query_Piles RIGHT JOIN (Query_Shafts RIGHT JOIN (Query_Insitu RIGHT JOIN (Query_Load
RIGHT JOIN [General] ON Query_Load.[Project ID] = General.[Project ID]) ON Query_Insitu.[Project ID] =
General.[Project ID]) ON Query_Shafts.[Project ID] = General.[Project ID]) ON Query_Piles.[Project ID] =
General.[Project ID]) "
    General_WHERE = "WHERE ([General].[Project ID])>0 "
    'Creates the basic Criteria for Query_Piles (All piles are selected)
    Piles_SELECT = "SELECT DISTINCT Piles.[Pile ID], Piles.[Project ID], Piles.Latitude, Piles.Pile, " & _
      "Piles.Longitude, Piles.Station, Piles.Offset, Piles.Units, Piles.[Pile Type], Piles.[Ground Surface Elev],
Piles.[Water Table Elev]," & _
      "Piles.[Type of Load test], Piles.[Pile Elastic Modulus], Piles.[Capwap Elastic Modulus]," & _
      "Piles.[Pile Wave Speed ( c )], Piles.[Pile Impedance ( E*A/c )], Piles.Comments," & _
      "Piles.[Total Length (m)], Piles.[Embedded Length (m)], Piles.[Diameter (mm)], Piles.[Driving Data], Piles.[GRL
Data] " & _
      "FROM Piles LEFT JOIN Query_Load ON Piles.[Pile ID] = Query_Load.[Pile ID] "
    Piles_WHERE = "WHERE ([Piles].[Project ID])>0 "
    'Creates the basic Criteria for Query_Shafts (All shafts are selected)
    Shafts_SELECT = "SELECT DISTINCT Shafts.[Shaft ID], Shafts.[Project ID], Shafts.Shaft, Shafts.Latitude," & _
      "Shafts.Longitude, Shafts.Station, Shafts.Offset, Shafts.Units, Shafts.Diameter," & _
      "Shafts.[Total Length], Shafts.[Embedded Length], Shafts.[Construction Method]," & _
```

```
    "Shafts.[Casing Length], Shafts.[Type of Base], Shafts.[Bell Diameter], Shafts.[Bell Length]," & _
    "Shafts.[Top Elev], Shafts.[Tip Elev], Shafts.[Scour Elev], Shafts.[Rock Socket Length], Shafts.[Ground Surface
Elev], Shafts.[Water Table Elev]," & _
    "Shafts.[Design Load], Shafts.[As Built Diameter], Shafts.[As Built Tip Elev]," & _
    "Shafts.[As Built Total Length], Shafts.[As Built Rock Socket Length], Shafts.[As Built Design Load]," & _
    "Shafts.[Slump of Concrete], Shafts.[Concrete strength  f'c], Shafts.[Core Modulus  Ec]," & _
    "Shafts.[Diameter (mm)], Shafts.[Total Length (m)], Shafts.[Embedded Length (m)]," & _
    "Shafts.[Comments] FROM Shafts LEFT JOIN Query_Load ON Shafts.[Shaft ID] = Query_Load.[Shaft ID] "
  Shafts_WHERE = "WHERE ([Shafts].[Project ID])>0 "
  'Creates the basic Criteria for Query_Insitu (All Insitu Tests are selected)
  Insitu_SELECT = "SELECT [InSitu Tests].[Project ID], [InSitu Tests].[Insitu test ID],[InSitu Tests].[Insitu test], " &
_
    "[InSitu Tests].[Date of Test], [InSitu Tests].Latitude, [InSitu Tests].Longitude," & _
    "[InSitu Tests].Station, [InSitu Tests].Offset, [InSitu Tests].Units, [InSitu Tests].[Test Type]," & _
    "[InSitu Tests].[Ground Surface Elev], [InSitu Tests].[Water Table Elev], [InSitu Tests].[Test Depth], [InSitu
Tests].[DMT Delta A (bars)]," & _
    "[InSitu Tests].[DMT Delta B (bars)], [InSitu Tests].[DMT Gauge Zero (bars)], [InSitu Tests].Comments " & _
    "FROM [InSitu Tests] "
  Insitu_WHERE = "WHERE ([InSitu Tests].[Project ID])>0 "
  'Creates the basic Criteria for Query_Load (All Load Tests are selected)
  Load_SELECT = "SELECT [Load Tests].[Project ID], [Load Tests].[Pile ID], [Load Tests].[Shaft ID], [Load
Tests].[Load test ID]," & _
  "[Load Tests].Units, [Load Tests].[Date of Load Test], [Load Tests].[Type of Test]," & _
  "[Load Tests].[Comments] FROM [Load Tests] "
  Load_WHERE = "WHERE ([Load Tests].[Load test ID])>0 "
End Sub


Private Sub Form_Open(Cancel As Integer)
  Call Clear
End Sub


Private Sub Search_Click()
  'Concatenates all criteria to form the SQL statement
  General_SQL = General_SELECT & General_WHERE & General_Element_WHERE & ProjectName_WHERE &
ProjectNumber_WHERE & State_WHERE & County_WHERE & GeotechnicalCompany_WHERE &
MaxLatitude_WHERE & MinLatitude_WHERE & MaxLongitude_WHERE & MinLongitude_WHERE &
General_Insitu_WHERE & General_LoadTest_WHERE
  Piles_SQL = Piles_SELECT & Piles_WHERE & ElemWHERE_Piles & PileType_WHERE &
MaxTotalLength_WHERE_Piles & MinTotalLength_WHERE_Piles & MaxEmbLength_WHERE_Piles &
MinEmbLength_WHERE_Piles & MaxDiameter_WHERE_Piles & MinDiameter_WHERE_Piles & Driving_WHERE
& GRL_WHERE & Piles_LoadTest_WHERE
  Shafts_SQL = Shafts_SELECT & Shafts_WHERE & ElemWHERE_Shafts & MaxTotalLength_WHERE_Shafts &
MinTotalLength_WHERE_Shafts & MaxEmbLength_WHERE_Shafts & MinEmbLength_WHERE_Shafts &
MaxDiameter_WHERE_Shafts & MinDiameter_WHERE_Shafts & Shafts_LoadTest_WHERE
  Insitu_SQL = Insitu_SELECT & Insitu_WHERE & InsituTestType_WHERE
  Load_SQL = Load_SELECT & Load_WHERE & LoadTestType_WHERE
 'Calls the Query subprocedure
 Call Query
  QueryCount = DCount("Query_General.[Project ID]", "Query_General")
  'Displays a message if no records matching the specified criteria are found
  If QueryCount = 0 Then
  MsgBox "No records were found. Please try a different search criteria"
  Exit Sub
  End If
 'Opens the View_General form to display the search results
 DoCmd.OpenForm "View_General", acNormal, , , , acWindowNormal
End Sub

Private Sub Reset_Click()
 Call Clear
```

```
End Sub

Private Sub Project_Name_AfterUpdate()
'Creates the Project Name criteria
   ProjectName_WHERE = " AND [General].[Project Name] Like '*" & Me![Project Name] & "*'"
End Sub

Private Sub Project_Number_AfterUpdate()
'Creates the Project Number criteria
   ProjectNumber_WHERE = " AND [General].[Project Number] Like '*" & Me![Project Number] & "*'"
End Sub

Private Sub State_AfterUpdate()
'Creates the Project State criteria
   State_WHERE = " AND [General].[State] Like '*" & Me![State] & "*'"
End Sub

Private Sub County_AfterUpdate()
'Creates the Project County criteria
   County_WHERE = " AND [General].[County] Like '*" & Me![County] & "*'"
End Sub

Private Sub Geotechnical_Company_AfterUpdate()
'Creates the Project Geotechnical Company criteria
   GeotechnicalCompany_WHERE = " AND [General].[Geotechnical Company] Like '*" & Me![Geotechnical
Company] & "*'"
End Sub

Private Sub Max_Latitude_AfterUpdate()
'Creates the Project Maximum Latitude criteria
   If IsNull(Me![Max Latitude]) Then
   MaxLatitude_WHERE = ""
   Else
   MaxLatitude_WHERE = " AND [General].[Latitude] <= " & Me![Max Latitude]
   End If
End Sub

Private Sub Min_Latitude_AfterUpdate()
'Creates the Project Minimum Latitude criteria
   If IsNull(Me![Max Latitude]) Then
   MaxLatitude_WHERE = ""
   Else
   MinLatitude_WHERE = " AND [General].[Latitude] >= " & Me![Min Latitude]
   End If
End Sub

Private Sub Max_Longitude_AfterUpdate()
'Creates the Project Maximum Longitude criteria
   If IsNull(Me![Max Latitude]) Then
   MaxLatitude_WHERE = ""
   Else
   MaxLongitude_WHERE = " AND [General].[Longitude] <= " & Me![Max Longitude]
   End If
End Sub

Private Sub Min_Longitude_AfterUpdate()
'Creates the Project Minimum Longitude criteria
   If IsNull(Me![Max Latitude]) Then
   MaxLatitude_WHERE = ""
   Else
   MinLongitude_WHERE = " AND [General].[Longitude] >= " & Me![Min Longitude]
   End If
```

```
End Sub

Private Sub Element_AfterUpdate()
    'Creates the Element Type  Criteria
    General_Element_WHERE = ""
    Select Case Element
     Case 1
     Me!Pile.Enabled = False
     Me!Driving.Enabled = False
     Me!GRL.Enabled = False
     ElemWHERE_Piles = ""
     ElemWHERE_Shafts = ""
     Me!Pile.Value = 1
     Me!Driving = False
     Me!GRL = False
     Case 2
     'Enables the Pile control to allow the user to choose a Pile Type
     Me!Pile.Enabled = True
     Me!Driving.Enabled = True
     Me!GRL.Enabled = True
     ElemWHERE_Shafts = " AND ([Shafts].[Shaft ID])= 0 "
     ElemWHERE_Piles = ""
     General_Element_WHERE = " AND ((General.[Project ID])=[Query_Piles].[Project ID])"
     Me!Pile.Value = 1
     Me!Driving = False
     Me!GRL = False
     Case 3
     Me!Pile.Enabled = False
     Me!Driving.Enabled = False
     Me!GRL.Enabled = False
     ElemWHERE_Piles = " AND ([Piles].[Pile ID])= 0 "
     ElemWHERE_Shafts = ""
     General_Element_WHERE = " AND ((General.[Project ID])=[Query_Shafts].[Project ID])"
     Me!Pile.Value = 1
     Me!Driving = False
     Me!GRL = False

    End Select
End Sub

Private Sub Pile_AfterUpdate()
    'Completes pile criteria depending on the selected pile type
    Select Case Pile
     Case 1
     PileType_WHERE = ""
     Case 2
     PileType_WHERE = " AND (Piles.[Pile Type])='Concrete Pile'"
     Case 3
     PileType_WHERE = " AND (Piles.[Pile Type])='H Steel Pile'"
     Case 4
     PileType_WHERE = " AND (Piles.[Pile Type])='Pipe Steel Pile'"
    End Select
End Sub

Private Sub Insitu_AfterUpdate()
    'Completes the Insitu test criteria depending on the selected test
    Select Case Insitu
     Case 1
     InsituTestType_WHERE = ""
     General_Insitu_WHERE = ""
     Case 2
     InsituTestType_WHERE = " AND ([InSitu Tests].[Test Type])='SPT'"
```

```
      General_Insitu_WHERE = " AND ((General.[Project ID])=([Query_Insitu].[Project ID]))"
      Case 3
      InsituTestType_WHERE = " AND ([InSitu Tests].[Test Type])='CPT'"
      General_Insitu_WHERE = " AND ((General.[Project ID])=([Query_Insitu].[Project ID]))"
      Case 4
      InsituTestType_WHERE = " AND ([InSitu Tests].[Test Type])='DMT'"
      General_Insitu_WHERE = " AND ((General.[Project ID])=([Query_Insitu].[Project ID]))"
      Case 5
      InsituTestType_WHERE = " AND ([InSitu Tests].[Test Type])='PMT'"
      General_Insitu_WHERE = " AND ((General.[Project ID])=([Query_Insitu].[Project ID]))"
    End Select
End Sub

Private Sub Load_AfterUpdate()
  'Completes the Load test criteria depending on the selected test
    Select Case Load
      Case 1
      LoadTestType_WHERE = ""
      Piles_LoadTest_WHERE = ""
      Shafts_LoadTest_WHERE = ""
      General_LoadTest_WHERE = ""
      Case 2
      LoadTestType_WHERE = " AND ([Load Tests].[Type of test])='Static Test'"
      Piles_LoadTest_WHERE = " AND ((Piles.[Pile ID])=[Query_Load].[Pile ID])"
      Shafts_LoadTest_WHERE = " AND ((Shafts.[Shaft ID])=[Query_Load].[Shaft ID])"
      General_LoadTest_WHERE = " AND ((General.[Project ID])=[Query_Load].[Project ID])"
      Case 3
      LoadTestType_WHERE = " AND ([Load Tests].[Type of test])='Statnamic Test'"
      Piles_LoadTest_WHERE = "AND ((Piles.[Pile ID])=[Query_Load].[Pile ID])"
      Shafts_LoadTest_WHERE = "AND ((Shafts.[Shaft ID])=[Query_Load].[Shaft ID])"
      General_LoadTest_WHERE = " AND ((General.[Project ID])=[Query_Load].[Project ID])"
      Case 4
      LoadTestType_WHERE = " AND ([Load Tests].[Type of test])='Osterberg Test'"
      Piles_LoadTest_WHERE = "AND ((Piles.[Pile ID])=[Query_Load].[Pile ID])"
      Shafts_LoadTest_WHERE = "AND ((Shafts.[Shaft ID])=[Query_Load].[Shaft ID])"
      General_LoadTest_WHERE = " AND ((General.[Project ID])=[Query_Load].[Project ID])"
    End Select
End Sub

Private Sub Max_Total_Length_AfterUpdate()
Dim Length As String
  'Calculates the Max Total Length in SI units
  Select Case (Me![Textbox_Units])
  Case "SI"
    Length = Me![Max Total Length]
  Case "English"
    Length = Me![Max Total Length] * 0.3048
  End Select
  'Creates Max Total Length Criteria
  If IsNull(Length) Then
    MaxTotalLength_WHERE_Piles = ""
    MaxTotalLength_WHERE_Shafts = ""
  Else
    MaxTotalLength_WHERE_Piles = " AND [Piles].[Total Length (m)] <= " & Length
    MaxTotalLength_WHERE_Shafts = " AND [Shafts].[Total Length (m)] <= " & Length
  End If
End Sub

Private Sub Min_Total_Length_AfterUpdate()
Dim Length As String
  'Calculates the Min Total Length in SI units
  Select Case (Me![Textbox_Units])
```

```
    Case "SI"
      Length = Me![Min Total Length]
    Case "English"
      Length = Me![Min Total Length] * 0.3048
    End Select
    'Creates Min Total Length Criteria
      If IsNull(Length) Then
      MinTotalLength_WHERE_Piles = ""
      MinTotalLength_WHERE_Shafts = ""
    Else
      MinTotalLength_WHERE_Piles = " AND [Piles].[Total Length (m)] >= " & Length
      MinTotalLength_WHERE_Shafts = " AND [Shafts].[Total Length (m)] >= " & Length
    End If
End Sub

Private Sub Max_Embedded_Length_AfterUpdate()
Dim Length As String
    'Calculates the Max Embedded Length in SI units
    Select Case (Me![Textbox_Units])
    Case "SI"
      Length = Me![Max Embedded Length]
    Case "English"
      Length = Me![Max Embedded Length] * 0.3048
    End Select
    'Creates Max Embedded Length Criteria
      If IsNull(Length) Then
      MaxEmbLength_WHERE_Piles = ""
      MaxEmbLength_WHERE_Shafts = ""
    Else
      MaxEmbLength_WHERE_Piles = " AND [Piles].[Embedded Length (m)] <= " & Length
      MaxEmbLength_WHERE_Shafts = " AND [Shafts].[Embedded Length (m)] <= " & Length
    End If
End Sub

Private Sub Min_Embedded_Length_AfterUpdate()
Dim Length As String
    'Calculates the Min Embedded Length in SI units
    Select Case (Me![Textbox_Units])
    Case "SI"
      Length = Me![Min Embedded Length]
    Case "English"
      Length = Me![Min Embedded Length] * 0.3048
    End Select
    'Creates Min Embedded Length Criteria
        If IsNull(Length) Then
      MinEmbLength_WHERE_Piles = ""
      MinEmbLength_WHERE_Shafts = ""
    Else
      MinEmbLength_WHERE_Piles = " AND [Piles].[Embedded Length (m)] >= " & Length
      MinEmbLength_WHERE_Shafts = " AND [Shafts].[Embedded Length (m)] >= " & Length
    End If
End Sub

Private Sub Max_Diameter_AfterUpdate()
Dim Diameter As String
    'Calculates the Max Diameter in SI units
    Select Case (Me![Textbox_Units])
    Case "SI"
      Diameter = Me![Max Diameter]
    Case "English"
      Diameter = Me![Max Diameter] * 2.54
    End Select
```

```
    'Creates Max Diameter Criteria
    If IsNull(Diameter) Then
        MaxDiameter_WHERE_Piles = ""
        MaxDiameter_WHERE_Shafts = ""
    Else
        MaxDiameter_WHERE_Piles = " AND [Piles].[Diameter (mm)] <= " & Diameter
        MaxDiameter_WHERE_Shafts = " AND [Shafts].[Diameter (mm)] <= " & Diameter
    End If
End Sub

Private Sub Min_Diameter_AfterUpdate()
Dim Diameter As String
    'Calculates the Min Diameter in SI units
    Select Case (Me![Textbox_Units])
    Case "SI"
        Diameter = Me![Min Diameter]
    Case "English"
        Diameter = Me![Min Diameter] * 2.54
    End Select
    'Creates Min Diameter Criteria
    If IsNull(Diameter) Then
        MinDiameter_WHERE_Piles = ""
        MinDiameter_WHERE_Shafts = ""
    Else
        MinDiameter_WHERE_Piles = " AND [Piles].[Diameter (mm)] >= " & Diameter
        MinDiameter_WHERE_Shafts = " AND [Shafts].[Diameter (mm)] >= " & Diameter
    End If
End Sub

Private Sub Driving_Click()
    If Me![Driving] = True Then
        'Creates Driving Criteria
        Driving_WHERE = " AND ([Piles].[Driving Data])= 'Yes'"
    Else
        Driving_WHERE = ""
    End If
End Sub

Private Sub GRL_Click()
    If Me![GRL] = True Then
        'Creates GRL Criteria
      GRL_WHERE = " AND ([Piles].[GRL Data])= 'Yes'"
    Else
        GRL_WHERE = ""
    End If
End Sub

'Closes Form
Private Sub MainMenuButton_Click()
On Error Resume Next
    DoCmd.Close acForm, "frmSearch"
End Sub


Private Sub Units_AfterUpdate()
    'Enables the controls involving units
    Me![Min Total Length].Enabled = True
    Me![Max Total Length].Enabled = True
    Me![Min Embedded Length].Enabled = True
    Me![Max Embedded Length].Enabled = True
    Me![Min Diameter].Enabled = True
    Me![Max Diameter].Enabled = True
```

```
    'Resets all controls involving Units
    Me![Max Total Length] = Null
    Me![Min Total Length] = Null
    Me![Max Embedded Length] = Null
    Me![Min Embedded Length] = Null
    Me![Max Diameter] = Null
    Me![Min Diameter] = Null
     'Displays all labels with the selected set of units
     Select Case Me![Units]
      Case 1
        Me![Textbox_Units].Value = "SI"
        Me![Max Total Length Label].Caption = "Max Total Length (m)"
        Me![Min Total Length Label].Caption = "Min Total Length (m)"
        Me![Max Embedded Length Label].Caption = "Max Embedded Length (m)"
        Me![Min Embedded Length Label].Caption = "Min Embedded Length (m)"
        Me![Max Diameter Label].Caption = "Max Diameter (mm)"
        Me![Min Diameter label].Caption = "Min Diameter (mm)"
      Case 2
        Me![Textbox_Units].Value = "English"
        Me![Max Total Length Label].Caption = "Max Total Length (ft)"
        Me![Min Total Length Label].Caption = "Min Total Length (ft)"
        Me![Max Embedded Length Label].Caption = "Max Embedded Length (ft)"
        Me![Min Embedded Length Label].Caption = "Min Embedded Length (ft)"
        Me![Max Diameter Label].Caption = "Max Diameter (in)"
        Me![Min Diameter label].Caption = "Min Diameter (in)"
     End Select
End Sub
```

frmPrint

```
Option Compare Database
Option Explicit
Dim Print_Report As String

Private Sub CancelButton_Click()
'Closes the form
DoCmd.Close
End Sub

Private Sub Form_Current()
'Disables the Units and Print Options for reports that do not involve units or tabulated data
 If (Print_Units = 1) Then
     Me!Units.Enabled = False
   Else
     Me!Units.Enabled = True
   End If
   If (Print_Excel = 1) Then
     Me!Option35.Enabled = False
   Else
     Me!Option35.Enabled = True
   End If
End Sub

Private Sub OKButton_Click()
On Error Resume Next
'Determines the name of the report depending on the units chosen
   Select Case (Me![Textbox_Units])
     Case "English"
       Print_Report = Report_Name & "_English"
     Case Else
       Print_Report = Report_Name
   End Select
  'Applies a filter so that only the data from the current record is included in the report
  DoCmd.OpenReport Print_Report, acViewPreview, , Print_Filter
   'Prepares the report depending on the chosen print option
   Select Case (Me![Textbox_Option])
     Case "Excel"
       DoCmd.OutputTo acOutputReport, , acFormatXLS, , True
     Case "Word"
       DoCmd.OutputTo acOutputReport, , acFormatRTF, , True
     Case Else
       DoCmd.PrintOut
  End Select
 'Closes the report and the form frmPrint
 DoCmd.Close acReport, Print_Report
 DoCmd.Close acForm, "frmPrint"
End Sub

Private Sub Units_AfterUpdate()
   'Updates the Textbox_Units control
   Select Case (Me!Units)
     Case 1
       Me![Textbox_Units].Value = "SI"
     Case 2
       Me![Textbox_Units].Value = "English"
   End Select
End Sub
```

```
Private Sub Option_AfterUpdate()
   'Updates the Textbox_Option control
   Select Case (Me!Option)
      Case 1
         Me![Textbox_Option].Value = "Print"
      Case 2
         Me![Textbox_Option].Value = "Excel"
      Case 3
         Me![Textbox_Option].Value = "Word"
   End Select
End Sub
```

APPENDIX C

REPORT ON PILE AND SHAFT BASED ACCESS DATABASE

CHAPTER 2.
DATABASE ARCHITECTURE

The Deep Foundations Database is contained in a single Microsoft Access 97 file; it contains several tables, forms, queries and reports. The data are normally stored in the tables by means of forms. The forms are also used to edit and display the data and to interact with the user by controlling the opening and closing of reports and other forms. The reports provide the user with the ability to print the data or export it to an Excel spreadsheet or a Word document. The queries are used to display search results or as base tables to create the reports.

2.1. Database Tables

A total of eighteen tables are included in the database. Each table stores a particular set of related data. Each row in a table represents a record while each column represents a field. The reason for having several tables instead of grouping all the information in one table is that for a given element, some of the fields require just one record while others require several records. The project name is an example of a single-record field while an SPT boring log represents a multi-record field.

The table contents are presented in Figures 2.1 to 2.4. The General table, Figure 2.1, constitutes the main index and contains the project information. Elements in the database are divided into two main groups, driven piles and drilled shafts. Driven piles are then subdivided into concrete piles, steel H piles and steel pipe piles. The Piles table, Figure 2.2, contains the information common to all the driven piles and there is one table for each driven pile subtype, namely Piles Concrete, Piles H Steel and Piles Pipe Steel. Each of these tables contains data related to the pile geometry and material properties.

The data related to the pile penetration are stored in two tables, Figure 2.2; the Piles Driving table contains information that can be stored in a single-record field while the Piles Penetration table contains the penetration log. Similarly, the Pile Driving Analyzer (PDA) data are stored in two tables; the GRL Data table contains all the information that can be stored in a single-record field while the GRL PDA Trace table contains the PDA log.

The Shafts table, Figure 2.3, contains the data related to the drilled shaft geometry and material properties. The analysis of drilled shafts founded in rock (Intermediate Geomaterials or IGM) using capacity prediction software such as SHAFT98 requires some additional parameters, which can be stored in a multi-record table named Shafts IGM.

The InSitu Tests table, Figure 2.4, contains the data related to the insitu tests that can be stored in single-record fields. Also, since several insitu tests can be related to a single element, this table also functions as an index for the Insitu SPT, Insitu CPT and Insitu DMT tables. In like manner, the Load Tests table, Figure 2.4, contains the data related to the load tests that can be stored in single-record fields and provides an index for the Load Test Static and Load Test Statnamic tables.

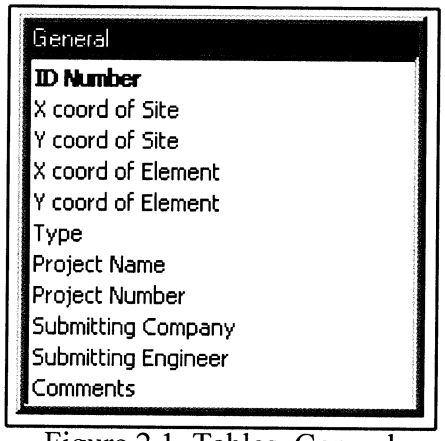


Figure 2.1. Tables, General

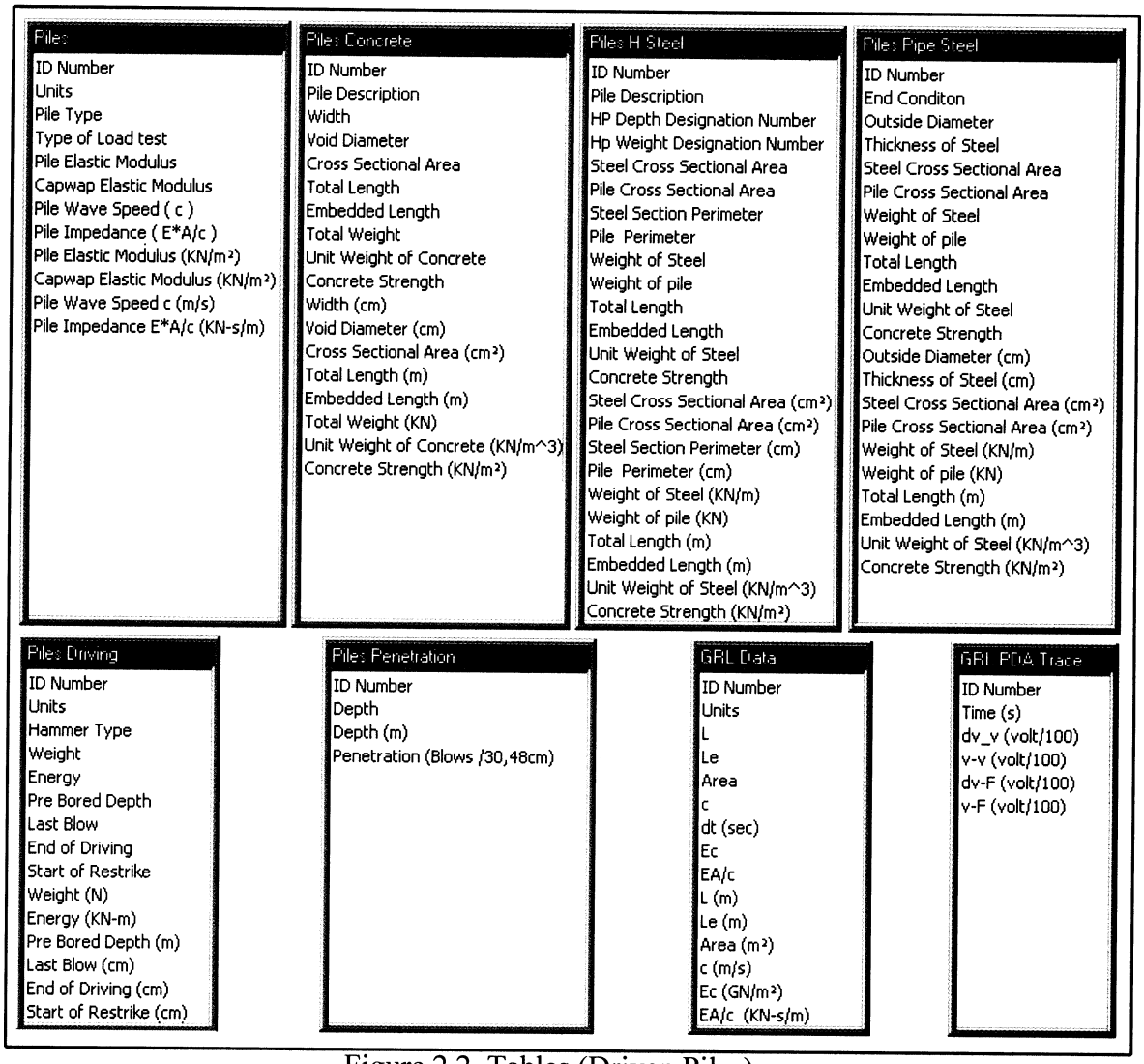


Figure 2.2 Tables (Driven Piles)

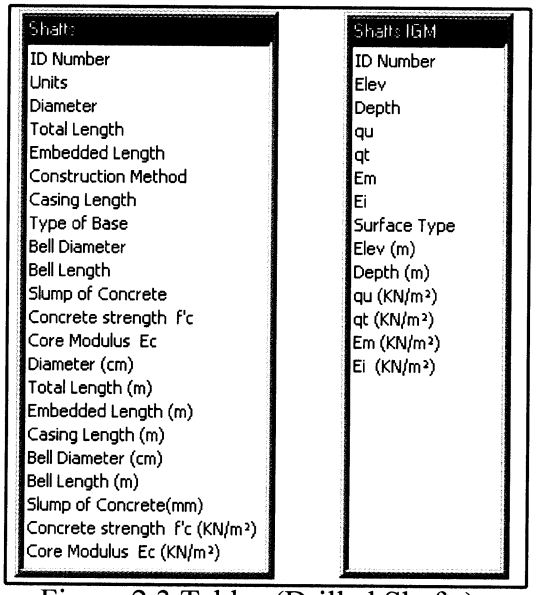


Figure 2.3 Tables (Drilled Shafts)

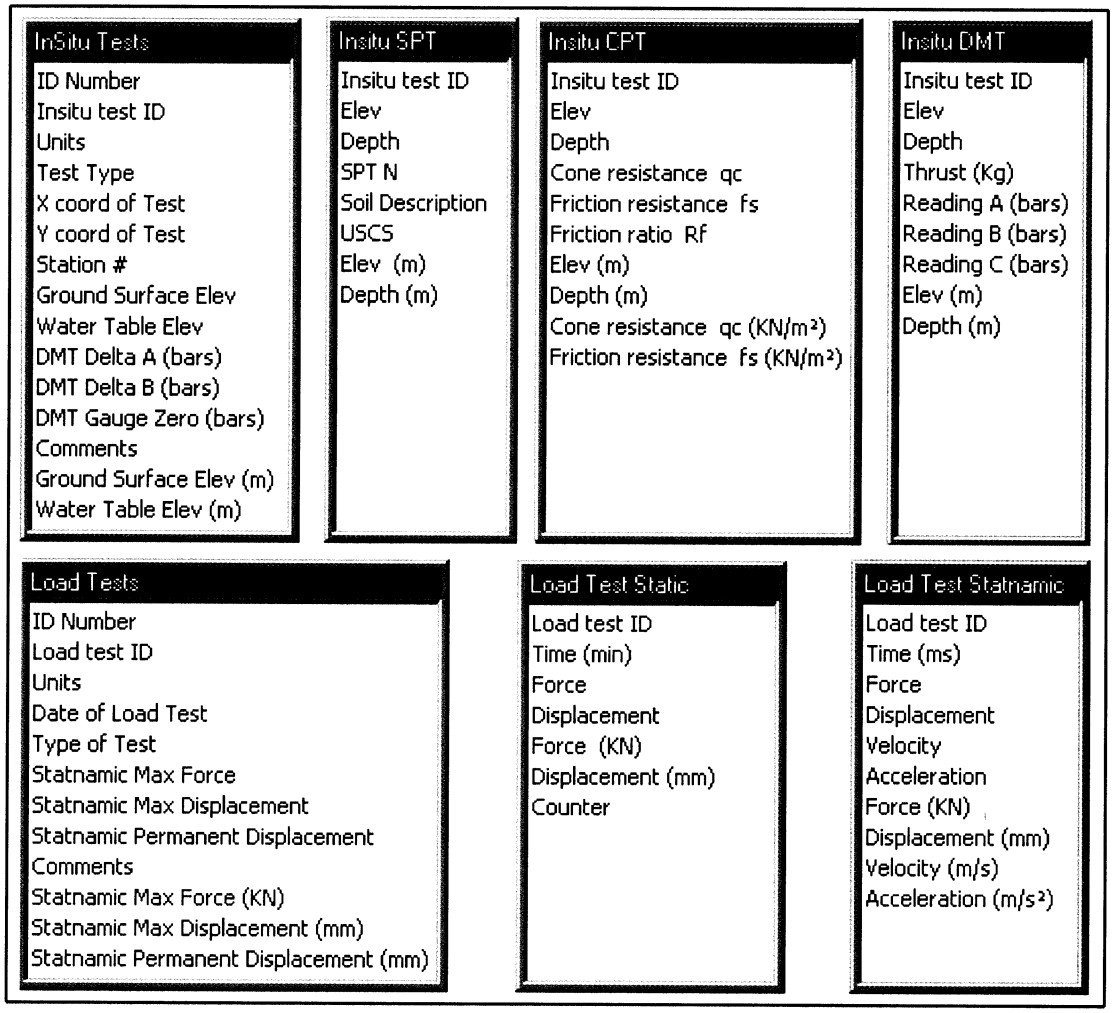


Figure 2.4. Tables (Insitu Tests and Load Tests)

Relationships between the tables are defined using index fields to relate the data in the different tables and to assure data integrity. These relationships are presented in Figure 2.5. The General table is the primary table of the database. Many of the other tables in the database are directly related to it through an index field named ID Number. Other tables are related to associated index tables, which are then related to the General table. For example, the load tests tables are related to the Load Tests table through the Load Test ID index field, and this table is related to the General table through the ID Number field. Similarly, the insitu tests tables are related to the InSitu Tests table, which is related to the General table. Each time a record is erased from an index table, the

database automatically removes the record from the related tables, assuring the integrity of the data. An infinity symbol at the end of a relationship line means that the related table can have more than one record for each record in the primary table.

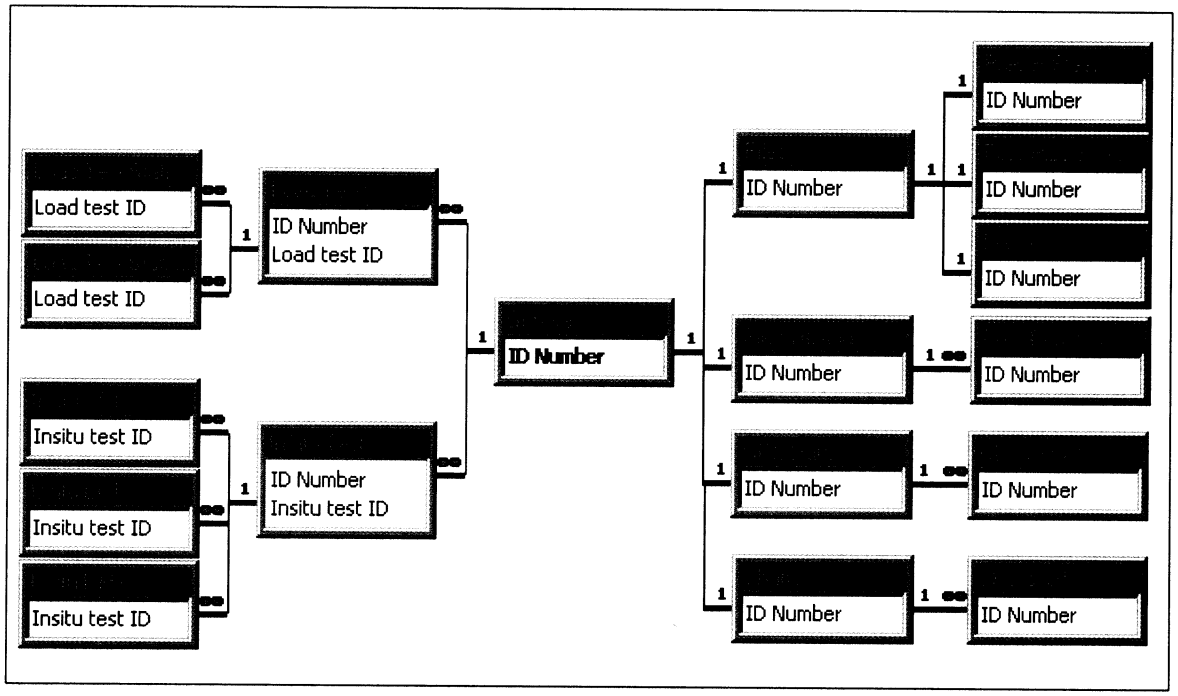


Figure 2.5 Table Relationships

2.2. Database Forms

The forms in the Deep Foundations Database can be classified into two groups; those used to add, edit or view data in the database tables and those used to control database operations, allowing the user to perform common tasks such as printing.

2.2.1. Add/Edit Forms

Data processing forms can operate directly on the tables when only one table is involved. However, if a single form is used for adding or editing data in more than one table, a query containing the fields from the different tables has to be created and the form has to be based on the query. Alternatively, a form based on a single table and

containing subforms based on other tables can be implemented. The second approach was considered when creating the forms for adding, editing and displaying data in this database. The forms are related one to another in the same manner as the tables. The master forms are based on primary tables and the child forms (or subforms) are based on the related tables.

The forms for adding and editing the data are presented in Figures 2.6 through 2.18. All the data are entered through a single form named frmGeneral, which contains several subforms related to the different database tables. The accessibility of the subforms is controlled by means of tabs. For example, when the user selects "Driven Pile" from the Element Type control, six tabs are displayed which can then be chosen in turn for the inputting of data. If "Drilled Shaft" were selected only four relevant subform tabs would be displayed. Similarly, the availability of the subforms for different pile shapes, insitu tests and load tests are controlled by means of the Pile Type, Insitu Test Type and Load Test Type controls respectively. These input forms will be discussed in more detail in Chapter 3.

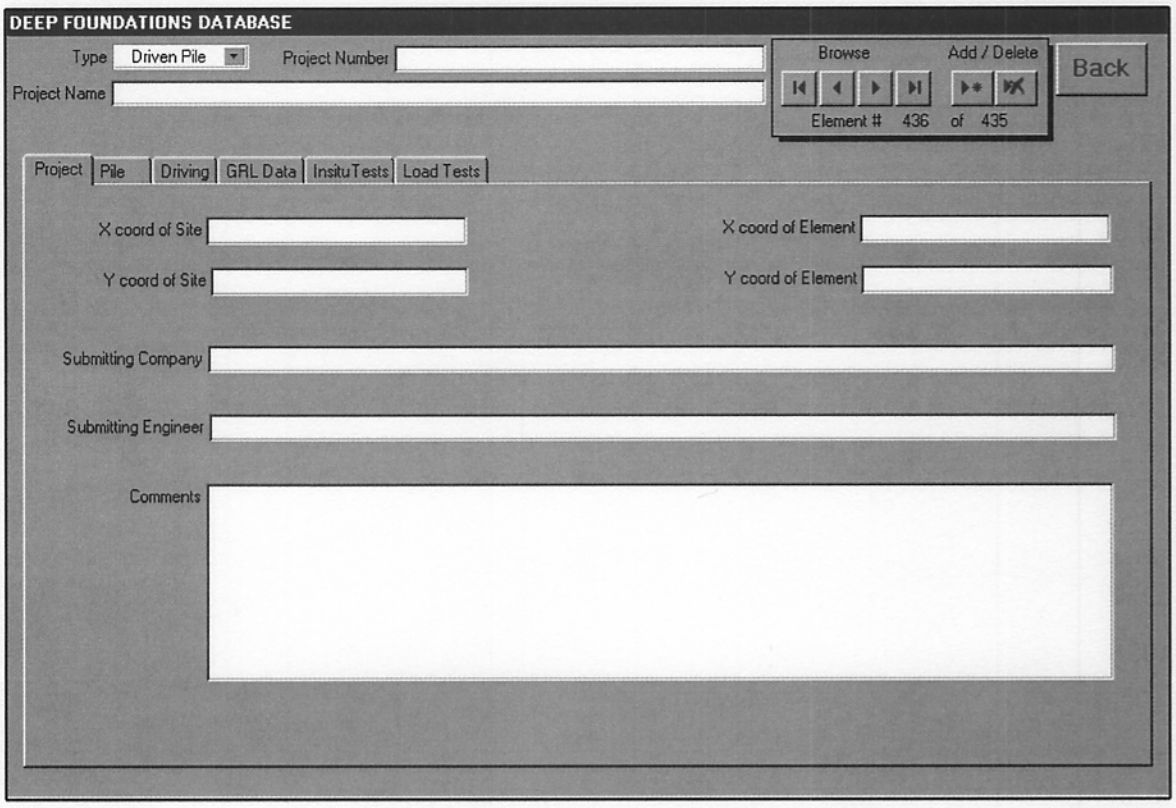


Figure 2.6 Add/Edit form (General, Driven Piles)

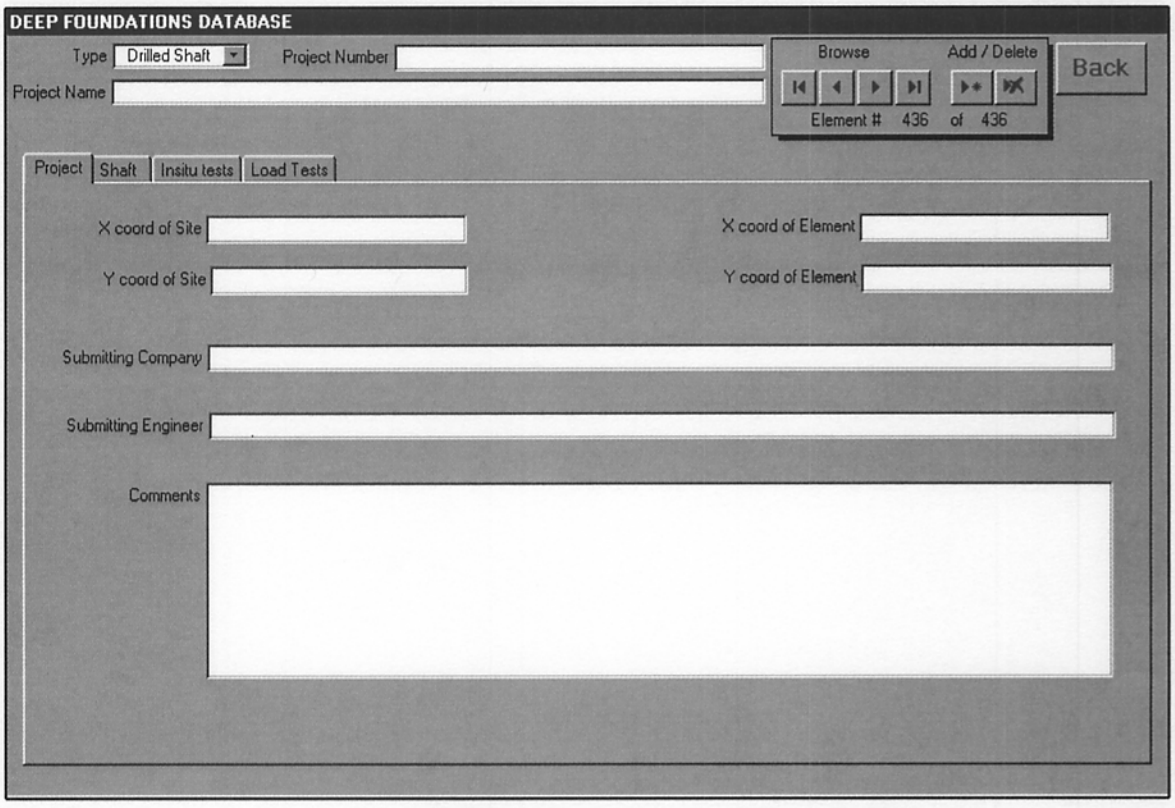


Figure 2.7 Add/Edit form (General, Drilled Shafts)

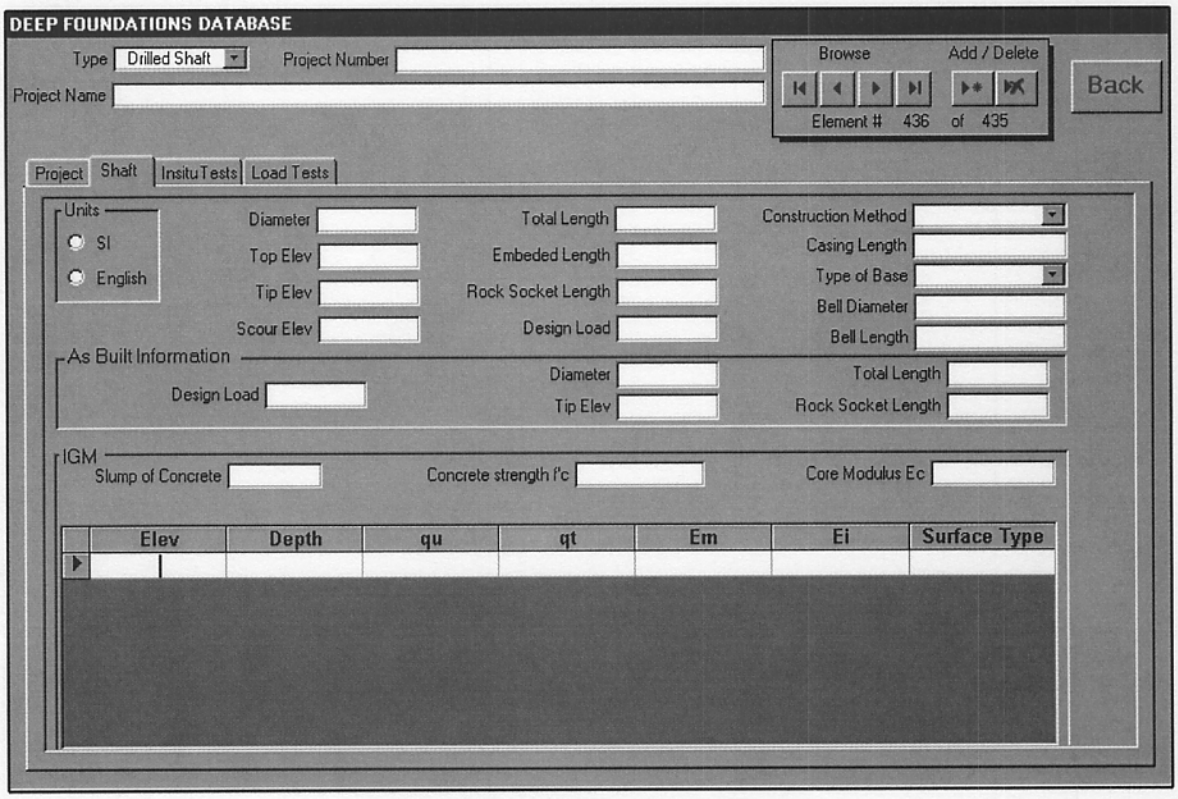


Figure 2.8 Add/Edit form (Shafts)

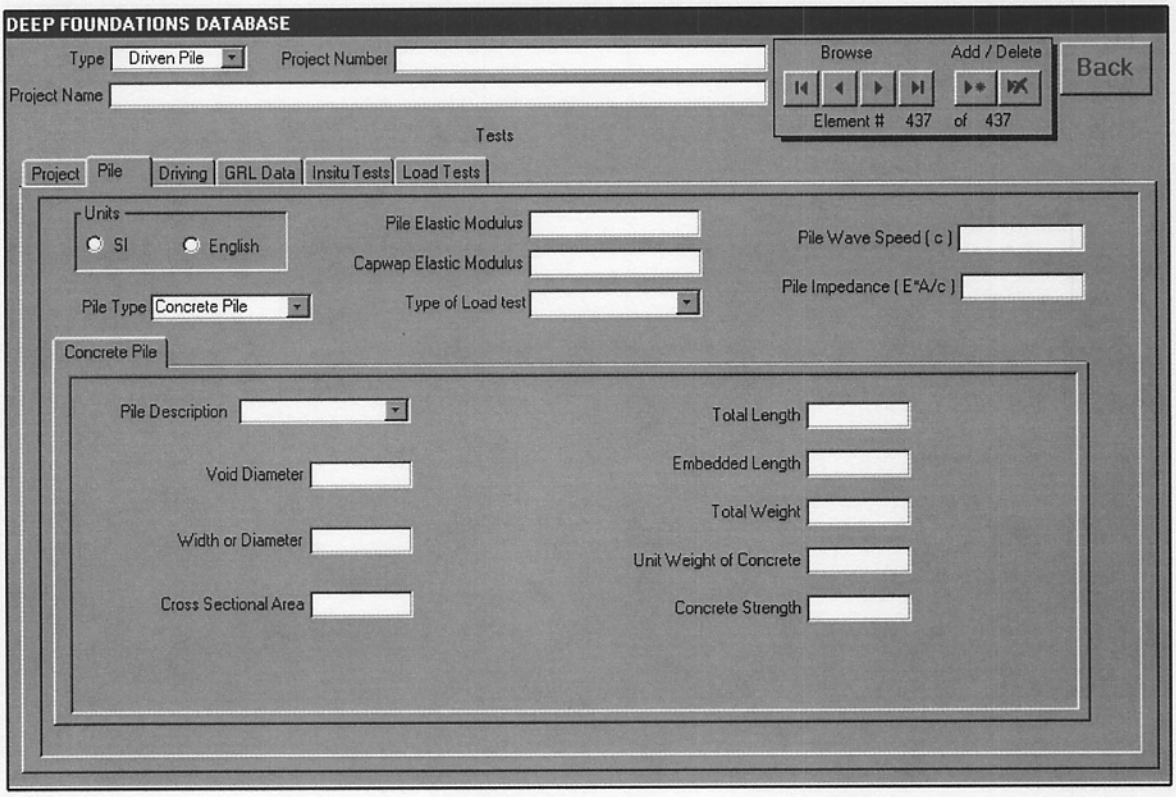


Figure 2.9 Add/Edit form (Piles, Concrete Piles)

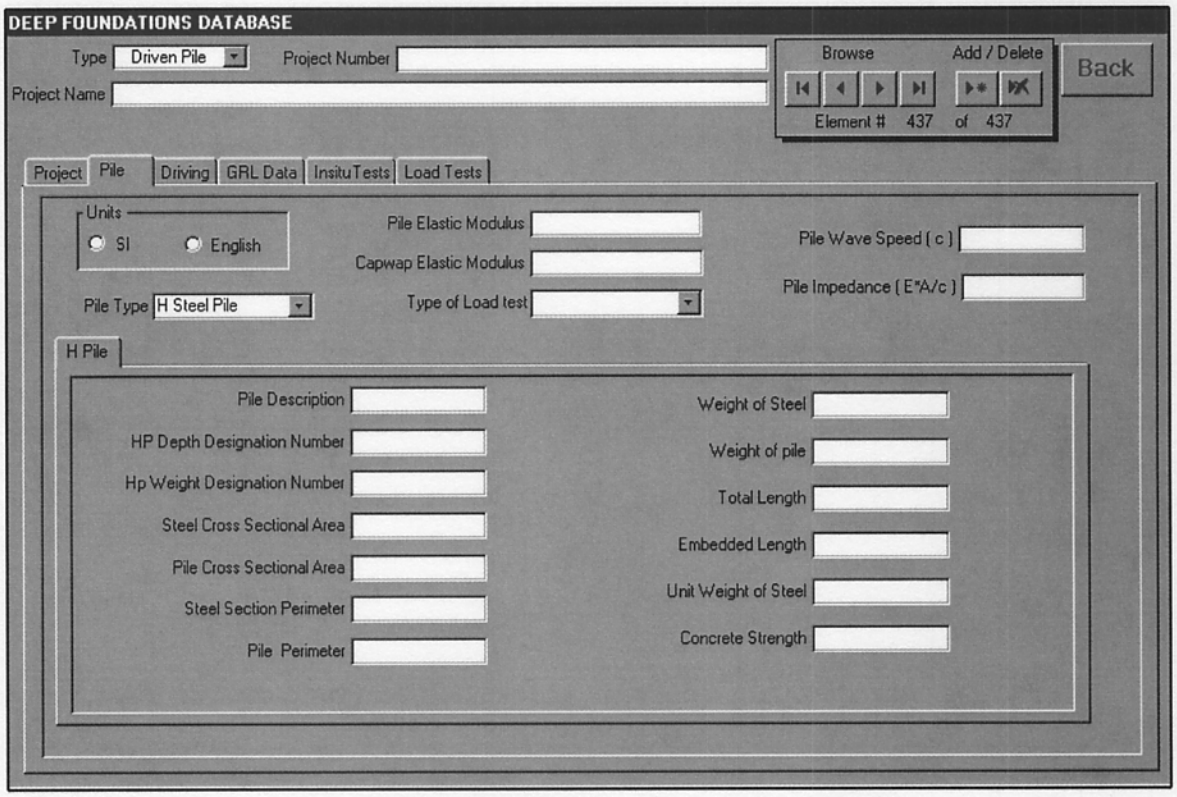


Figure 2.10 Add/Edit form (Piles, H Steel Piles)

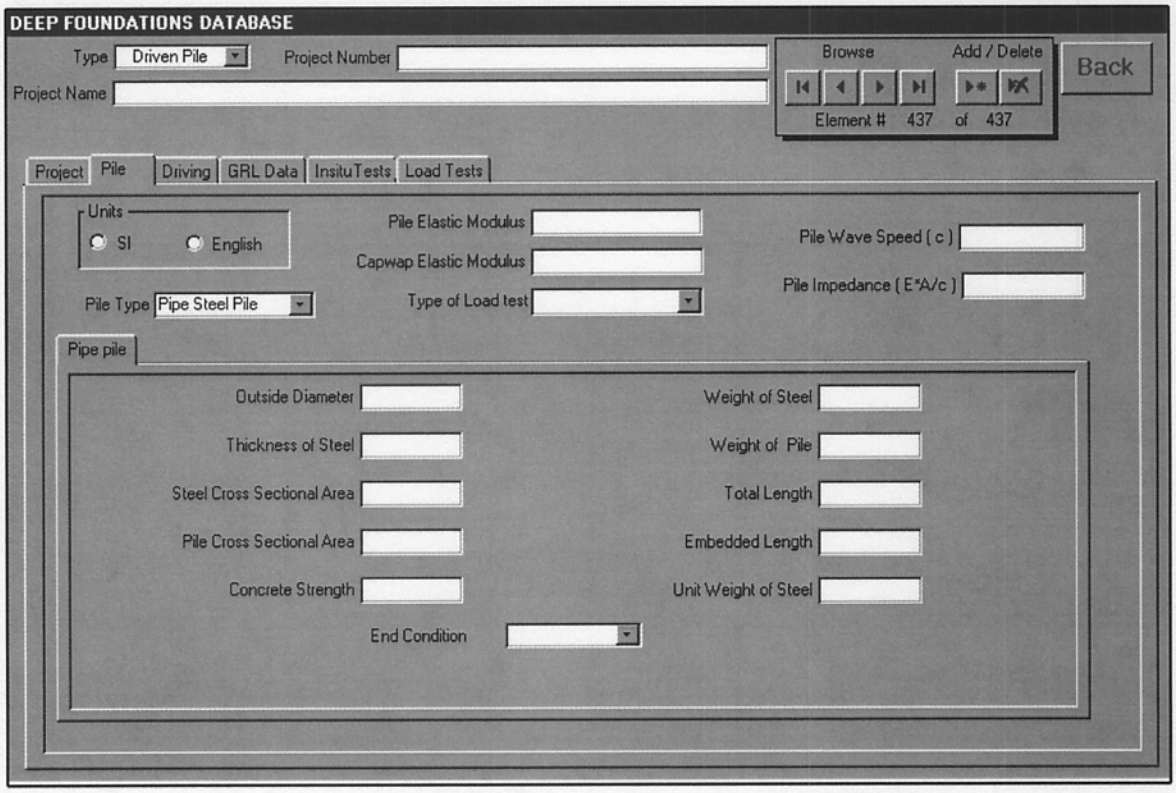


Figure 2.11 Add/Edit form (Piles, Pipe Steel Piles)

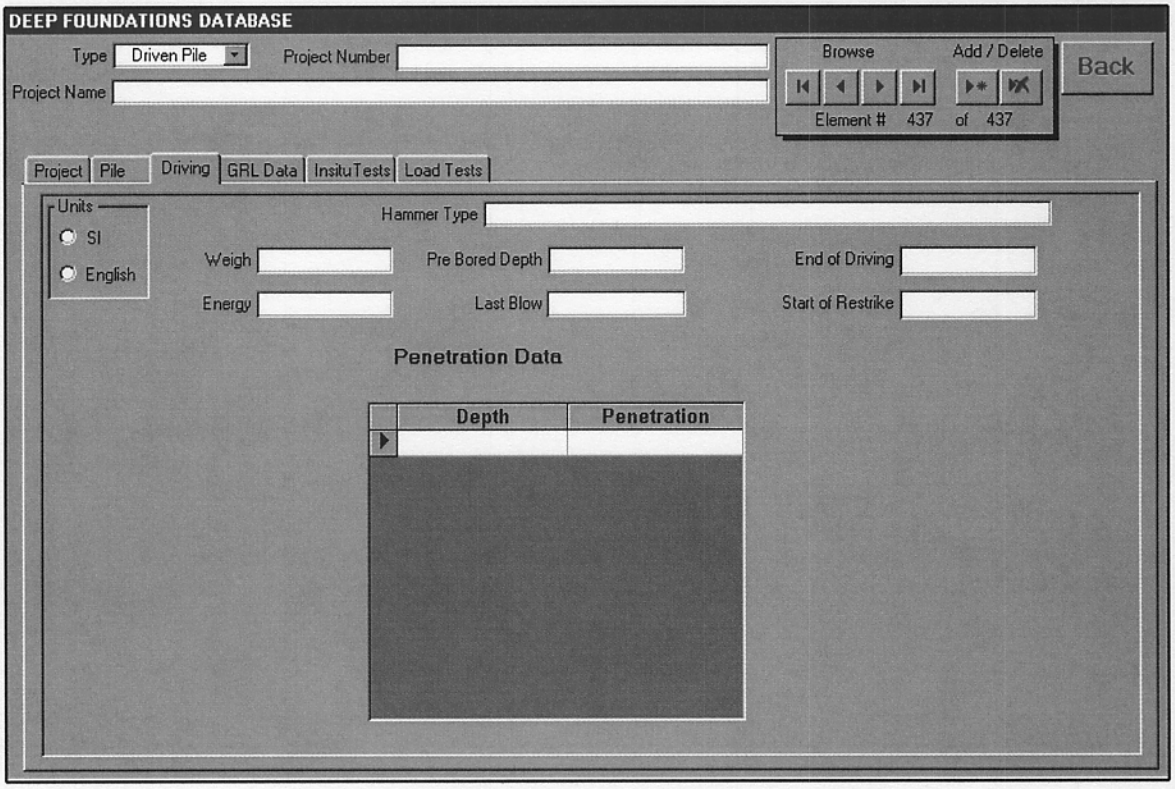


Figure 2.12 Add/Edit form (Pile Driving Data)

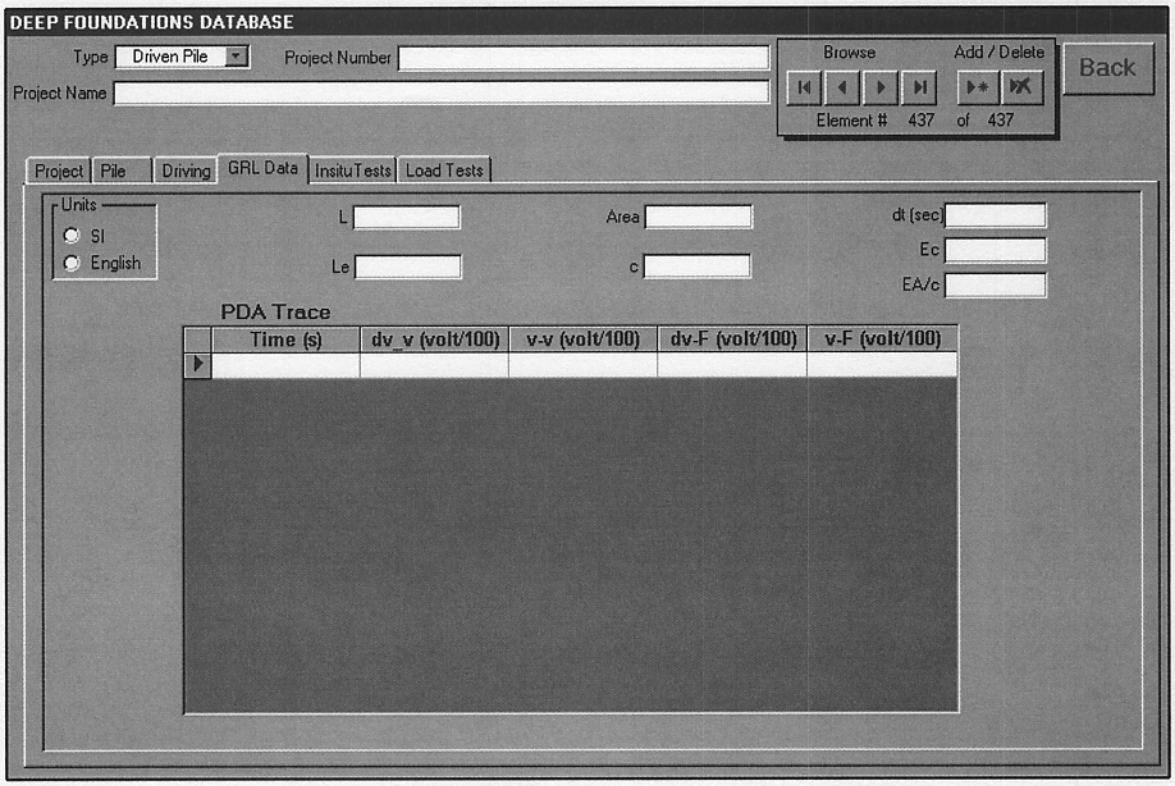


Figure 2.13 Add/Edit form (GRL Driving Data)

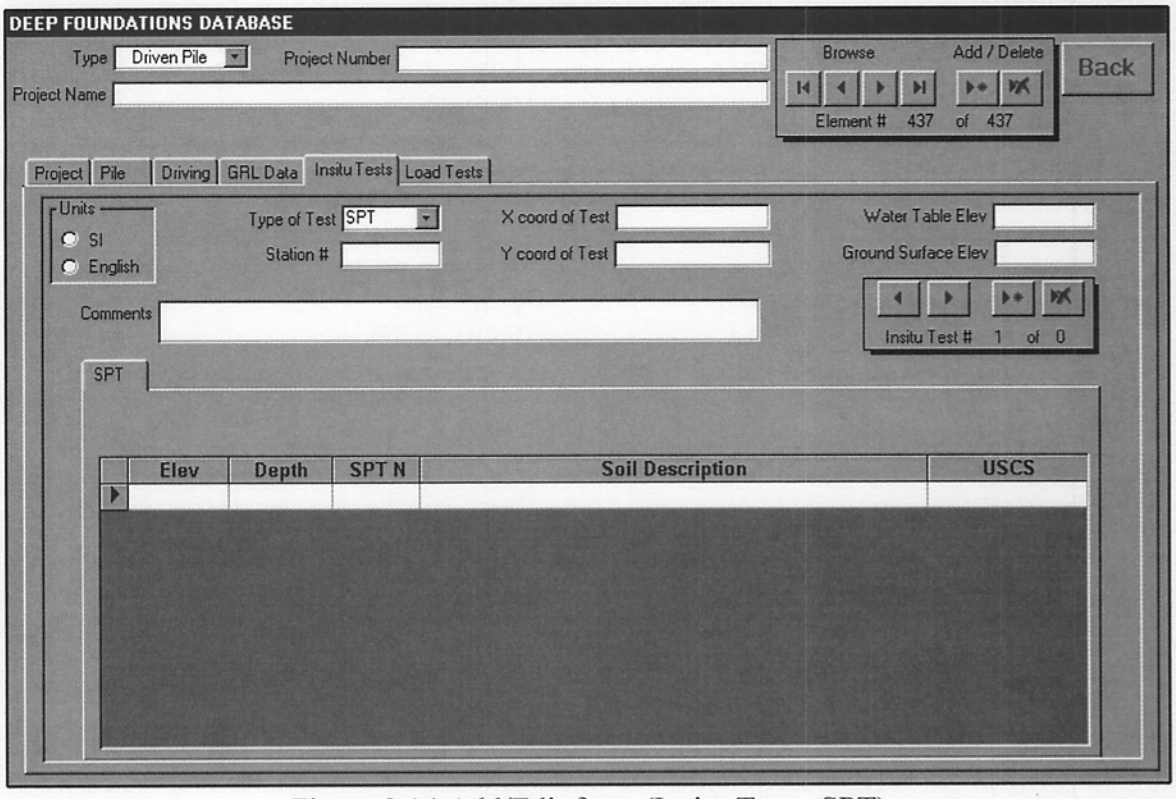


Figure 2.14 Add/Edit form (Insitu Tests, SPT)

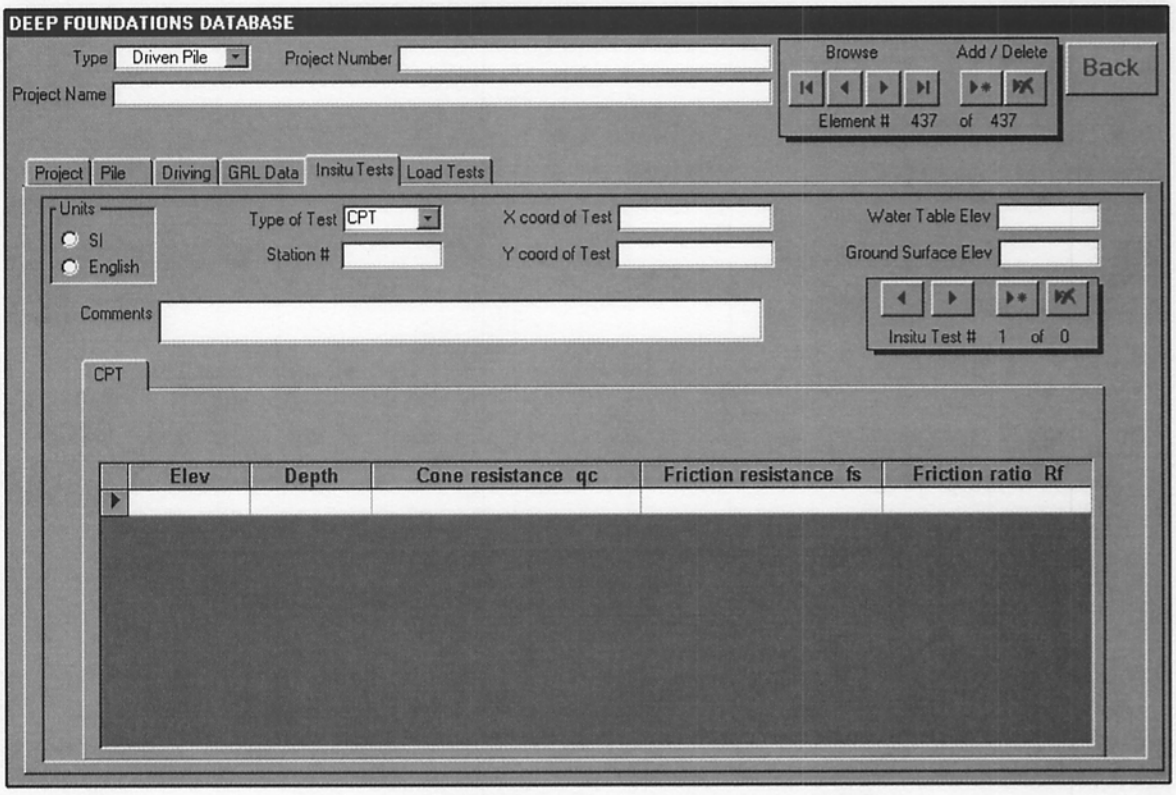


Figure 2.15 Add/Edit form (Insitu Tests, CPT)

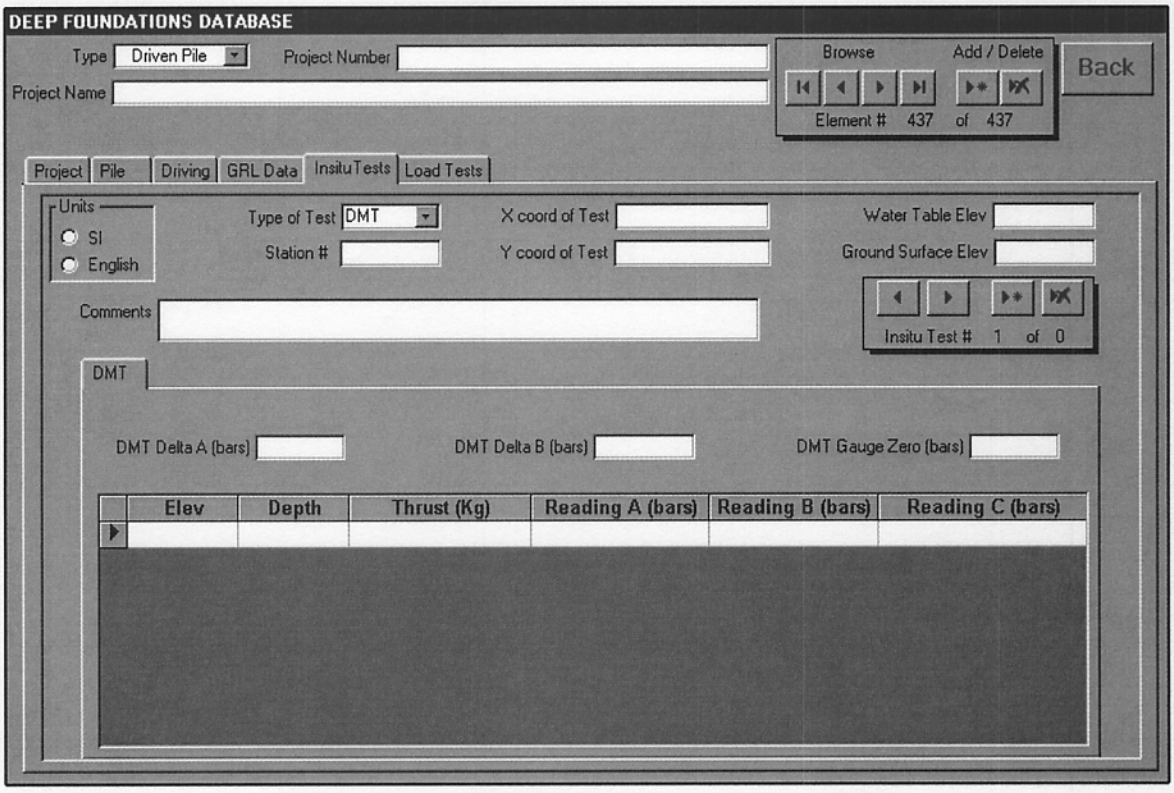


Figure 2.16 Add/Edit form (Insitu Tests, DMT)

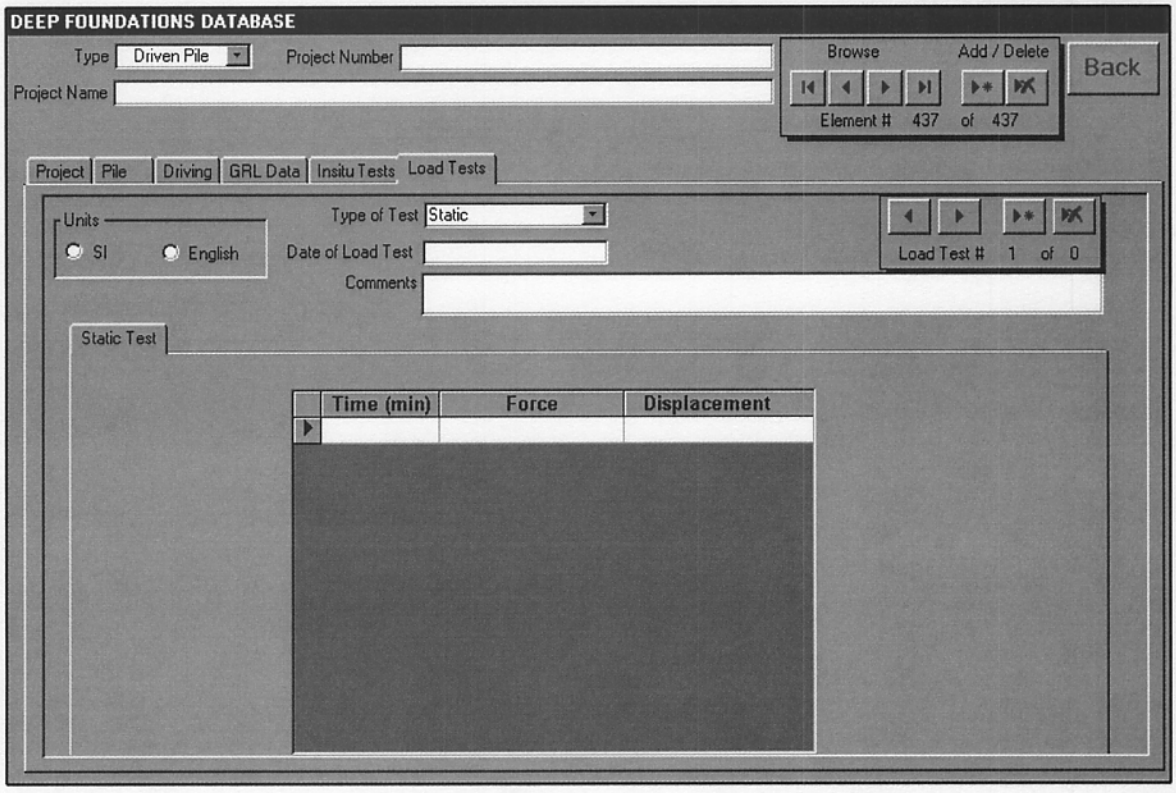


Figure 2.17 Add/Edit form (Load Tests, Static)

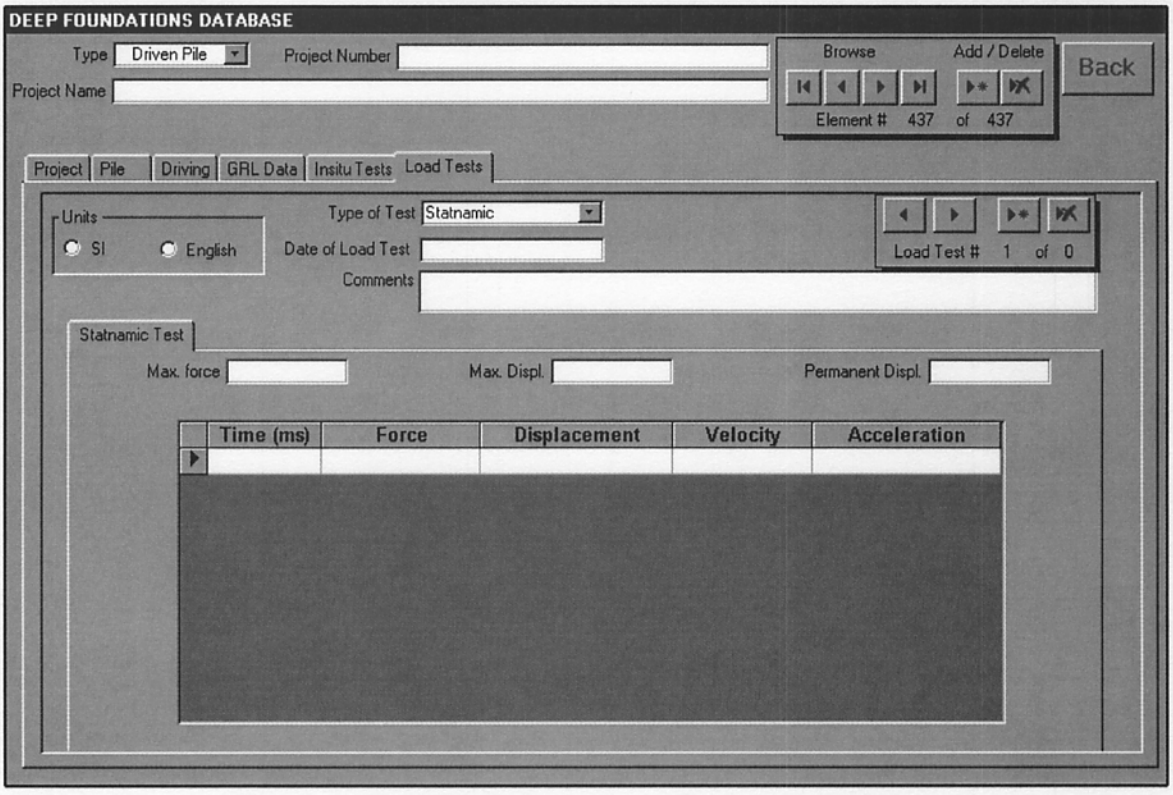


Figure 2.18 Add/Edit form (Load Tests, Statnamic)

2.2.2. Data Display Forms

Most of the forms used for adding and editing the data on the database tables are also used for displaying search results. Search operations are performed by means of a select query. The resulting query includes all the fields in the general table for those records that match the searching criteria. When displaying the search results, the source table for the main form is the Search query instead of the General table; the relationships between the forms remain unchanged but the primary table is no longer the General table but the Search query. Consequently, the data have to be displayed in read only mode since changes made to the data on the Data Display form would not be recorded on the General table. The navigation buttons enclosed in a box at the upper right corner of the main form had to be modified to comply with the read only view mode. That is, the buttons used for adding new records and deleting existing records are missing and the

number of records matching the search criteria is displayed instead of the total number of records in the database. The same modifications were performed to the Insitu Tests and Load Tests subforms. In addition, command buttons for printing or editing the current record and for returning to the search form are available at the bottom of the main form.

The Data Display form is shown in Figure 2.19; the Insitu Tests tab was selected in order to illustrate the changes made to the subforms.

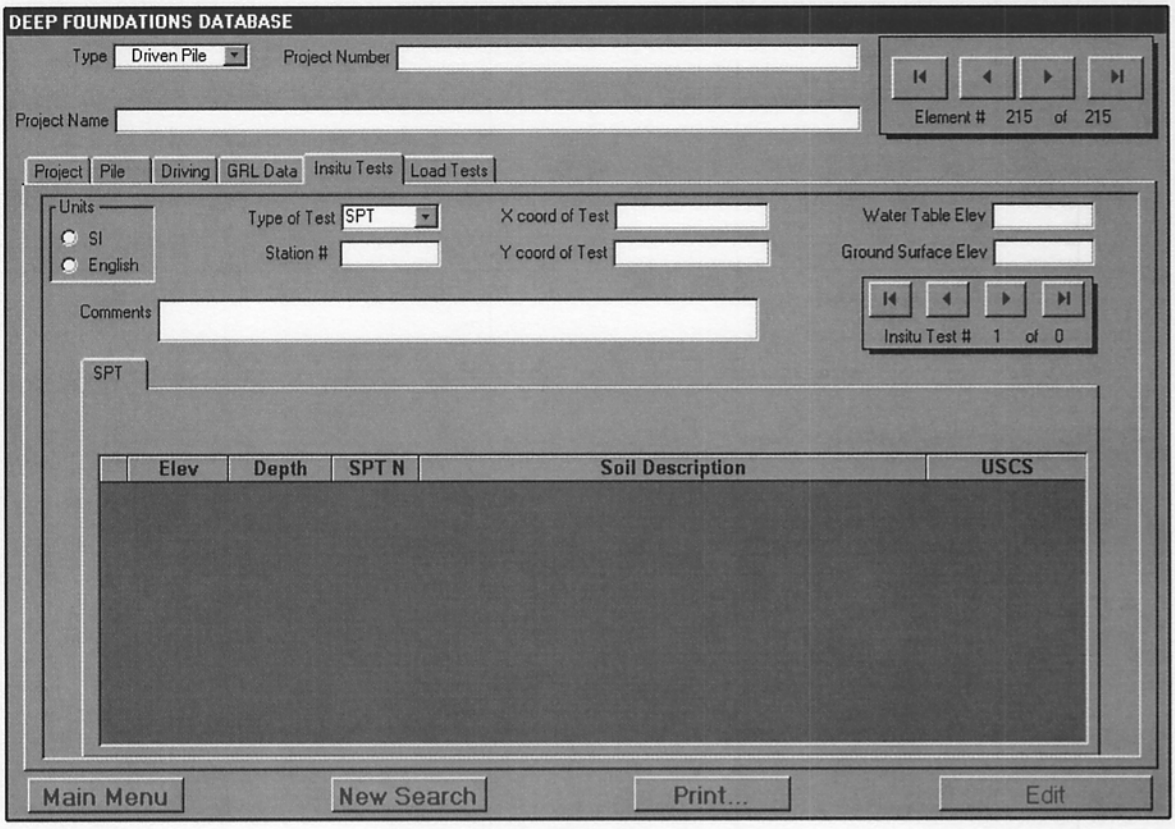


Figure 2.19 Data Display form

2.2.3. Database Control Forms

Three forms of this kind are included in the database, namely the Menu form, the Search form and the Print form. The Menu form identifies the database and displays the main menu, the Search form displays a set of options for the searching criteria and the

Print form provides a set of options for the data output. The Menu, Search and Print forms are presented in Figures 2.20, 2.21 and 2.22 respectively.

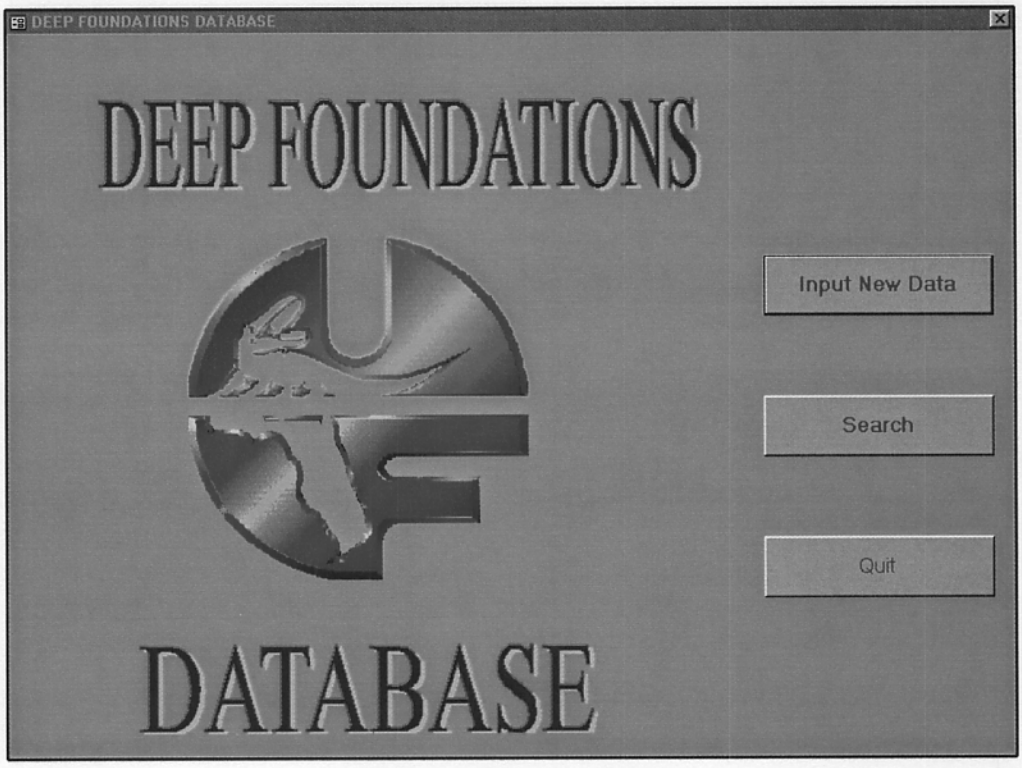


Figure 2.20 Menu form

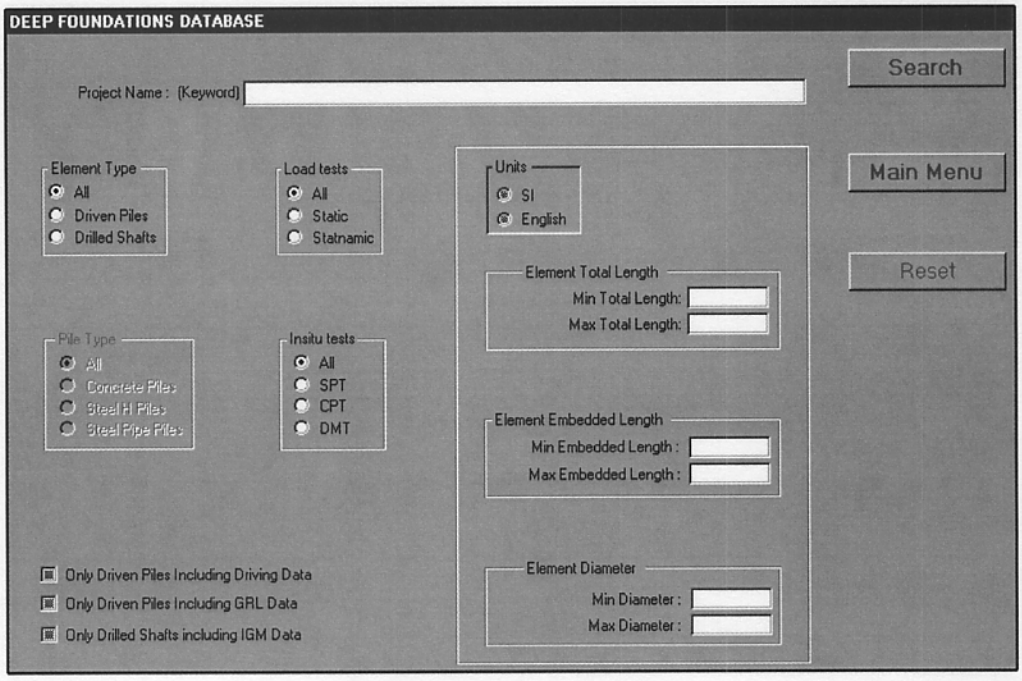


Figure 2.21 Search form

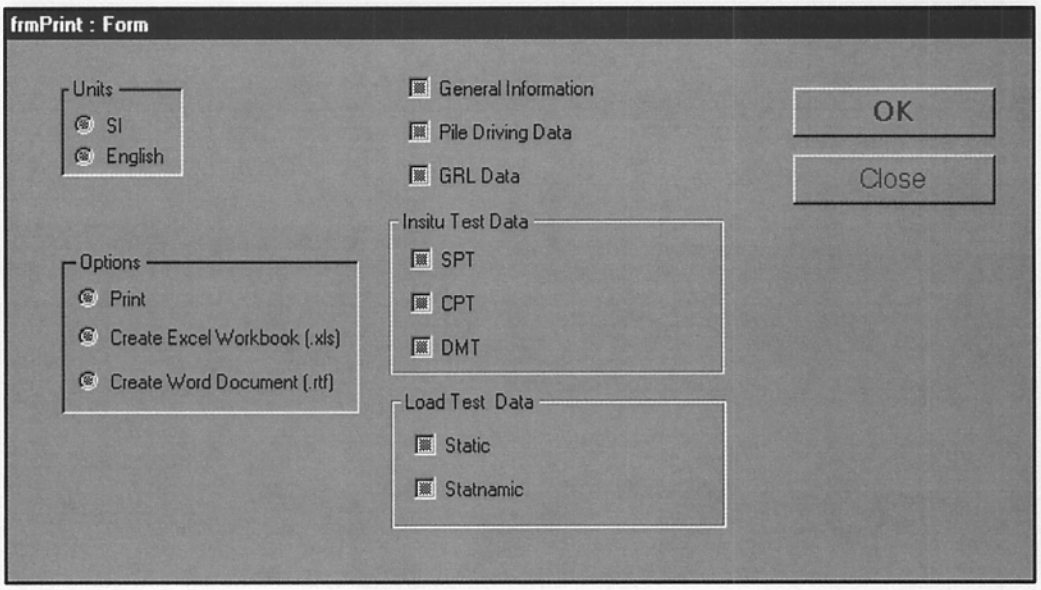


Figure 2.22 Print form

2.3. Database Reports

The Deep Foundations Database uses Access reports to output the data; reports can be printed or exported to other applications. The data are divided into twelve sections and there is one report for each section, namely Concrete Pile Report, H Pile Report, Pipe Pile Report, Driving Report, GRL Data Report, Shafts Report, IGM Report, SPT Report, CPT Report, DMT Report, Static Test Report and Statnamic Test Report. A query is designed to select the data required by each report from the different database tables and group them in a single table; the query will then function as the source data for the related report.

2.4. Database Queries

The Deep Foundations Database uses select queries for all searching and printing operations. Select queries are tables that display only the records that meet specified criteria. Queries in Access can be created graphically using the design interface provided by the program (Query by example "QBE") or by including Structured Query Language

"SQL" statements in Visual Basic Modules. All the queries in this database were defined using SQL.

The search process involves a single query named qrySearch. Each time the user clicks the Search command button in the corresponding form the existing qrySearch is erased and a new query with the same name is created from an SQL statement based on the criteria specified by the user. The Search query retrieves the information from all the fields in the General Table corresponding to the records that meet the specified criteria; however, the search criteria usually involve fields from several tables. The resulting query functions as a source table for the data displaying main form.

The print process involves twelve different queries, each functions as the source table for its related report as mentioned in the previous section. Consequently, each query has to contain all the fields required in its associated report. Each time the user clicks the Print command button in the corresponding form, all the existing queries are erased and a new query is created for each of the sections of the data that the user chooses to print. The name of its related report preceded by the word "qry_" is assigned to each new query (e.g., qry_SPT Report).

2.5. Database Units

The Deep Foundations Database allows the use of two different sets of units, English and SI. All the fields that involve units are stored twice in their corresponding tables, as entered through the forms (raw data, English or SI) and as converted to SI units. In addition, the table contains a field specifying the set of units used to input the data. The purpose of this is to have all the data stored in the same set of units to be able to query it, and still have the ability to display the data in their original format.

When data are entered through the specified form, they are automatically converted to SI units and stored in a different control on the form by means of a Visual Basic module. The input control is linked to the raw data field while the calculated control is linked to the SI field in the corresponding table. The calculated fields are not shown in form view.

The user has the ability to define the search criteria in either set of units. The criteria are then converted to SI units and the search query selects the records based on the fields containing the data in SI units. However, the Data Display forms are linked to the fields containing the raw data to present the results in their original format.

The queries used as source tables for the reports retrieve the information from the fields in SI units; the reports are designed to display the data in SI units as well. In order to provide the user with the ability to output the data in English units, twelve additional reports were created based on the same queries but with calculated controls to convert the data from SI to English units. The name of the existing report preceded by the word "English_" was given to each of the new reports (e.g., English_SPT Report).

The units and conversion factors used in the database are presented below:

	English	**To**	**SI**	**Multiply by**
Piles				
Pile Elastic Modulus	psi		kN/m^2	6.8948
Capwap Elastic Modulus	psi		kN/m^2	6.8948
Pile wave speed	ft/s		m/s	0.3048
Pile Impedance	lb-s/ft		kN-s/m	0.01459

	English	To	SI	Multiply by
Concrete Piles				
Diameter	in		cm	2.54
Void Diameter	in		cm	2.54
Cross Sectional Area	in^2		cm^2	6.4516
Total Length	ft		m	0.3048
Embedded Length	ft		m	0.3048
Total Weight	tons		kN	8.8964
Unit Weight of Concrete	pcf		kN/m^3	0.1571
Concrete Strength	psi		kN/m^2	6.8948
Steel H Piles				
Steel Cross Sectional Area	in^2		cm^2	6.4516
Pile Cross Sectional Area	in^2		cm^2	6.4516
Steel Section Perimeter	in		cm	2.54
Pile Perimeter	in		cm	2.54
Weight of Steel	tons/ft		kN/m	29.1878
Weight of pile	tons		kN	8.8964
Total Length	ft		m	0.3048
Embedded Length	ft		m	0.3048
Unit Weight of Steel	pcf		kN/m^3	0.1571
Concrete Strength	psi		kN/m^2	6.8948
Steel Pipe Piles				
Outside Diameter	in		cm	2.54
Thickness of Steel	in		cm	2.54
Steel Cross Sectional Area	in^2		cm^2	6.4516
Pile Cross Sectional Area	in^2		cm^2	6.4516
Weight of Steel	tons/ft		kN/m	29.1878
Weight of pile	tons		kN	8.8964
Total Length	ft		m	0.3048
Embedded Length	ft		m	0.3048
Unit Weight of Steel	pcf		kN/m^3	0.1571
Concrete Strength	psi		kN/m^2	6.8948
Driving Data				
Weight of Hammer	lb		kN	4.4482
Energy	lb-ft		kN-m	0.001355
Pre-bored Depth	ft		m	0.3048
Penetration (Last Blow)	in		cm	2.54
Penetration (End of Driving)	in		cm	2.54
Penetration (Start of Restriking)	in		cm	2.54
Penetration Log				
Depth	ft		m	0.3048

	English	To	SI	Multiply by
GRL Data				
L	ft		m	0.3048
Le	ft		m	0.3048
Area	ft^2		m^2	0.0929
C	ft/s		m/s	0.3048
Ec	$tons/ft^2$		GN/m^2	0.00009576
EA/c	tons-s/ft		kN-s/m	29.1878
Drilled Shafts				
Diameter	in		cm	2.54
Total Length	ft		m	0.3048
Embedded Length	ft		m	0.3048
Casing Length	ft		m	0.3048
Bell Diameter	in		cm	2.54
Bell Length	ft		m	0.3048
IGM				
Slump of Concrete	in		mm	25.4
f'_c	$tons/ft^2$		kN/m^2	95.7605
E'_c	$tons/ft^2$		kN/m^2	95.7605
Elev	ft		m	0.3048
Depth	ft		m	0.3048
q_U	$tons/ft^2$		kN/m^2	95.7605
q_T	$tons/ft^2$		kN/m^2	95.7605
E_M	$tons/ft^2$		kN/m^2	95.7605
Ei	$tons/ft^2$		kN/m^2	95.7605
Insitu tests				
Water Table Elevation	ft		m	0.3048
Ground Surface elevation	ft		m	0.3048
SPT				
Elevation	ft		m	0.3048
Depth	ft		m	0.3048
CPT				
Elevation	ft		m	0.3048
Depth	ft		m	0.3048
Qc	$tons/ft^2$		kN/m^2	95.7605
Fs	$tons/ft^2$		kN/m^2	95.7605
DMT				
Elevation	ft		m	0.3048
Depth	ft		m	0.3048

	English	To	SI	Multiply by
Static Load Test				
Force	tons		kN	8.8964
Displacement	in		mm	25.4
Statnamic Load Test				
Max Force	tons		kN	8.8964
Max Displacement	in		mm	25.4
Permanent Displacement	in		mm	25.4
Displacement	in		mm	25.4
Velocity	ft/s		m/s	0.3048
Acceleration	ft/s^2		m/s^2	0.3048

CHAPTER 3.
USING THE DATABASE

3.1. Opening the Database

Start Microsoft Access. If the Access startup dialog box appears, select "More Files…" on the Open an Existing Database section and click "OK"; otherwise, click on the "File" menu and choose "Open Database…". When the open database dialog box appears, find and select the database file Geotech.mdb and click "Open" to continue. Since the Deep Foundations Database is contained in a single file, it can be stored in any convenient location. The database can also be loaded directly by double clicking on its icon when using the Windows Explorer.

Once loaded, the database Menu form will appear as shown in Figure 3.1.

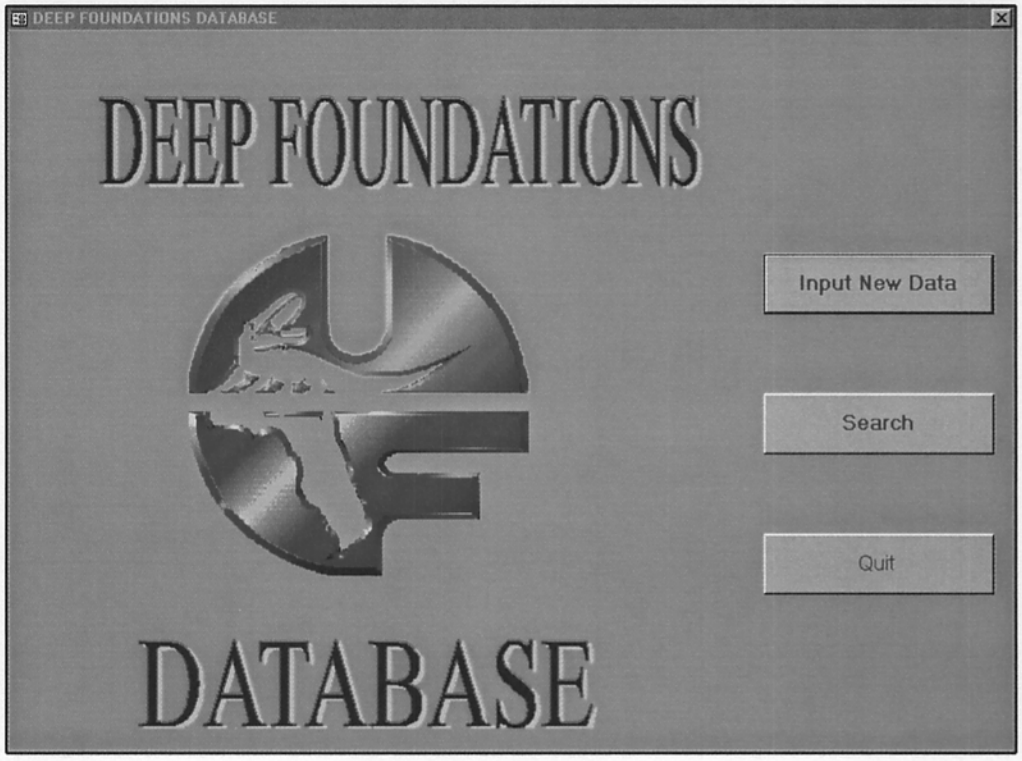


Figure 3.1 Menu form

The Menu form contains three buttons for the following options:

Input New Data — Input New Data. Opens the Add/Edit form and starts a new record.

Search — Search. Opens the Search form.

Quit — Quit. Closes the database and terminates Access.

3.2. Creating a New Record.

Click on the "Input New Data" button on the Menu form. The Add/Edit form will appear. After the user selects the element type from the "Type" control, the tabs will become visible. At this time, the form should look as shown in Figure 3.2 or Figure 3.3, depending on the selected element type.

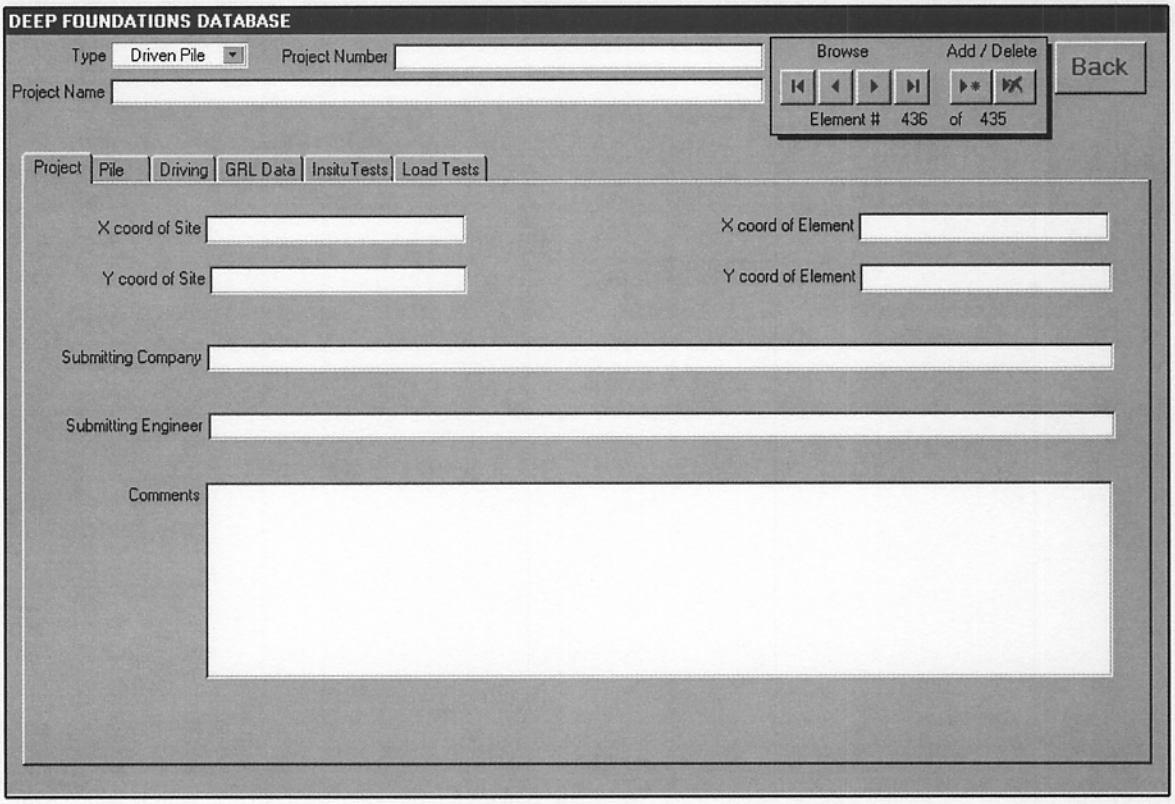


Figure 3.2 Add/Edit form (General, Driven Piles)

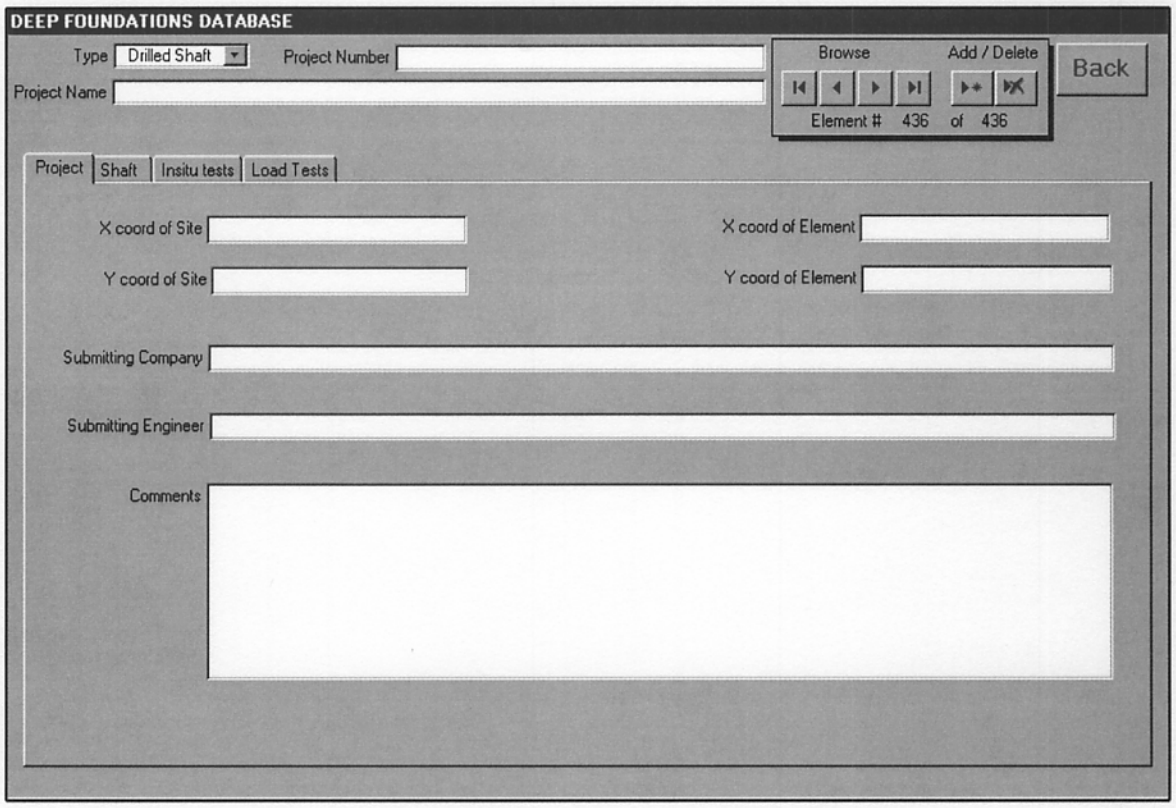


Figure 3.3 Add/Edit form (General, Drilled Shafts)

Data can be entered in any order by clicking on the different tabs and entering the available information. The data are automatically saved. If the value of a field is changed it is automatically updated.

When a section is accessed, all the control labels display the name of the fields without units; except for the "Project" tab, which only has dimensionless fields. Once the user selects the desired set of units, the labels will display the name of the field with the specified units. Units cannot be changed. If the user attempts to change the units the data in that tab section are erased and will have to be reentered. Similarly, if the user attempts to change the pile type, the insitu test type or the load test type in their respective sections, all the data in that section are erased and will have to be reentered.

Multiple insitu tests and load tests can be entered for a single element. A record navigation box is located at the upper right corner of the "Insitu Tests" and "Load Tests" sections as well as in the main form. It allows the user to add and delete records and to visit the available records. The functions of the controls in the navigation box are explained in Figure 3.4. Click "Back" to close the form and return to the menu form.

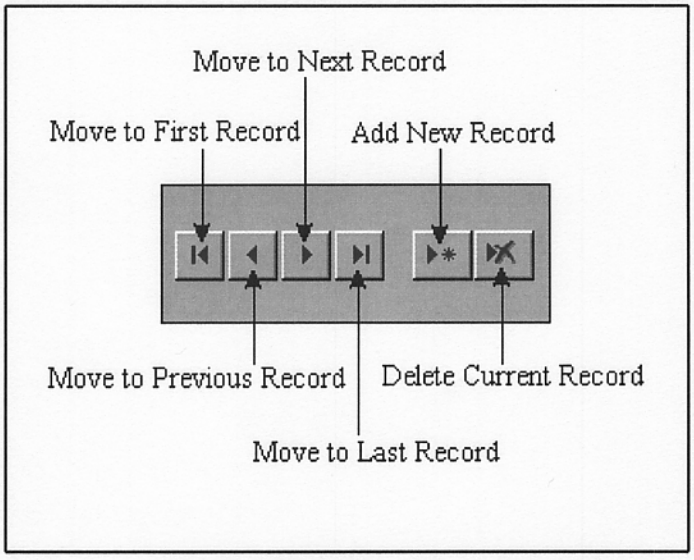


Figure 3.4 Record Navigation Box

3.3. Searching and Displaying Records.

Click on the "Search" button on the Menu form. The Search form will appear as shown in Figure 3.5. If the "Search" button is clicked without making any selections or typing any criteria the search will include all the records in the database. Select the desired criteria and click "Search" to narrow the search to only those records matching the specified criteria. The results from the search are shown by means of the Data Display form as shown in Figure 3.6. Navigation controls similar to those in the Add/Edit form are provided in this form (see Figure 3.4). The Add/Delete buttons are not available in this form because it is displayed in read only mode.

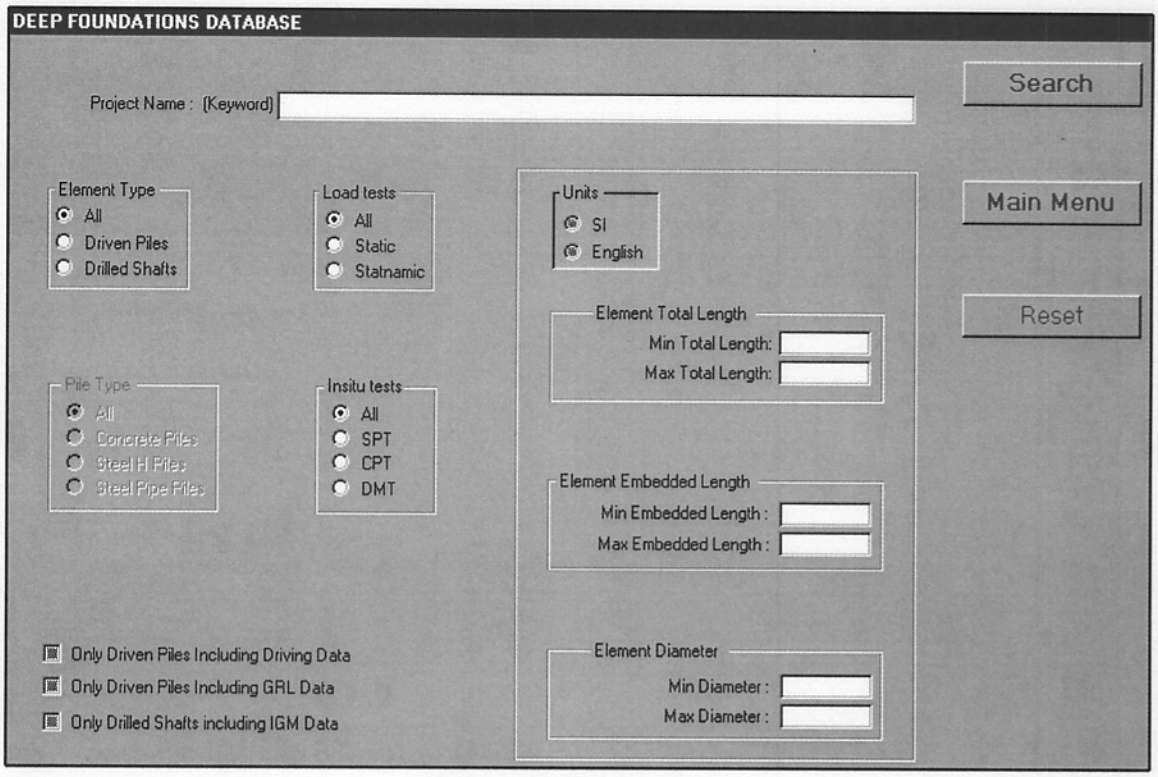


Figure 3.5 Search form

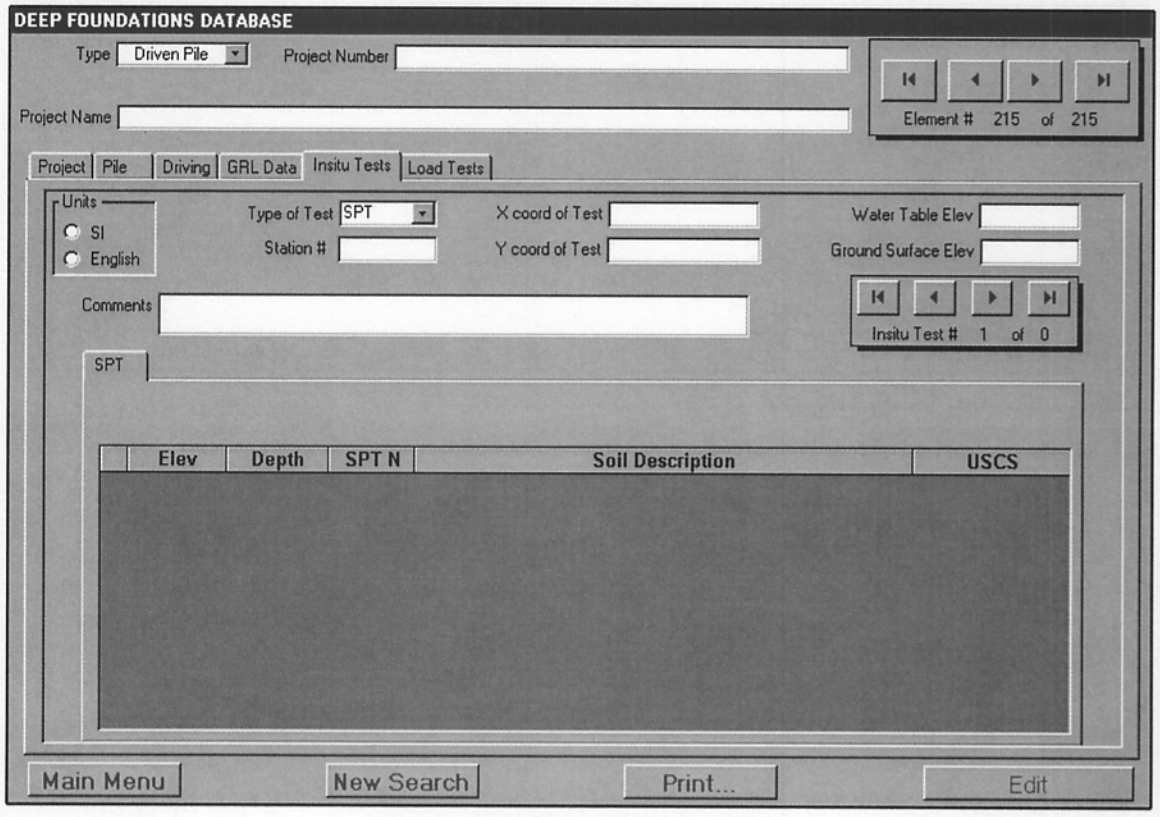


Figure 3.6 Data Display form

The following options are available by means of four command buttons located at the bottom of the form:

Closes the Display Data form and the Search form and returns to the Menu form.

Closes the Display Data form and returns to the Search form. The user can modify the previous search criteria to perform a new search. Alternatively, the user can click "Reset" to delete the specified criteria and then either click "Search" to display all the records or specify new criteria.

Opens the Print form to allow output of the data related to the current record.

Opens the Add/Edit form for the current record. The user is allowed to edit or erase the current record.

3.4. Editing Records.

When displaying the record to be edited in the Data Display form, click on the "Edit" command button. The Add/Edit form will appear as shown in Figure 3.7. This form only allows editing or deleting the current record. Therefore, the record navigation box is not visible; only the Delete button is available. Click "Back" to close the Add/Edit form and return to the Display Data form.

Data can also be edited by clicking "Add New Data" and locating the records to be edited using the record navigation box on the Add/Edit form (see Figures 3.2 and 3.3). This technique will allow the user to edit several records without leaving the Add/Edit form; however, it is much more difficult to find the records.

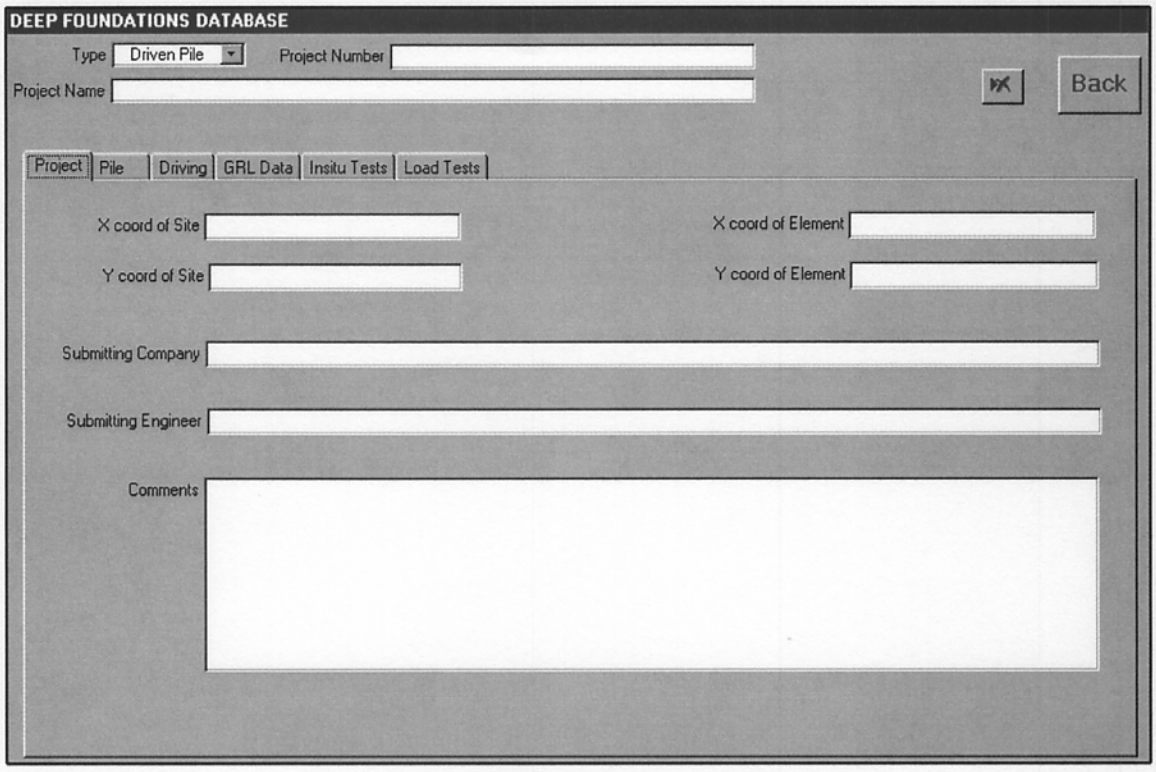


Figure 3.7 Add/Edit form (Current Record)

3.5. Outputting Records.

When displaying a specified record in the Data Display form, click on the "Print…" command button. The Print form will appear as shown in Figure 3.8. Reports can be printed in either SI or English units. In addition, each report can be printed, exported to a Microsoft Excel workbook or exported to a Microsoft Word document. Select the sections of the data to be output, the units and the output option and click "Print". The units, the output mode and at least one section have to be selected in order for the database to execute the "Print" command. Otherwise the command will not respond.

If either "Create Excel Workbook" or "Create Word Document" is selected, the user is prompted to assign a name and a location for each new file created. If the print

option is selected, the reports are automatically printed when the "Print" button is clicked.

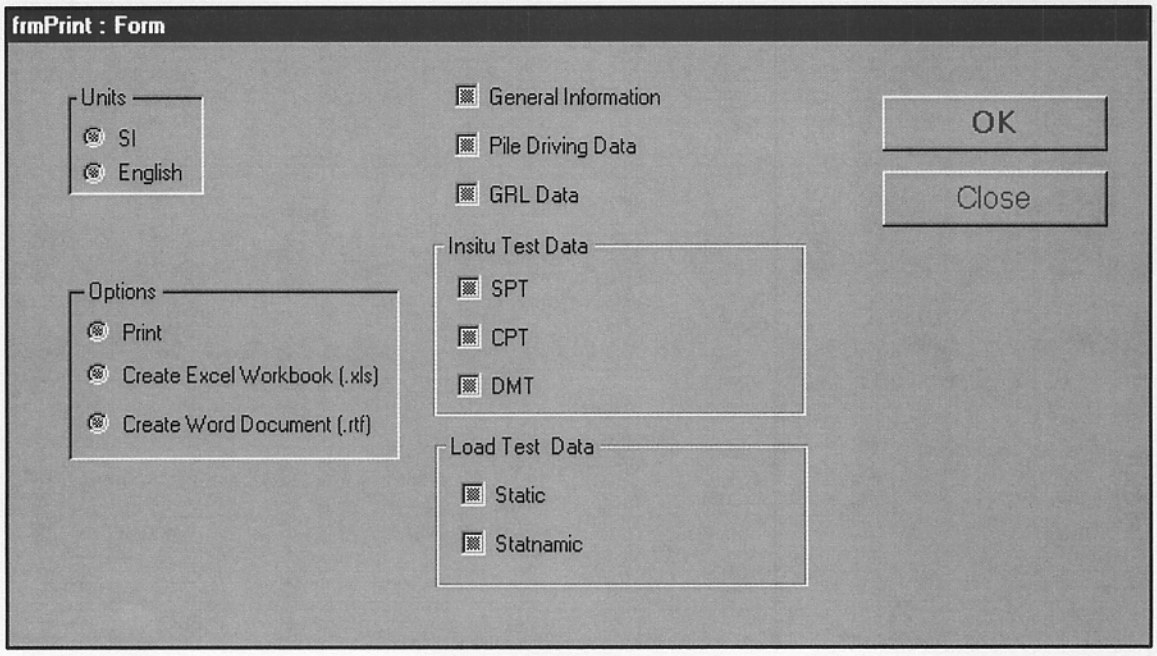


Figure 3.8 Print form

3.6. Optimizing Database Performance

After numerous additions and changes are made to the objects within a database file, especially additions and deletions of data in tables, the database file can become disorganized. When a record is deleted, the space in the file that the deleted data occupied is not automatically regained. The database must be compacted periodically to optimize its file size and the organization of data within the tables that the file contains. To compact the database the user should go to the Menu form. Then close it by clicking on the close control (☒), which is located in the upper right corner of the form. Choose "Tools, Database Utilities, Compact Database" from the Menu toolbar. Access immediately begins compacting the database. After a few seconds, the database compaction is completed and the Access automatically returns to the Menu form.

If a message that the database is corrupted is displayed or if the database behaves in an irregular manner, one or more of its objects may be corrupted as the result of a hardware error. For example, databases can become corrupt as the result of a power failure when the computer is writing on the database file. In such cases, close the Menu form and choose Tools, Database Utilities, Repair Database from the Menu toolbar to attempt to repair the damage. If Access cannot repair the corruption, the user would have to restore the latest database backup.

MODULES

PublicVars Declaration

```
Option Compare Database
Option Explicit

'Defines global variables. Used in more than one form
Public db As Database
Public ID As Integer
Public PileUnits As Integer
Public ShaftUnits As Integer
Public DrivingUnits As Integer
Public InsituUnits As Integer
Public LoadTestUnits As Integer
Public qdfSearch As QueryDef
Public SQL As String
```

CLASS MODULES
(Contained in the forms)

frmGeneral

```
Option Compare Database
Option Explicit
Dim CurrentType As String

Private Sub Form_Current()
  DoCmd.Maximize
  'Updates the value of the current record on the navigation box
  Me![CurrentRecordNumber].Value = Me.CurrentRecord
  'Calls the subprocedure ViewG
  ViewG
End Sub

Private Sub Type_Change()
  If CurrentType = "No Type" Then
    'Calls the subprocedure ViewG if the element type is selected for the first time
    ViewG
  Else
    'Displays a message warning the deletion of the current data
    'if the user attempts to change the element type
    If MsgBox("Changing the Type will erase the current Data", vbOKCancel) = vbOK Then
      'The record is deleted if the user clicks "OK"
      DoCmd.RunCommand acCmdDeleteRecord
      'Resets the CurrentType variable
      CurrentType = "No Type"
    Else
      'Displays the name of the current element type in the Type control
      'if the user clicks Cancel
      Me![Type] = CurrentType
    End If
  End If
End Sub



Private Sub ViewG()
  'Resets the visible property of the tab control
  Me!Box.Visible = False
  Me![Pile].Visible = True
  Me![Driving].Visible = True
  Me![Shaft].Visible = True
  Me![GRL Data].Visible = True
  'Updates the CurrentType variable
  CurrentType = "No Type"
  'Displays only the tabs related to the selected element type
  'and updates the CurrentTest variable
  Select Case Me![Type]
      Case "Drilled Shaft"
      Me!Box.Visible = True
      Me![Pile].Visible = False
      Me![Driving].Visible = False
      Me![GRL Data].Visible = False
      CurrentType = "Drilled Shaft"
      Case "Driven Pile"
      Me!Box.Visible = True
```

```
        Me![Shaft].Visible = False
        CurrentType = "Driven Pile"
    End Select
End Sub

'Moves to the previous record
Private Sub GoBack_Click()
On Error Resume Next
    DoCmd.GoToRecord , , acPrevious
End Sub

'Moves to the first record
Private Sub GoFirst_Click()
On Error Resume Next
    DoCmd.GoToRecord , , acFirst
End Sub

'Moves to the next record
Private Sub GoForward_Click()
On Error Resume Next
    DoCmd.GoToRecord , , acNext
End Sub

'Moves to the last record
Private Sub GoLast_Click()
On Error Resume Next
    DoCmd.GoToRecord , , acLast
End Sub

'Deletes the current record
Private Sub DeleteRecord_Click()
On Error Resume Next
    DoCmd.RunCommand acCmdDeleteRecord
    DoCmd.GoToRecord , , acPrevious
End Sub

'Starts a new record
Private Sub AddRecord_Click()
On Error Resume Next
    DoCmd.GoToRecord , , acNewRec
End Sub

'Closes the form
Private Sub BackButton_Click()
On Error Resume Next
    DoCmd.Close acForm, "frmGeneral"
End Sub
```

frmShafts

```
Option Compare Database
Option Explicit

Private Sub UnitsOff()
   'Displays a message if no units are selected prompting the user to select a set of units
   If IsNull(Me![Textbox_Units]) Then
      MsgBox "Please specify units"
      Me![Units].SetFocus
      'Erases the value assigned to the control
      DoCmd.RunCommand acCmdUndo
   End If
End Sub

Private Sub Form_Current()
   'Checks if a set of units is selected and updates the Units control
   Select Case Me![Textbox_Units]
   Case "SI"
   Me![Units].Value = 1
   Case "English"
   Me![Units].Value = 2
   Case Else
   Me![Units].Value = 0
   End Select
   'Updates the ShaftUnits variable
   ShaftUnits = Me![Units]
   'Calls the UnitsShaft subprocedure
   UnitsShaft
End Sub

Private Sub Units_AfterUpdate()
 'Updates the Textbox_Units control if units are selected for the first time
 If ShaftUnits = 0 Then
    Select Case Me![Units]
     Case 1
       Me![Textbox_Units].Value = "SI"
     Case 2
       Me![Textbox_Units].Value = "English"
    End Select
    'Updates the ShaftUnits variable
    ShaftUnits = Me![Units]
    'Calls the UnitsShaft subprocedure
    UnitsShaft
 Else
    'Displays a message warning the deletion of the current data
    'if the user attempts to change the units
    If MsgBox("Changing the Units will Erase the current Shaft Data", vbOKCancel) = vbOK Then
      'Deletes the record if the user clicks OK
      DoCmd.RunCommand acCmdDeleteRecord
      'Resets the ShaftUnits variable
      ShaftUnits = 0
    Else
      'Displays the current units if the user clicks Cancel
      Me![Units] = ShaftUnits
    End If
 End If
End Sub
```

```
Private Sub UnitsShaft()
'Changes all the labels to the specified set of units
Select Case ShaftUnits
  Case 1
    Me![Diameter Label].Caption = "Diameter (cm)"
    Me![Total Length Label].Caption = "Total Length (m)"
    Me![Embeded Length Label].Caption = "Embeded Length (m)"
    Me![Casing Length Label].Caption = "Casing Length (m)"
    Me![Bell Diameter Label].Caption = "Bell Diameter (cm)"
    Me![Bell Length Label].Caption = "Bell Length (m)"
    Me![Slump of Concrete Label].Caption = "Slump of Concrete (mm)"
    Me![Concrete strength Label].Caption = "Concrete strength f'c (KN/m²)"
    Me![Core Modulus Label].Caption = "Core Modulus Ec (KN/m²)"
    Me![Elev Label].Caption = "(m)"
    Me![Depth Label].Caption = "(m)"
    Me![qu Label].Caption = "(KN/m²)"
    Me![qt Label].Caption = "(KN/m²)"
    Me![Em Label].Caption = "(KN/m²)"
    Me![Ei Label].Caption = "(KN/m²)"
    Me![As built Design Load  Label].Caption = "Design Load (kN)"
    Me![As Built Diameter Label].Caption = "Diameter (cm)"
    Me![As Built Tip Elev Label].Caption = "Tip Elev (m)"
    Me![As Built Total Length Label].Caption = "Total Length (m)"
    Me![As Built Rock Socket Length Label].Caption = "Rock Socket Length (m)"
    Me![Top Elev Label].Caption = "Top Elev (m)"
    Me![Tip Elev Label].Caption = "Tip Elev (m)"
    Me![Scour Elev Label].Caption = "Scour Elev (m)"
    Me![Rock Socket Length Label].Caption = "Rock Socket Length (m)"
    Me![Design Load Label].Caption = "Design Load (kN)"
  Case 2
    Me![Diameter Label].Caption = "Diameter (in)"
    Me![Total Length Label].Caption = "Total Length (ft)"
    Me![Embeded Length Label].Caption = "Embeded Length (ft)"
    Me![Casing Length Label].Caption = "Casing Length (ft)"
    Me![Bell Diameter Label].Caption = "Bell Diameter (in)"
    Me![Bell Length Label].Caption = "Bell Length (ft)"
    Me![Slump of Concrete Label].Caption = "Slump of Concrete (in)"
    Me![Concrete strength Label].Caption = "Concrete strength f'c (Ton/ft²)"
    Me![Core Modulus Label].Caption = "Core Modulus Ec (Ton/ft²)"
    Me![Elev Label].Caption = "(ft)"
    Me![Depth Label].Caption = "(ft)"
    Me![qu Label].Caption = "(Ton/ft²)"
    Me![qt Label].Caption = "(Ton/ft²)"
    Me![Em Label].Caption = "(Ton/ft²)"
    Me![Ei Label].Caption = "(Ton/ft²)"
    Me![As built Design Load  Label].Caption = "Design Load (tons)"
    Me![As Built Diameter Label].Caption = "Diameter (in)"
    Me![As Built Tip Elev Label].Caption = "Tip Elev (ft)"
    Me![As Built Total Length Label].Caption = "Total Length (ft)"
    Me![As Built Rock Socket Length Label].Caption = "Rock Socket Length (ft)"
    Me![Top Elev Label].Caption = "Top Elev (ft)"
    Me![Tip Elev Label].Caption = "Tip Elev (ft)"
    Me![Scour Elev Label].Caption = "Scour Elev (ft)"
    Me![Rock Socket Length Label].Caption = "Rock Socket Length (ft)"
    Me![Design Load Label].Caption = "Design load (tons)"
  Case Else
    Me![Diameter Label].Caption = "Diameter"
    Me![Total Length Label].Caption = "Total Length"
    Me![Embeded Length Label].Caption = "Embeded Length"
    Me![Casing Length Label].Caption = "Casing Length"
    Me![Bell Diameter Label].Caption = "Bell Diameter"
    Me![Bell Length Label].Caption = "Bell Length"
```

```
        Me![Slump of Concrete Label].Caption = "Slump of Concrete"
        Me![Concrete strength Label].Caption = "Concrete strength f'c"
        Me![Core Modulus Label].Caption = "Core Modulus Ec"
        Me![Elev Label].Caption = ""
        Me![Depth Label].Caption = ""
        Me![qu Label].Caption = ""
        Me![qt Label].Caption = ""
        Me![Em Label].Caption = ""
        Me![Ei Label].Caption = ""
        Me![As built Design Load  Label].Caption = "Design Load"
        Me![As Built Diameter Label].Caption = "Diameter"
        Me![As Built Tip Elev Label].Caption = "Tip Elev"
        Me![As Built Total Length Label].Caption = "Total Length"
        Me![As Built Rock Socket Length Label].Caption = "Rock Socket Length"
        Me![Top Elev Label].Caption = "Top Elev"
        Me![Tip Elev Label].Caption = "Tip Elev"
        Me![Scour Elev Label].Caption = "Scour Elev"
        Me![Rock Socket Length Label].Caption = "Rock Socket Length"
        Me![Design Load Label].Caption = "Design Load"
    End Select
End Sub

'All of the following subprocedures have the same structure, they assign a value
'to the SI units control based on the value input in the Raw data field and on
'the selected set of units
Private Sub Diameter_AfterUpdate()
    'Calls the UnitsOff subprocedure
    UnitsOff
    'Calculates the value of the SI control
    Select Case ShaftUnits
    Case 1
        Me![Diameter (cm)] = Me![Diameter]
    Case 2
        Me![Diameter (cm)] = Me![Diameter] * 2.54
    End Select
    Forms![frmGeneral].Form![Diameter (cm)] = Me![Diameter (cm)]
End Sub

Private Sub Total_Length_AfterUpdate()
    UnitsOff
    Select Case ShaftUnits
    Case 1
        Me![Total Length (m)] = Me![Total Length]
    Case 2
        Me![Total Length (m)] = Me![Total Length] * 0.3048
    End Select
    Forms![frmGeneral].Form![Length (m)].Value = Me![Total Length (m)]
End Sub

Private Sub Embedded_Length_AfterUpdate()
    UnitsOff
    Select Case ShaftUnits
    Case 1
        Me![Embedded Length (m)] = Me![Embedded Length]
    Case 2
        Me![Embedded Length (m)] = Me![Embedded Length] * 0.3048
    End Select
        Forms![frmGeneral].Form![Emb Length (m)].Value = Me![Embedded Length (m)]
End Sub
```

```
Private Sub Casing_Length_AfterUpdate()
  UnitsOff
  Select Case ShaftUnits
  Case 1
    Me![Casing Length (m)] = Me![Casing Length]
  Case 2
    Me![Casing Length (m)] = Me![Casing Length] * 0.3048
  End Select
End Sub

Private Sub Bell_Diameter_AfterUpdate()
  UnitsOff
  Select Case ShaftUnits
  Case 1
    Me![Bell Diameter (cm)] = Me![Bell Diameter]
  Case 2
    Me![Bell Diameter (cm)] = Me![Bell Diameter] * 2.54
  End Select
End Sub

Private Sub Bell_Length_AfterUpdate()
  UnitsOff
  Select Case ShaftUnits
  Case 1
    Me![Bell Length (m)] = Me![Bell Length]
  Case 2
    Me![Bell Length (m)] = Me![Bell Length] * 0.3048
  End Select
End Sub

Private Sub Slump_of_Concrete_AfterUpdate()
  UnitsOff
  Select Case ShaftUnits
  Case 1
    Me![Slump of Concrete(mm)] = Me![Slump of Concrete]
  Case 2
    Me![Slump of Concrete(mm)] = Me![Slump of Concrete] * 25.4
  End Select
End Sub

Private Sub Concrete_strength__f_c_AfterUpdate()
  UnitsOff
  Select Case ShaftUnits
  Case 1
    Me![Concrete strength  f'c (KN/m²)] = Me![Concrete strength  f'c]
  Case 2
    Me![Concrete strength  f'c (KN/m²)] = Me![Concrete strength  f'c] * 95.7605
  End Select
End Sub

Private Sub Core_Modulus__Ec_AfterUpdate()
  UnitsOff
  Select Case ShaftUnits
  Case 1
    Me![Core Modulus  Ec (KN/m²)] = Me![Core Modulus  Ec]
  Case 2
    Me![Core Modulus  Ec (KN/m²)] = Me![Core Modulus  Ec] * 95.7605
  End Select
End Sub
```

```
Private Sub Top_Elev_AfterUpdate()
  UnitsOff
  Select Case ShaftUnits
  Case 1
    Me![Top Elev (m)] = Me![Top Elev]
  Case 2
    Me![Top Elev (m)] = Me![Top Elev] * 0.3048
  End Select
End Sub

Private Sub Tip_Elev_AfterUpdate()
  UnitsOff
  Select Case ShaftUnits
  Case 1
    Me![Tip Elev (m)] = Me![Tip Elev]
  Case 2
    Me![Tip Elev (m)] = Me![Tip Elev] * 0.3048
  End Select
End Sub

Private Sub Scour_Elev_AfterUpdate()
  UnitsOff
  Select Case ShaftUnits
  Case 1
    Me![Scour Elev (m)] = Me![Scour Elev]
  Case 2
    Me![Scour Elev (m)] = Me![Scour Elev] * 0.3048
  End Select
End Sub

Private Sub Rock_Socket_Length_AfterUpdate()
  UnitsOff
  Select Case ShaftUnits
  Case 1
    Me![Rock Socket Length (m)] = Me![Rock Socket Length]
  Case 2
    Me![Rock Socket Length (m)] = Me![Rock Socket Length] * 0.3048
  End Select
End Sub

Private Sub Design_Load_AfterUpdate()
  UnitsOff
  Select Case ShaftUnits
  Case 1
    Me![Design Load (kN)] = Me![Design Load]
  Case 2
    Me![Design Load (kN)] = Me![Design Load] * 8.8964
  End Select
End Sub

Private Sub As_Built_Diameter_AfterUpdate()
  UnitsOff
  Select Case ShaftUnits
  Case 1
    Me![As Built Diameter (cm)] = Me![As Built Diameter]
  Case 2
    Me![As Built Diameter (cm)] = Me![As Built Diameter] * 2.54
  End Select
End Sub
```

```
Private Sub As_Built_Tip_Elev_AfterUpdate()
  UnitsOff
  Select Case ShaftUnits
  Case 1
    Me![As Built Tip Elev (m)] = Me![As Built Tip Elev]
  Case 2
    Me![As Built Tip Elev (m)] = Me![As Built Tip Elev] * 0.3048
  End Select
End Sub

Private Sub As_Built_Total_Length_AfterUpdate()
  UnitsOff
  Select Case ShaftUnits
  Case 1
    Me![As Built Total length (m)] = Me![As Built Total length]
  Case 2
    Me![As Built Total length (m)] = Me![As Built Total length] * 0.3048
  End Select
End Sub

Private Sub As_Built_Rock_Socket_Length_AfterUpdate()
  UnitsOff
  Select Case ShaftUnits
  Case 1
    Me![As Built Rock Socket Length (m)] = Me![As Built Rock Socket Length]
  Case 2
    Me![As Built Rock Socket Length (m)] = Me![As Built Rock Socket Length] * 0.3048
  End Select
End Sub

Private Sub As_built_Design_Load_AfterUpdate()
  UnitsOff
  Select Case ShaftUnits
  Case 1
    Me![As built Design Load (kN)] = Me![As built Design Load]
  Case 2
    Me![As built Design Load (kN)] = Me![As built Design Load] * 8.8964
  End Select
End Sub
```

SbfIGM

```
Option Compare Database
Option Explicit

'All of the following subprocedures have the same structure. they assign a value
'to the SI units control based on the value input in the Raw data field and on
'the selected set of units
Public Sub Elev_AfterUpdate()
   'Calculates the value of the SI control
   Select Case ShaftUnits
   Case 1
      Me![Elev (m)] = Me![Elev]
   Case 2
      Me![Elev (m)] = Me![Elev] * 0.3048
   End Select
End Sub



Public Sub Depth_AfterUpdate()
   Select Case ShaftUnits
   Case 1
      Me![Depth (m)] = Me![Depth]
   Case 2
      Me![Depth (m)] = Me![Depth] * 0.3048
   End Select
End Sub

Public Sub qu_AfterUpdate()
   Select Case ShaftUnits
   Case 1
      Me![qu (KN/m²)] = Me![qu]
   Case 2
      Me![qu (KN/m²)] = Me![qu] * 95.7605
   End Select
End Sub

Private Sub qt_AfterUpdate()
   Select Case ShaftUnits
   Case 1
      Me![qt (KN/m²)] = Me![qt]
   Case 2
      Me![qt (KN/m²)] = Me![qt] * 95.7605
   End Select
End Sub

Private Sub Em_AfterUpdate()
   Select Case ShaftUnits
   Case 1
      Me![Em (KN/m²)] = Me![Em]
   Case 2
      Me![Em (KN/m²)] = Me![Em] * 95.7605
   End Select
End Sub
```

```
Private Sub Ei_AfterUpdate()
   Select Case ShaftUnits
   Case 1
      Me![Ei  (KN/m²)] = Me![Ei]
   Case 2
      Me![Ei  (KN/m²)] = Me![Ei] * 95.7605
   End Select
End Sub
```

frmPiles

```
Option Compare Database
Option Explicit
Dim CurrentPile As String

Private Sub Form_Current()
   'Calls the ViewP subprocedure
   ViewP
   'Checks if a set of units is selected and updates the Units control
   Select Case Me![Textbox_Units]
   Case "SI"
   Me![Units].Value = 1
   Case "English"
   Me![Units].Value = 2
   Case Else
   Me![Units].Value = 0
   End Select
   'Updates the PileUnits variable
   PileUnits = Me![Units]
   'Calls the UnitsPile subprocecure
   UnitsPile
End Sub

Private Sub UnitsOff()
   'Displays a message if no units are selected prompting the user to select a set of units
   If IsNull(Me![Textbox_Units]) Then
      MsgBox "Please specify units"
      Me![Units].SetFocus
      'Erases the value assigned to the control
      DoCmd.RunCommand acCmdUndo
   End If
End Sub

Private Sub Units_AfterUpdate()
  'Updates the Textbox_Units control if units are selected for the first time
  If PileUnits = 0 Then
     Select Case Me![Units]
       Case 1
         Me![Textbox_Units].Value = "SI"
       Case 2
         Me![Textbox_Units].Value = "English"
     End Select
     'Updates the PileUnits variable
     PileUnits = Me![Units]
     'Calls the UnitsPile subprocedure
     UnitsPile
  Else
     'Displays a message warning the deletion of the current data
     'if the user attempts to change the units
     If MsgBox("Changing the Units will Erase the current Pile Data", vbOKCancel) = vbOK Then
       'Deletes the record if the user clicks OK
       DoCmd.RunCommand acCmdDeleteRecord
       'Resets the PileUnits variable
       PileUnits = 0
     Else
       'Displays the current units if the user clicks Cancel
       Me![Units] = PileUnits
     End If
  End If
End Sub
```

```
Private Sub Pile_Type_Change()
   'Calls the UnitsOff subprocedure
   UnitsOff
   'Calls the ViewP subprocedure if the Pile Type is selected for the first time
   If CurrentPile = "No Pile" Then
   ViewP
   Else
    'Displays a message warning the deletion of the current data
    'if the user attempts to change the insitu test type
      If MsgBox("Changing the Pile type will Erase the current Pile Data", vbOKCancel) = vbOK Then
         'Deletes the record if the user clicks OK
         DoCmd.RunCommand acCmdDeleteRecord
         'Resets the CurrentPile variable
         CurrentPile = "No Pile"
      Else
         'Displays the current pile type if the user clicks Cancel
         Me![Pile Type] = CurrentPile
      End If
   End If
End Sub

Private Sub ViewP()
   'Resets the visible properties of the tab control
   Me![Box].Visible = True
   Me![H Pile].Visible = False
   Me![Pipe Pile].Visible = False
   Me![Concrete Pile].Visible = False
  'Displays only the tab for the selected pile type and Updates the CurrentPile variable
   Select Case Me![Pile Type]
        Case "Concrete Pile"
         Me![Concrete Pile].Visible = True
         CurrentPile = "Concrete Pile"
        Case "H Steel Pile"
         Me![H Pile].Visible = True
         CurrentPile = "H Steel Pile"
        Case "Pipe Steel Pile"
         Me![Pipe Pile].Visible = True
         CurrentPile = "Pipe Steel Pile"
        Case Else
         CurrentPile = "No Pile"
         Me![Box].Visible = False
  End Select
End Sub

Private Sub UnitsPile()
   'Changes all the labels to the specified set of units
   Select Case PileUnits
   Case 1
      Me![Pile Elastic Modulus Label].Caption = "Pile Elastic Modulus (KN/m²)"
      Me![Capwap Elastic Modulus Label].Caption = "Capwap Elastic Modulus KN/m²)"
      Me![Pile Wave Speed Label].Caption = "Pile Wave Speed c (m/s)"
      Me![Pile Impedance Label].Caption = "Pile Impedance E*A/c (KN-s/m)"
      Me![Concrete]![Void Diameter Label].Caption = "Void Diameter (cm)"
      Me![Concrete]![Width Label].Caption = "Width or Diameter (cm)"
      Me![Concrete]![Cross Sectional Area Label].Caption = "Cross Sectional Area (cm²)"
      Me![Concrete]![Total Length Label].Caption = "Total Length (m)"
      Me![Concrete]![Embedded Length Label].Caption = "Embedded Length (m)"
      Me![Concrete]![Total Weight Label].Caption = "Total Weight (KN)"
      Me![Concrete]![Unit Weight of Concrete Label].Caption = "Unit Weight of Concrete (KN/m^3)"
      Me![Concrete]![Concrete strength Label].Caption = "Concrete Strength (KN/m²)"
      Me![H]![Steel Cross Sectional Area Label].Caption = "Steel Cross Sectional Area (cm²)"
```

```
    Me![H]![Pile Cross Sectional Area Label].Caption = "Pile Cross Sectional Area (cm²)"
    Me![H]![Steel Section Perimeter Label].Caption = "Steel Section Perimeter (cm)"
    Me![H]![Pile  Perimeter Label].Caption = "Pile  Perimeter (cm)"
    Me![H]![Weight of Steel  Label].Caption = "Weight of Steel (KN/m)"
    Me![H]![Weight of pile Label].Caption = "Weight of pile (KN)"
    Me![H]![Total Length Label].Caption = "Total Length (m)"
    Me![H]![Embedded Length Label].Caption = "Embedded Length (m)"
    Me![H]![Unit Weight of Steel Label].Caption = "Unit Weight of Steel (KN/m^3)"
    Me![H]![Concrete strength Label].Caption = "Concrete Strength (KN/m²)"
    Me![Pipe]![Outside Diameter Label].Caption = "Outside Diameter (cm)"
    Me![Pipe]![Thickness of Steel Label].Caption = "Thickness of Steel (cm)"
    Me![Pipe]![Steel Cross Sectional Area Label].Caption = "Steel Cross Sectional Area (cm²)"
    Me![Pipe]![Pile Cross Sectional Area Label].Caption = "Pile Cross Sectional Area (cm²)"
    Me![Pipe]![Concrete strength Label].Caption = "Concrete Strength (KN/m²)"
    Me![Pipe]![Weight of Steel Label].Caption = "Weight of Steel (KN/m)"
    Me![Pipe]![Weight of pile Label].Caption = "Weight of  Pile (KN)"
    Me![Pipe]![Total Length Label].Caption = "Total Length (m)"
    Me![Pipe]![Embedded Length Label].Caption = "Embedded Length (m)"
    Me![Pipe]![Unit Weight of Steel Label].Caption = "Unit Weight of Steel (KN/m^3)"
Case 2
    Me![Pile Elastic Modulus Label].Caption = "Pile Elastic Modulus (psi)"
    Me![Capwap Elastic Modulus Label].Caption = "Capwap Elastic Modulus (psi)"
    Me![Pile Wave Speed Label].Caption = "Pile Wave Speed c (ft/s)"
    Me![Pile Impedance Label].Caption = "Pile Impedance E*A/c (lb-s/ft)"
    Me![Concrete]![Void Diameter Label].Caption = "Void Diameter (in)"
    Me![Concrete]![Width Label].Caption = "Width or Diameter (in)"
    Me![Concrete]![Cross Sectional Area Label].Caption = "Cross Sectional Area (in²)"
    Me![Concrete]![Total Length Label].Caption = "Total Length (ft)"
    Me![Concrete]![Embedded Length Label].Caption = "Embedded Length (ft)"
    Me![Concrete]![Total Weight Label].Caption = "Total Weight (tons)"
    Me![Concrete]![Unit Weight of Concrete Label].Caption = "Unit Weight of Concrete (pcf)"
    Me![Concrete]![Concrete strength Label].Caption = "Concrete Strength (psi)"
    Me![H]![Steel Cross Sectional Area Label].Caption = "Steel Cross Sectional Area (in²)"
    Me![H]![Pile Cross Sectional Area Label].Caption = "Pile Cross Sectional Area (in²)"
    Me![H]![Steel Section Perimeter Label].Caption = "Steel Section Perimeter (in)"
    Me![H]![Pile  Perimeter Label].Caption = "Pile  Perimeter (in)"
    Me![H]![Weight of Steel  Label].Caption = "Weight of Steel (tons/ft)"
    Me![H]![Weight of pile Label].Caption = "Weight of pile (tons)"
    Me![H]![Total Length Label].Caption = "Total Length (ft)"
    Me![H]![Embedded Length Label].Caption = "Embedded Length (ft)"
    Me![H]![Unit Weight of Steel Label].Caption = "Unit Weight of Steel (pcf)"
    Me![H]![Concrete strength Label].Caption = "Concrete Strength (psi)"
    Me![Pipe]![Outside Diameter Label].Caption = "Outside Diameter (in)"
    Me![Pipe]![Thickness of Steel Label].Caption = "Thickness of Steel (in)"
    Me![Pipe]![Steel Cross Sectional Area Label].Caption = "Steel Cross Sectional Area (in²)"
    Me![Pipe]![Pile Cross Sectional Area Label].Caption = "Pile Cross Sectional Area (in²)"
    Me![Pipe]![Concrete strength Label].Caption = "Concrete Strength (psi)"
    Me![Pipe]![Weight of Steel Label].Caption = "Weight of Steel (tons/ft)"
    Me![Pipe]![Weight of pile Label].Caption = "Weight of  Pile (tons)"
    Me![Pipe]![Total Length Label].Caption = "Total Length (ft)"
    Me![Pipe]![Embedded Length Label].Caption = "Embedded Length (ft)"
    Me![Pipe]![Unit Weight of Steel Label].Caption = "Unit Weight of Steel (pcf)"
Case Else
    Me![Pile Elastic Modulus Label].Caption = "Pile Elastic Modulus"
    Me![Capwap Elastic Modulus Label].Caption = "Capwap Elastic Modulus"
    Me![Pile Wave Speed Label].Caption = "Pile Wave Speed ( c )"
    Me![Pile Impedance Label].Caption = "Pile Impedance ( E*A/c )"
    Me![Capwap Elastic Modulus Label].Caption = "Capwap Elastic Modulus"
    Me![Concrete]![Void Diameter Label].Caption = "Void Diameter"
    Me![Concrete]![Width Label].Caption = "Width or Diameter"
    Me![Concrete]![Cross Sectional Area Label].Caption = "Cross Sectional Area"
    Me![Concrete]![Total Length Label].Caption = "Total Length"
```

```
        Me![Concrete]![Embedded Length Label].Caption = "Embedded Length"
        Me![Concrete]![Total Weight Label].Caption = "Total Weight"
        Me![Concrete]![Unit Weight of Concrete Label].Caption = "Unit Weight of Concrete"
        Me![Concrete]![Concrete strength Label].Caption = "Concrete Strength"
        Me![H]![Steel Cross Sectional Area Label].Caption = "Steel Cross Sectional Area"
        Me![H]![Pile Cross Sectional Area Label].Caption = "Pile Cross Sectional Area"
        Me![H]![Steel Section Perimeter Label].Caption = "Steel Section Perimeter"
        Me![H]![Pile  Perimeter Label].Caption = "Pile  Perimeter"
        Me![H]![Weight of Steel  Label].Caption = "Weight of Steel"
        Me![H]![Weight of pile Label].Caption = "Weight of pile"
        Me![H]![Total Length Label].Caption = "Total Length"
        Me![H]![Embedded Length Label].Caption = "Embedded Length"
        Me![H]![Unit Weight of Steel Label].Caption = "Unit Weight of Steel"
        Me![H]![Concrete strength Label].Caption = "Concrete Strength"
        Me![Pipe]![Outside Diameter Label].Caption = "Outside Diameter"
        Me![Pipe]![Thickness of Steel Label].Caption = "Thickness of Steel"
        Me![Pipe]![Steel Cross Sectional Area Label].Caption = "Steel Cross Sectional Area"
        Me![Pipe]![Pile Cross Sectional Area Label].Caption = "Pile Cross Sectional Area"
        Me![Pipe]![Concrete strength Label].Caption = "Concrete Strength"
        Me![Pipe]![Weight of Steel Label].Caption = "Weight of Steel"
        Me![Pipe]![Weight of pile Label].Caption = "Weight of  Pile"
        Me![Pipe]![Total Length Label].Caption = "Total Length"
        Me![Pipe]![Embedded Length Label].Caption = "Embedded Length"
        Me![Pipe]![Unit Weight of Steel Label].Caption = "Unit Weight of Steel"
    End Select
End Sub

'All of the following subprocedures have the same structure, they assign a value
'to the SI units control based on the value input in the Raw data field and on
'the selected set of units
Private Sub Pile_Elastic_Modulus_AfterUpdate()
    'Calls the UnitsOff subprocedure
    UnitsOff
    'Calculates the value of the SI control
    Select Case PileUnits
    Case 1
        Me![Pile Elastic Modulus (KN/m²)] = Me![Pile Elastic Modulus]
    Case 2
        Me![Pile Elastic Modulus (KN/m²)] = Me![Pile Elastic Modulus] * 6.8948
    End Select
End Sub

Private Sub Capwap_Elastic_Modulus_AfterUpdate()
    UnitsOff
    Select Case PileUnits
    Case 1
        Me![Capwap Elastic Modulus (KN/m²)] = Me![Capwap Elastic Modulus]
    Case 2
        Me![Capwap Elastic Modulus (KN/m²)] = Me![Capwap Elastic Modulus] * 6.8948
    End Select
End Sub

Private Sub Pile_Wave_Speed_AfterUpdate()
    UnitsOff
    Select Case PileUnits
    Case 1
        Me![Pile Wave Speed c (m/s)] = Me![Pile Wave Speed]
    Case 2
        Me![Pile Wave Speed c (m/s)] = Me![Pile Wave Speed] * 0.3048
    End Select
End Sub
```

```
Private Sub Pile_Impedance_AfterUpdate()
  UnitsOff
  Select Case PileUnits
  Case 1
    Me![Pile Impedance E*A/c (KN-s/m)] = Me![Pile Impedance]
  Case 2
    Me![Pile Impedance E*A/c (KN-s/m)] = Me![Pile Impedance] * 0.01459
  End Select
End Sub
```

SbfPiles Concrete

```
Option Compare Database
Option Explicit

'All of the following subprocedures have the same structure, they assign a value
'to the SI units control based on the value input in the Raw data field and on
'the selected set of units
Private Sub Void_Diameter_AfterUpdate()
   'Calculates the value of the SI control
   Select Case PileUnits
   Case 1
      Me![Void Diameter (cm)] = Me![Void Diameter]
   Case 2
      Me![Void Diameter (cm)] = Me![Void Diameter] * 2.54
   End Select
End Sub

Public Sub Width_AfterUpdate()
   Select Case PileUnits
   Case 1
      Me![Width (cm)] = Me![Width]
   Case 2
      Me![Width (cm)] = Me![Width] * 2.54
   End Select
   Forms![frmGeneral].Form![Diameter (cm)] = Me![Width (cm)]
End Sub

Public Sub Cross_Sectional_Area_AfterUpdate()
   Select Case PileUnits
   Case 1
      Me![Cross Sectional Area (cm²)] = Me![Cross Sectional Area]
   Case 2
      Me![Cross Sectional Area (cm²)] = Me![Cross Sectional Area] * 6.4516
   End Select
End Sub

Public Sub Total_Length_AfterUpdate()
   Select Case PileUnits
   Case 1
      Me![Total Length (m)] = Me![Total Length]
   Case 2
      Me![Total Length (m)] = Me![Total Length] * 0.3048
   End Select
   Forms![frmGeneral].Form![Length (m)].Value = Me![Total Length (m)]
End Sub

Public Sub Embedded_Length_AfterUpdate()
   Select Case PileUnits
   Case 1
      Me![Embedded Length (m)] = Me![Embedded Length]
   Case 2
      Me![Embedded Length (m)] = Me![Embedded Length] * 0.3048
   End Select
      Forms![frmGeneral].Form![Emb Length (m)].Value = Me![Embedded Length (m)]
End Sub

Public Sub Total_Weight_AfterUpdate()
   Select Case PileUnits
   Case 1
      Me![Total Weight (KN)] = Me![Total Weight]
   Case 2
```

```
    Me![Total Weight (KN)] = Me![Total Weight] * 8.8964
  End Select
End Sub

Public Sub Unit_Weight_of_Concrete_AfterUpdate()
  Select Case PileUnits
  Case 1
    Me![Unit Weight of Concrete (KN/m^3)] = Me![Unit Weight of Concrete]
  Case 2
    Me![Unit Weight of Concrete (KN/m^3)] = Me![Unit Weight of Concrete] * 0.1571
  End Select
End Sub

Public Sub Concrete_Strength_AfterUpdate()
  Select Case PileUnits
  Case 1
    Me![Concrete Strength (KN/m²)] = Me![Concrete Strength]
  Case 2
    Me![Concrete Strength (KN/m²)] = Me![Concrete Strength] * 6.8948
  End Select
End Sub
```

SbfPiles H

```
Option Compare Database
Option Explicit

'All of the following subprocedures have the same structure, they assign a value
'to the SI units control based on the value input in the Raw data field and on
'the selected set of units
Private Sub Steel_Cross_Sectional_Area_AfterUpdate()
   'Calculates value of the SI control
   Select Case PileUnits
   Case 1
      Me![Steel Cross Sectional Area (cm²)] = Me![Steel Cross Sectional Area]
   Case 2
      Me![Steel Cross Sectional Area (cm²)] = Me![Steel Cross Sectional Area] * 6.4516
   End Select
End Sub

Private Sub Pile_Cross_Sectional_Area_AfterUpdate()
   Select Case PileUnits
   Case 1
      Me![Pile Cross Sectional Area (cm²)] = Me![Pile Cross Sectional Area]
   Case 2
      Me![Pile Cross Sectional Area (cm²)] = Me![Pile Cross Sectional Area] * 6.4516
   End Select
End Sub

Private Sub Steel_Section_Perimeter_AfterUpdate()
   Select Case PileUnits
   Case 1
      Me![Steel Section Perimeter (cm)] = Me![Steel Section Perimeter]
   Case 2
      Me![Steel Section Perimeter (cm)] = Me![Steel Section Perimeter] * 2.54
   End Select
End Sub

Private Sub Pile__Perimeter_AfterUpdate()
   Select Case PileUnits
   Case 1
      Me![Pile  Perimeter (cm)] = Me![Pile  Perimeter]
   Case 2
      Me![Pile  Perimeter (cm)] = Me![Pile  Perimeter] * 2.54
End Select
End Sub

Private Sub Weight_of_Steel_AfterUpdate()
   Select Case PileUnits
   Case 1
      Me![Weight of Steel (KN/m)] = Me![Weight of Steel]
   Case 2
      Me![Weight of Steel (KN/m)] = Me![Weight of Steel] * 29.1878
   End Select
End Sub

Private Sub Weight_of_pile_AfterUpdate()
   Select Case PileUnits
   Case 1
      Me![Weight of pile (KN)] = Me![Weight of pile]
   Case 2
      Me![Weight of pile (KN)] = Me![Weight of pile] * 8.8964
   End Select
End Sub
```

```
Private Sub Total_Length_AfterUpdate()
  Select Case PileUnits
  Case 1
    Me![Total Length (m)] = Me![Total Length]
  Case 2
    Me![Total Length (m)] = Me![Total Length] * 0.3048
  End Select
  Forms![frmGeneral].Form![Length (m)].Value = Me![Total Length (m)]
End Sub

Private Sub Embedded_Length_AfterUpdate()
  Select Case PileUnits
  Case 1
    Me![Embedded Length (m)] = Me![Embedded Length]
  Case 2
    Me![Embedded Length (m)] = Me![Embedded Length] * 0.3048
  End Select
    Forms![frmGeneral].Form![Emb Length (m)].Value = Me![Embedded Length (m)]
End Sub

Private Sub Unit_Weight_of_Steel_AfterUpdate()
  Select Case PileUnits
  Case 1
    Me![Unit Weight of Steel (KN/m^3)] = Me![Unit Weight of Steel]
  Case 2
    Me![Unit Weight of Steel (KN/m^3)] = Me![Unit Weight of Steel] * 0.1571
  End Select
End Sub

Private Sub Concrete_Strength_AfterUpdate()
  Select Case PileUnits
  Case 1
    Me![Concrete Strength (KN/m²)] = Me![Concrete Strength]
  Case 2
    Me![Concrete Strength (KN/m²)] = Me![Concrete Strength] * 6.8948
  End Select
End Sub
```

SbfPiles Pipe

```
Option Compare Database
Option Explicit

'All of the following subprocedures have the same structure, they assign a value
'to the SI units control based on the value input in the Raw data field and on
'the selected set of units
Public Sub Outside_Diameter_AfterUpdate()
   'Calculates value of SI control
   Select Case PileUnits
   Case 1
      Me![Outside Diameter (cm)] = Me![Outside Diameter]
   Case 2
      Me![Outside Diameter (cm)] = Me![Outside Diameter] * 2.54
   End Select
   Forms![frmGeneral].Form![Diameter (cm)] = Me![Outside Diameter (cm)]
End Sub

Public Sub Thickness_of_Steel_AfterUpdate()
   Select Case PileUnits
   Case 1
      Me![Thickness of Steel (cm)] = Me![Thickness of Steel]
   Case 2
      Me![Thickness of Steel (cm)] = Me![Thickness of Steel] * 2.54
   End Select
End Sub

Public Sub Steel_Cross_Sectional_Area_AfterUpdate()
   Select Case PileUnits
   Case 1
      Me![Steel Cross Sectional Area (cm²)] = Me![Steel Cross Sectional Area]
   Case 2
      Me![Steel Cross Sectional Area (cm²)] = Me![Steel Cross Sectional Area] * 6.4516
   End Select
End Sub

Public Sub Pile_Cross_Sectional_Area_AfterUpdate()
   Select Case PileUnits
   Case 1
      Me![Pile Cross Sectional Area (cm²)] = Me![Pile Cross Sectional Area]
   Case 2
      Me![Pile Cross Sectional Area (cm²)] = Me![Pile Cross Sectional Area] * 6.4516
   End Select
End Sub

Public Sub Concrete_Strength_AfterUpdate()
   Select Case PileUnits
   Case 1
      Me![Concrete Strength (KN/m²)] = Me![Concrete Strength]
   Case 2
      Me![Concrete Strength (KN/m²)] = Me![Concrete Strength] * 6.8948
   End Select
End Sub

Public Sub Weight_of_Steel_AfterUpdate()
   Select Case PileUnits
   Case 1
      Me![Weight of Steel (KN/m)] = Me![Weight of Steel]
   Case 2
      Me![Weight of Steel (KN/m)] = Me![Weight of Steel] * 29.1878
   End Select
```

```
End Sub

Public Sub Weight_of_pile_AfterUpdate()
  Select Case PileUnits
  Case 1
    Me![Weight of pile (KN)] = Me![Weight of pile]
  Case 2
    Me![Weight of pile (KN)] = Me![Weight of pile] * 8.8964
  End Select
End Sub

Public Sub Total_Length_AfterUpdate()
  Select Case PileUnits
  Case 1
    Me![Total Length (m)] = Me![Total Length]
  Case 2
    Me![Total Length (m)] = Me![Total Length] * 0.3048
  End Select
  Forms![frmGeneral].Form![Length (m)].Value = Me![Total Length (m)]
End Sub

Public Sub Embedded_Length_AfterUpdate()
  Select Case PileUnits
  Case 1
    Me![Embedded Length (m)] = Me![Embedded Length]
  Case 2
    Me![Embedded Length (m)] = Me![Embedded Length] * 0.3048
  End Select
  Forms![frmGeneral].Form![Emb Length (m)].Value = Me![Embedded Length (m)]
End Sub

Public Sub Unit_Weight_of_Steel_AfterUpdate()
  Select Case PileUnits
  Case 1
    Me![Unit Weight of Steel (KN/m^3)] = Me![Unit Weight of Steel]
  Case 2
    Me![Unit Weight of Steel (KN/m^3)] = Me![Unit Weight of Steel] * 0.1571
  End Select
End Sub
```

frmPiles Driving

```
Option Compare Database
Option Explicit

Private Sub UnitsOff()
  'Displays a message if no units are selected prompting the user to select a set of units
  If IsNull(Me![Textbox_Units]) Then
    MsgBox "Please specify units"
    Me![Units].SetFocus
    'Erases the value assigned to the control
    DoCmd.RunCommand acCmdUndo
  End If
End Sub

Private Sub Form_Current()
  'Checks if a set of units is selected and updates the Units control
  Select Case Me![Textbox_Units]
  Case "SI"
  Me![Units].Value = 1
  Case "English"
  Me![Units].Value = 2
  Case Else
  Me![Units].Value = 0
  End Select
  'Updates the DrivingUnits variable
  DrivingUnits = Me![Units]
  'Calls the UnitsDriving subprocecure
  UnitsDriving
End Sub

Private Sub Units_AfterUpdate()
 'Updates the Textbox_Units control if units are selected for the first time
 If DrivingUnits = 0 Then
    Select Case Me![Units]
     Case 1
       Me![Textbox_Units].Value = "SI"
     Case 2
       Me![Textbox_Units].Value = "English"
    End Select
    'Updates the DrivingUnits variable
    DrivingUnits = Me![Units]
    'Calls the UnitsDriving subprocedure
    UnitsDriving
 Else
    'Displays a message warning the deletion of the current data
    'if the user attempts to change the units
    If MsgBox("Changing the Units will Erase the current Driving Data", vbOKCancel) = vbOK Then
      'Deletes the record if the user clicks OK
      DoCmd.RunCommand acCmdDeleteRecord
      'Resets the PileUnits variable
      DrivingUnits = 0
    Else
      'Displays the current units if the user clicks Cancel
      Me![Units] = DrivingUnits
    End If
 End If
End Sub

Private Sub UnitsDriving()
'Changes all the labels to the specified set of units
Select Case DrivingUnits
```

```
    Case 1
        Me![Weigh Label].Caption = "Weigh (N)"
        Me![Energy Label].Caption = "Energy (KN-m)"
        Me![Pre Bored Depth Label].Caption = "Pre Bored Depth (m)"
        Me![Last Blow Label].Caption = "Last Blow (cm)"
        Me![End of Driving Label].Caption = "End of Driving (cm)"
        Me![Start of Restrike Label].Caption = "Start of Restrike (cm)"
        Me![Penet Depth Label].Caption = "(m)"
        Me![Penetration Label].Caption = "(Blows/30.48cm)"
    Case 2
        Me![Weigh Label].Caption = "Weigh (lb)"
        Me![Energy Label].Caption = "Energy (lb-ft)"
        Me![Pre Bored Depth Label].Caption = "Pre Bored Depth (ft)"
        Me![Last Blow Label].Caption = "Last Blow (in)"
        Me![End of Driving Label].Caption = "End of Driving (in)"
        Me![Start of Restrike Label].Caption = "Start of Restrike (in)"
        Me![Penet Depth Label].Caption = "(ft)"
        Me![Penetration Label].Caption = "Blows/ft"
    Case Else
        Me![Weigh Label].Caption = "Weigh"
        Me![Energy Label].Caption = "Energy"
        Me![Pre Bored Depth Label].Caption = "Pre Bored Depth"
        Me![Last Blow Label].Caption = "Last Blow"
        Me![End of Driving Label].Caption = "End of Driving"
        Me![Start of Restrike Label].Caption = "Start of Restrike"
        Me![Penet Depth Label].Caption = ""
        Me![Penetration Label].Caption = ""
    End Select
End Sub

'All of the following subprocedures have the same structure, they assign a value
'to the SI units control based on the value input in the Raw data field and on
'the selected set of units
Private Sub Weight_AfterUpdate()
    'Calls the UnitsOff subprocedure
    UnitsOff
    'Calculates the value of the SI control
    Select Case DrivingUnits
    Case 1
        Me![Weight (N)] = Me![Weight]
    Case 2
        Me![Weight (N)] = Me![Weight] * 4.4482
    End Select
End Sub

Private Sub Energy_AfterUpdate()
    UnitsOff
    Select Case DrivingUnits
    Case 1
        Me![Energy (KN-m)] = Me![Energy]
    Case 2
        Me![Energy (KN-m)] = Me![Energy] * 0.0013558
    End Select
End Sub

Private Sub Pre_Bored_Depth_AfterUpdate()
    UnitsOff
    Select Case DrivingUnits
    Case 1
        Me![Pre Bored Depth (m)] = Me![Pre Bored Depth]
    Case 2
        Me![Pre Bored Depth (m)] = Me![Pre Bored Depth] * 0.3048
```

```
  End Select
End Sub

Private Sub Last_Blow_AfterUpdate()
  UnitsOff
  Select Case DrivingUnits
  Case 1
    Me![Last Blow (cm)] = Me![Last Blow]
  Case 2
    Me![Last Blow (cm)] = Me![Last Blow] * 2.54
  End Select
End Sub

Private Sub End_of_Driving_AfterUpdate()
  UnitsOff
  Select Case DrivingUnits
  Case 1
    Me![End of Driving (cm)] = Me![End of Driving]
  Case 2
    Me![End of Driving (cm)] = Me![End of Driving] * 2.54
  End Select
End Sub

Private Sub Start_of_Restrike_AfterUpdate()
  UnitsOff
  Select Case DrivingUnits
  Case 1
    Me![Start of Restrike (cm)] = Me![Start of Restrike]
  Case 2
    Me![Start of Restrike (cm)] = Me![Start of Restrike] * 2.54
  End Select
End Sub
```

sbfDriving Penetration

```
Option Compare Database
Option Explicit

'assign a value to the SI units control based on the value input in the Raw data
'field and on the selected set of units
Private Sub Depth_AfterUpdate()
   'Calculates the value of the SI control
   Select Case DrivingUnits
   Case 1
      Me![Depth (m)] = Me![Depth]
   Case 2
      Me![Depth (m)] = Me![Depth] * 0.3048
   End Select
End Sub
```

frmGRL

```
Option Compare Database
Option Explicit
Dim GRLUnits As Integer

Private Sub Form_Current()
   'Checks if a set of units is selected and updates the Units control
   Select Case Me![Textbox_Units]
   Case "SI"
   Me![Units].Value = 1
   Case "English"
   Me![Units].Value = 2
   Case Else
   Me![Units].Value = 0
   End Select
   'Updates the GRLUnits variable
   GRLUnits = Me![Units]
   'Calls the UnitsGRL subprocecure
   UnitsGRL
End Sub

Private Sub UnitsOff()
   'Displays a message if no units are selected prompting the user to select a set of units
   If IsNull(Me![Textbox_Units]) Then
      MsgBox "Please specify units"
      Me![Units].SetFocus
      'Erases the value assigned to the control
      DoCmd.RunCommand acCmdUndo
   End If
End Sub

Private Sub Units_AfterUpdate()
 'Updates the Textbox_Units control if units are selected for the first time
 If GRLUnits = 0 Then
    Select Case Me![Units]
     Case 1
       Me![Textbox_Units].Value = "SI"
     Case 2
       Me![Textbox_Units].Value = "English"
    End Select
    'Updates the GRLUnits variable
    GRLUnits = Me![Units]
    'Calls the UnitsGRL subprocedure
    UnitsGRL
 Else
    'Displays a message warning the deletion of the current data
    'if the user attempts to change the units
    If MsgBox("Changing the Units will Erase the current GRL Data", vbOKCancel) = vbOK Then
      'Deletes the record if the user clicks OK
      DoCmd.RunCommand acCmdDeleteRecord
      'Resets the GRLUnits variable
      GRLUnits = 0
    Else
      'Displays the current units if the user clicks Cancel
      Me![Units] = GRLUnits
    End If
 End If
End Sub
```

```
Private Sub UnitsGRL()
   'Changes all the labels to the specified set of units
   Select Case GRLUnits
   Case 1
      Me![L Label].Caption = "L (m) "
      Me![Le Label].Caption = "Le (m) "
      Me![Area Label].Caption = "Area (m²)"
      Me![c Label].Caption = "c (m/s) "
      Me![Ec Label].Caption = "Ec (GN/m²) "
      Me![EA/c Label].Caption = "EA/c (KN-s/m) "
   Case 2
      Me![L Label].Caption = "L (ft) "
      Me![Le Label].Caption = "Le (ft) "
      Me![Area Label].Caption = "Area (ft²) "
      Me![c Label].Caption = "c (ft/s) "
      Me![Ec Label].Caption = "Ec (Tons/ft²) "
      Me![EA/c Label].Caption = "EA/c (Ton-s/ft) "
   Case Else
      Me![L Label].Caption = "L "
      Me![Le Label].Caption = "Le "
      Me![Area Label].Caption = "Area "
      Me![c Label].Caption = "c "
      Me![Ec Label].Caption = "Ec "
      Me![EA/c Label].Caption = "EA/c "
   End Select
End Sub

'All of the following subprocedures have the same structure, they assign a value
'to the SI units control based on the value input in the Raw data field and on
'the selected set of units
Private Sub L_AfterUpdate()
'Calls the UnitsOff subprocedure
UnitsOff
   'Calculates the value of the SI control
   Select Case GRLUnits
   Case 1
      Me![L (m)] = Me![L]
   Case 2
      Me![L (m)] = Me![L] * 0.3048
   End Select
End Sub

Private Sub Le_AfterUpdate()
UnitsOff
   Select Case GRLUnits
   Case 1
      Me![Le (m)] = Me![Le]
   Case 2
      Me![Le (m)] = Me![Le] * 0.3048
   End Select
End Sub

Private Sub Area_AfterUpdate()
UnitsOff
   Select Case GRLUnits
   Case 1
      Me![Area (m²)] = Me![Area]
   Case 2
      Me![Area (m²)] = Me![Area] * 0.0929
   End Select
End Sub
```

```
Private Sub c_AfterUpdate()
UnitsOff
  Select Case GRLUnits
  Case 1
    Me![c (m/s)] = Me![c]
  Case 2
    Me![c (m/s)] = Me![c] * 0.3048
  End Select
End Sub

Private Sub Ec_AfterUpdate()
UnitsOff
  Select Case GRLUnits
  Case 1
    Me![Ec (GN/m²)] = Me![Ec]
  Case 2
    Me![Ec (GN/m²)] = Me![Ec] * 0.00009576
  End Select
End Sub

Private Sub EA_c_AfterUpdate()
UnitsOff
  Select Case GRLUnits
  Case 1
    Me![EA/c  (KN-s/m)] = Me![EA/c]
  Case 2
    Me![EA/c  (KN-s/m)] = Me![EA/c] * 29.1878
  End Select
End Sub
```

frm_Insitu Tests

```
Option Compare Database
Option Explicit
Dim CurrentTest As String


Private Sub UnitsOff()
   'Displays a message if no units are selected prompting the user to select a set of units
   If IsNull(Me![Textbox_Units]) Then
      MsgBox "Please specify units"
      Me![Units].SetFocus
      'Erases the value assigned to the control
      DoCmd.RunCommand acCmdUndo
   End If
End Sub

Private Sub Form_Current()
   'Calls the ViewInsitu subprocedure
   ViewInsitu
   'Updates the value of the insitu test current record in the navigation box
   Me!CurrentRecordNumber.Value = Me.CurrentRecord
   'Updates the value of the total record number in the navigation box
   Me![TotalRecords].Value = DCount("[Units]", "InSitu Tests", " [ID Number] = (forms![frmGeneral]![ID Number])
")
   'Updates the Units control
   Select Case Me![Textbox_Units]
   Case "SI"
   Me![Units].Value = 1
   Case "English"
   Me![Units].Value = 2
   Case Else
   Me![Units].Value = 0
   End Select
   'Updates the InsituUnits variable
   InsituUnits = Me![Units]
   'Calls the UnitsInsitu subprocedure
   UnitsInsitu
End Sub

Private Sub Units_AfterUpdate()
 'Updates the Textbox_Units control if units are selected for the first time
 If InsituUnits = 0 Then
    Select Case Me![Units]
     Case 1
       Me![Textbox_Units].Value = "SI"
     Case 2
       Me![Textbox_Units].Value = "English"
    End Select
    'Updates the InsituUnits variable
    InsituUnits = Me![Units]
    'Calls the UnitsInsitu subprocedure
    UnitsInsitu
 Else
    'Displays a message warning the deletion of the current data
    'if the user attempts to change the units
    If MsgBox("Changing the Units will Erase the current Insitu Data Data", vbOKCancel) = vbOK Then
      'Deletes the record if the user clicks OK
      DoCmd.RunCommand acCmdDeleteRecord
      'Resets the InsituUnits variable
      InsituUnits = 0
```

```
    Else
      'Displays the current units if the user clicks Cancel
      Me![Units] = InsituUnits
    End If
  End If
End Sub

Private Sub Type_of_Test_Change()
  'Calls the UnitsOff subprocedure
  UnitsOff
  'Calls the ViewInsitu subprocedure if the Insitu test type is selected for the first time
  If CurrentTest = "No Test" Then
    ViewInsitu
  Else
   'Displays a message warning the deletion of the current data
   'if the user attempts to change the insitu test type
    If MsgBox("Changing the Test type will Erase the current Test Data", vbOKCancel) = vbOK Then
      'Deletes the record if the user clicks OK
      DoCmd.RunCommand acCmdDeleteRecord
      'Resets the CurrentTest variable
      CurrentTest = "No Test"
    Else
      'Displays the current type of test if the user clicks Cancel
      Me![Type of Test] = CurrentTest
    End If
  End If
End Sub

Private Sub ViewInsitu()
  'Resets the visible properties of the tab control
  Me![Box].Visible = True
  Me![CPT].Visible = False
  Me![DMT].Visible = False
  Me![SPT].Visible = False
 'Displays only the selected Insitu test tab and Updates the CurrentTest variable
  Select Case Me![Type of Test]
      Case "SPT"
      Me![SPT].Visible = True
      CurrentTest = "SPT"
      Case "CPT"
      Me![CPT].Visible = True
      CurrentTest = "CPT"
      Case "DMT"
      Me![DMT].Visible = True
      CurrentTest = "DMT"
      Case Else
      Me![Box].Visible = False
      CurrentTest = "No Test"
 End Select
End Sub

Private Sub UnitsInsitu()
  'Changes all the labels to the specified set of units
  Select Case Me![Units]
  Case 1
    Me![Water Table Elev Label].Caption = "Water Table Elev (m)"
    Me![Ground Surface Elev Label].Caption = "Ground Surface Elev (m)"
    Me![SPT Elev Label].Caption = "(m)"
    Me![SPT Depth Label].Caption = "(m)"
    Me![CPT Elev Label].Caption = "(m)"
    Me![CPT Depth Label].Caption = "(m)"
    Me![Cone Resistance Label].Caption = "(KN/m²)"
```

```
        Me![Friction Resistance Label].Caption = "(KN/m²)"
        Me![DMT Elev Label].Caption = "(m)"
        Me![DMT Depth Label].Caption = "(m)"
    Case 2
        Me![Water Table Elev Label].Caption = "Water Table Elev (ft)"
        Me![Ground Surface Elev Label].Caption = "Ground Surface Elev (ft)"
        Me![SPT Elev Label].Caption = "(ft)"
        Me![SPT Depth Label].Caption = "(ft)"
        Me![CPT Elev Label].Caption = "(ft)"
        Me![CPT Depth Label].Caption = "(ft)"
        Me![Cone Resistance Label].Caption = "(Ton/ft²)"
        Me![Friction Resistance Label].Caption = "(Ton/ft²)"
        Me![DMT Elev Label].Caption = "(ft)"
        Me![DMT Depth Label].Caption = "(ft)"
    Case Else
        Me![Water Table Elev Label].Caption = "Water Table Elev"
        Me![Ground Surface Elev Label].Caption = "Ground Surface Elev"
        Me![SPT Elev Label].Caption = ""
        Me![SPT Depth Label].Caption = ""
        Me![CPT Elev Label].Caption = ""
        Me![CPT Depth Label].Caption = ""
        Me![Cone Resistance Label].Caption = ""
        Me![Friction Resistance Label].Caption = ""
        Me![DMT Elev Label].Caption = ""
        Me![DMT Depth Label].Caption = ""
    End Select
End Sub

'Moves to previous record
Private Sub GoBack_Click()
On Error Resume Next
      DoCmd.GoToRecord , , acPrevious
End Sub

'Moves to Next record
Private Sub GoForward_Click()
On Error Resume Next
    DoCmd.GoToRecord , , acNext
End Sub

'Deletes current record
Private Sub DeleteRecord_Click()
On Error Resume Next
    DoCmd.RunCommand acCmdDeleteRecord
    DoCmd.GoToRecord , , acPrevious
End Sub

'Starts a new record
Private Sub AddRecord_Click()
On Error Resume Next
    DoCmd.GoToRecord , , acNewRec
End Sub

'All of the following subprocedures have the same structure, they assign a value
'to the SI units control based on the value input in the Raw data field and on
'the selected set of units
Private Sub Water_Table_Elev_AfterUpdate()
'Calls the UnitsOff subprocedure
UnitsOff
    'Calculates the value of the SI control
    Select Case InsituUnits
    Case 1
```

```
    Me![Water Table Elev (m)] = Me![Water Table Elev]
  Case 2
    Me![Water Table Elev (m)] = Me![Water Table Elev] * 0.3048
  End Select
End Sub


Private Sub Ground_Surface_Elev_AfterUpdate()
  UnitsOff
  Select Case InsituUnits
  Case 1
    Me![Ground Surface Elev (m)] = Me![Ground Surface Elev]
  Case 2
    Me![Ground Surface Elev (m)] = Me![Ground Surface Elev] * 0.3048
  End Select
End Sub
```

sbfSPT

```
Option Compare Database
Option Explicit

'All of the following subprocedures have the same structure. they assign a value
'to the SI units control based on the value input in the Raw data field and on
'the selected set of units
Private Sub Elev_AfterUpdate()
   'Calculates value of SI control
   Select Case InsituUnits
   Case 1
      Me![Elev (m)] = Me![Elev]
   Case 2
      Me![Elev (m)] = Me![Elev] * 0.3048
   End Select
End Sub

Private Sub Depth_AfterUpdate()
 Select Case InsituUnits
   Case 1
      Me![Depth (m)] = Me![Depth]
   Case 2
      Me![Depth (m)] = Me![Depth] * 0.3048
   End Select
End Sub
```

SbfCPT

```
Option Compare Database
Option Explicit

'All of the following subprocedures have the same structure. they assign a value
'to the SI units control based on the value input in the Raw data field and on
'the selected set of units
Private Sub Elev_AfterUpdate()
   'Calculates the value of the SI control
   Select Case InsituUnits
   Case 1
      Me![Elev (m)] = Me![Elev]
   Case 2
      Me![Elev (m)] = Me![Elev] * 0.3048
   End Select
End Sub

Private Sub Depth_AfterUpdate()
 Select Case InsituUnits
   Case 1
      Me![Depth (m)] = Me![Depth]
   Case 2
      Me![Depth (m)] = Me![Depth] * 0.3048
   End Select
End Sub

Private Sub Cone_resistance__qc_AfterUpdate()
   Select Case InsituUnits
   Case 1
      Me![Cone resistance  qc (KN/m²)] = Me![Cone resistance  qc]
   Case 2
      Me![Cone resistance  qc (KN/m²)] = Me![Cone resistance  qc] * 95.7605
   End Select
End Sub

Private Sub Friction_resistance__fs_AfterUpdate()
   Select Case InsituUnits
   Case 1
      Me![Friction resistance  fs (KN/m²)] = Me![Friction resistance  fs]
   Case 2
      Me![Friction resistance  fs (KN/m²)] = Me![Friction resistance  fs] * 95.7605
   End Select
End Sub
```

sbfDMT

```
Option Compare Database
Option Explicit

'All of the following subprocedures have the same structure, they assign a value
'to the SI units control based on the value input in the Raw data field and on
'the selected set of units
Private Sub Elev_AfterUpdate()
   'Calculates the value of the SI control
   Select Case InsituUnits
   Case 1
      Me![Elev (m)] = Me![Elev]
   Case 2
      Me![Elev (m)] = Me![Elev] * 0.3048
   End Select
End Sub

Private Sub Depth_AfterUpdate()
 Select Case InsituUnits
   Case 1
      Me![Depth (m)] = Me![Depth]
   Case 2
      Me![Depth (m)] = Me![Depth] * 0.3048
   End Select
End Sub
```

frmLoad Tests

```
Option Compare Database
Option Explicit
Dim CurrentTest As String

Private Sub UnitsOff()
  'Displays a message if no units are selected prompting the user to select a set of units
  If IsNull(Me![Textbox_Units]) Then
    MsgBox "Please specify units"
    Me![Units].SetFocus
    'Erases the value assigned to the control
    DoCmd.RunCommand acCmdUndo
    ViewLT
  End If
End Sub

Private Sub Form_Current()
  'Calls the ViewLT subprocedure
  ViewLT
  'Updates the value of the Load test current record in the navigation box
  Me!CurrentRecordNumber.Value = Me.CurrentRecord
  'Updates the value of the total record number in the navigation box
  Me![TotalRecords].Value = DCount("[Units]", "Load tests", " [ID Number] = (forms![frmGeneral]![ID Number]) ")
  'Updates the Units control
  Select Case Me![Textbox_Units]
  Case "SI"
  Me![Units].Value = 1
  Case "English"
  Me![Units].Value = 2
  Case Else
  Me![Units].Value = 0
  End Select
  'Updates the LoadTestUnits variable
  LoadTestUnits = Me![Units]
  'Calls the UnitsLT subprocedure
  UnitsLT
End Sub

Private Sub Units_AfterUpdate()
 'Updates the Textbox_Units control if units are selected for the first time
 If LoadTestUnits = 0 Then
    Select Case Me![Units]
     Case 1
       Me![Textbox_Units].Value = "SI"
     Case 2
       Me![Textbox_Units].Value = "English"
    End Select
    'Updates the LoadtestUnits variable
    LoadTestUnits = Me![Units]
    'Calls the UnitsLT subprocedure
    UnitsLT
 Else
    'Displays a message warning the deletion of the current data
    'if the user attempts to change the units
    If MsgBox("Changing the Units will Erase the current Load Test Data", vbOKCancel) = vbOK Then
      'Deletes the record if the user clicks OK
      DoCmd.RunCommand acCmdDeleteRecord
      'Resets the InsituUnits variable
      LoadTestUnits = 0
    Else
```

```
        'Displays the current units if the user clicks Cancel
        Me![Units] = LoadTestUnits
      End If
    End If
End Sub

Private Sub Type_of_Test_Change()
    'Calls the UnitsOff subprocedure
    UnitsOff
      'Calls the ViewLT subprocedure if the Insitu test type is selected for the first time
    If CurrentTest = "No Test" Then
      ViewLT
    Else
      'Displays a message warning the deletion of the current data
      'if the user attempts to change the insitu test type
      If MsgBox("Changing the Test type will Erase the current Test Data", vbOKCancel) = vbOK Then
        'Deletes the record if the user clicks OK
        DoCmd.RunCommand acCmdDeleteRecord
        'Resets the CurrentTest variable
        CurrentTest = "No Test"
      Else
        'Displays the current type of test if the user clicks Cancel
        Me![Type of Test] = CurrentTest
      End If
    End If
End Sub

Private Sub ViewLT()
    'Resets the visible properties of the tab control
    Me![Box].Visible = True
    Me![Statnamic test].Visible = False
    Me![Static test].Visible = False
   'Displays only the selected Load test tab and Updates the CurrentTest variable
    Select Case Me![Type of Test]
        Case "Static"
        Me![Static test].Visible = True
        CurrentTest = "Static"
        Case "Statnamic"
        Me![Statnamic test].Visible = True
        CurrentTest = "Statnamic"
        Case Else
        Me![Box].Visible = False
        CurrentTest = "No Test"
    End Select
End Sub

Private Sub UnitsLT()
    'Changes all the labels to the specified set of units
    Select Case LoadTestUnits
    Case 1
      Me![Static Time Label].Caption = "(min)"
      Me![Static Force Label].Caption = "(KN)"
      Me![Static Displacement Label].Caption = "(mm)"
      Me![Statnamic Time Label].Caption = "(ms)"
      Me![Statnamic Force Label].Caption = "(KN)"
      Me![Statnamic Displacement Label].Caption = "(mm)"
      Me![Velocity Label].Caption = "(m/s)"
      Me![Acceleration Label].Caption = "(m/s²)"
      Me![Max force Label].Caption = "Max. force (KN)"
      Me![Max Displacement Label].Caption = "Max. Displ. (mm)"
      Me![Permanent Displacement Label].Caption = "Permanent Displ.(mm)"
    Case 2
```

```
        Me![Static Time Label].Caption = "(min)"
        Me![Static Force Label].Caption = "(Ton)"
        Me![Static Displacement Label].Caption = "(in)"
        Me![Statnamic Time Label].Caption = "(ms)"
        Me![Statnamic Force Label].Caption = "(Tons)"
        Me![Statnamic Displacement Label].Caption = "(in)"
        Me![Velocity Label].Caption = "(ft/s)"
        Me![Acceleration Label].Caption = "(ft/s²)"
        Me![Max force Label].Caption = "Max. force (Ton)"
        Me![Max Displacement Label].Caption = "Max. Displ. (in)"
        Me![Permanent Displacement Label].Caption = "Permanent Displ.(in)"
    Case Else
        Me![Static Time Label].Caption = ""
        Me![Static Force Label].Caption = ""
        Me![Static Displacement Label].Caption = ""
        Me![Statnamic Time Label].Caption = ""
        Me![Statnamic Force Label].Caption = ""
        Me![Statnamic Displacement Label].Caption = ""
        Me![Velocity Label].Caption = ""
        Me![Acceleration Label].Caption = ""
        Me![Max force Label].Caption = "Max. force"
        Me![Max Displacement Label].Caption = "Max. Displ."
        Me![Permanent Displacement Label].Caption = "Permanent Displ."
    End Select
End Sub

'Moves to the previous record
Private Sub GoBack_Click()
On Error Resume Next
    DoCmd.GoToRecord , , acPrevious
End Sub

'Moves to the next record
Private Sub GoForward_Click()
On Error Resume Next
    DoCmd.GoToRecord , , acNext
End Sub

'Deletes the current record
Private Sub DeleteRecord_Click()
On Error Resume Next
    DoCmd.RunCommand acCmdDeleteRecord
    DoCmd.GoToRecord , , acPrevious
End Sub

'Starts a new record
Private Sub AddRecord_Click()
On Error Resume Next
    DoCmd.GoToRecord , , acNewRec
End Sub

'All of the following subprocedures have the same structure, they assign a value
'to the SI units control based on the value input in the Raw data field and on
'the selected set of units
Private Sub Max_Force_AfterUpdate()
    'Calls the UnitsOff subprocedure
    UnitsOff
    'Calculates the value of the SI control
    Select Case LoadTestUnits
    Case 1
        Me![Statnamic Max Force (KN)] = Me![Max Force]
    Case 2
```

```
    Me![Statnamic Max Force (KN)] = Me![Max Force] * 8.8964
  End Select
End Sub

Private Sub Max_Displacement_AfterUpdate()
  UnitsOff
  Select Case LoadTestUnits
  Case 1
    Me![Statnamic Max Displacement (mm)] = Me![Max Displacement]
  Case 2
    Me![Statnamic Max Displacement (mm)] = Me![Max Displacement] * 25.4
  End Select
End Sub

Private Sub Permanent_Displacement_AfterUpdate()
  UnitsOff
  Select Case LoadTestUnits
  Case 1
    Me![Statnamic Permanent Displacement (mm)] = Me![Permanent Displacement]
  Case 2
    Me![Statnamic Permanent Displacement (mm)] = Me![Permanent Displacement] * 25.4
  End Select
End Sub
```

sbfStatic

```
Option Compare Database
Option Explicit

'All of the following subprocedures have the same structure, they assign a value
'to the SI units control based on the value input in the Raw data field and on
'the selected set of units
Private Sub Force_AfterUpdate()
   'Calculates the value of the SI control
   Select Case LoadTestUnits
   Case 1
      Me![Force (KN)] = Me![Force]
   Case 2
      Me![Force (KN)] = Me![Force] * 8.8964
   End Select
End Sub

Private Sub Displacement_AfterUpdate()
   Select Case LoadTestUnits
   Case 1
      Me![Displacement (mm)] = Me![Displacement]
   Case 2
      Me![Displacement (mm)] = Me![Displacement] * 25.4
   End Select
End Sub
```

sbfStatnamic

```
Option Compare Database
Option Explicit

'All of the following subprocedures have the same structure. they assign a value
'to the SI units control based on the value input in the Raw data field and on
'the selected set of units
Private Sub Force_AfterUpdate()
   'Calculates the value of the SI control
   Select Case LoadTestUnits
   Case 1
      Me![Force (KN)] = Me![Force]
   Case 2
      Me![Force (KN)] = Me![Force] * 8.8964
   End Select
End Sub

Private Sub Displacement_AfterUpdate()
   Select Case LoadTestUnits
   Case 1
      Me![Displacement (mm)] = Me![Displacement]
   Case 2
      Me![Displacement (mm)] = Me![Displacement] * 25.4
   End Select
End Sub

Private Sub Velocity_AfterUpdate()
   Select Case LoadTestUnits
   Case 1
      Me![Velocity (m/s)] = Me![Velocity]
   Case 2
      Me![Velocity (m/s)] = Me![Velocity] * 0.3048
   End Select
End Sub

Private Sub Acceleration_AfterUpdate()
   Select Case LoadTestUnits
   Case 1
      Me![Acceleration (m/s²)] = Me![Acceleration]
   Case 2
      Me![Acceleration (m/s²)] = Me![Acceleration] * 0.3048
   End Select
End Sub
```

View-General

```
Option Compare Database
Option Explicit
Dim CurrentType As String


Private Sub Form_Current()
   DoCmd.Maximize
   'Updates the CurrentRecordNumber control on the navigation box
   Me![CurrentRecordNumber].Value = Me.CurrentRecord
   'Calls the ViewG subprocedure
   ViewG
End Sub

Private Sub ViewG()
   'Resets the visible property of the tab control
   Me!Box.Visible = False
   Me![Pile].Visible = True
   Me![Driving].Visible = True
   Me![Shaft].Visible = True
   Me![GRL Data].Visible = True
   CurrentType = "No Type"
   'Displays only tabs with information regarding the actual element type
   Select Case Me![Type]
        Case "Drilled Shaft"
        Me!Box.Visible = True
        Me![Pile].Visible = False
        Me![Driving].Visible = False
        Me![GRL Data].Visible = False
        CurrentType = "Drilled Shaft"
        Case "Driven Pile"
        Me!Box.Visible = True
        Me![Shaft].Visible = False
        CurrentType = "Driven Pile"
   End Select
End Sub

'Moves to the first record
Private Sub GoFirst_Click()
On Error Resume Next
     DoCmd.GoToRecord , , acFirst
End Sub

'Moves to the previous record
Private Sub GoBack_Click()
On Error Resume Next
     DoCmd.GoToRecord , , acPrevious
End Sub

'Moves to the next record
Private Sub GoForward_Click()
On Error Resume Next
     DoCmd.GoToRecord , , acNext
End Sub

'Moves to the last record
Private Sub GoLast_Click()
On Error Resume Next
     DoCmd.GoToRecord , , acLast
End Sub
```

```
'Closes the frmSearch and View_General forms
Private Sub MainMenuButton_Click()
On Error Resume Next
      DoCmd.Close acForm, "frmSearch"
      DoCmd.Close acForm, "View_General"
End Sub

'Closes the View_General form
Private Sub NewSearchButton_Click()
On Error Resume Next
      DoCmd.Close acForm, "View_General"
End Sub

'Opens the frmPrint form
Private Sub PrintButton_Click()
   ID = Me.[ID Number]
   DoCmd.OpenForm "frmPrint", acNormal, , , , acWindowNormal
End Sub

Private Sub EditButton_Click()
   'Opens frmGeneral
   DoCmd.OpenForm "frmGeneral", , , "[ID Number]=" & Me![ID Number]
   'Hides all the controls in the navigation box, except the delete record button
   Forms![frmGeneral]![ControlBox].Visible = False
   Forms![frmGeneral]![Browse].Visible = False
   Forms![frmGeneral]![Add Delete].Visible = False
   Forms![frmGeneral]![GoFirst].Visible = False
   Forms![frmGeneral]![GoBack].Visible = False
   Forms![frmGeneral]![GoForward].Visible = False
   Forms![frmGeneral]![GoLast].Visible = False
   Forms![frmGeneral]![AddRecord].Visible = False
   Forms![frmGeneral]![Element].Visible = False
   Forms![frmGeneral]![CurrentRecordNumber].Visible = False
   Forms![frmGeneral]![TotalRecords].Visible = False
End Sub
```

View_Insitu Tests

```
Option Compare Database
Option Explicit
Dim CurrentTest As String

Private Sub Form_Current()
   'Updates the CurrentRecordNumber control on the navigation box
   Me!CurrentRecordNumber.Value = Me.CurrentRecord
   'Updates the TotalRecords control on the navigation box
   Me![TotalRecords].Value = DCount("[Units]", "InSitu Tests", " [ID Number] = (forms![View_General]![ID
Number]) ")
   'Calls the ViewInsitu subprocedure
   ViewInsitu
   'Updates the Units control
   Select Case Me![Textbox_Units]
   Case "SI"
   Me![Units].Value = 1
   Case "English"
   Me![Units].Value = 2
   Case Else
   Me![Units].Value = 0
   End Select
   'Updates the InsituUnits variable
   InsituUnits = Me![Units]
   'Calls the UnitsInsitu subprocedure
   UnitsInsitu
End Sub

Private Sub ViewInsitu()
   'Resets the visible properties of the tab control
   Me![Box].Visible = True
   Me![CPT].Visible = False
   Me![DMT].Visible = False
   Me![SPT].Visible = False
   'Displays only tabs with information regarding the actual Insitu test
   Select Case Me![Type of Test]
        Case "SPT"
        Me![SPT].Visible = True
        CurrentTest = "SPT"
        Case "CPT"
        Me![CPT].Visible = True
        CurrentTest = "CPT"
        Case "DMT"
        Me![DMT].Visible = True
        CurrentTest = "DMT"
        Case Else
        Me![Box].Visible = False
        CurrentTest = "No Test"
  End Select
End Sub

Private Sub UnitsInsitu()
   'Changes the labels to the actual set of units
   Select Case Me![Units]
   Case 1
      Me![Water Table Elev Label].Caption = "Water Table Elev (m)"
      Me![Ground Surface Elev Label].Caption = "Ground Surface Elev (m)"
      Me![SPT Elev Label].Caption = "(m)"
      Me![SPT Depth Label].Caption = "(m)"
      Me![CPT Elev Label].Caption = "(m)"
      Me![CPT Depth Label].Caption = "(m)"
```

```
        Me![Cone Resistance Label].Caption = "(KN/m²)"
        Me![Friction Resistance Label].Caption = "(KN/m²)"
        Me![DMT Elev Label].Caption = "(m)"
        Me![DMT Depth Label].Caption = "(m)"
    Case 2
        Me![Water Table Elev Label].Caption = "Water Table Elev (ft)"
        Me![Ground Surface Elev Label].Caption = "Ground Surface Elev (ft)"
        Me![SPT Elev Label].Caption = "(ft)"
        Me![SPT Depth Label].Caption = "(ft)"
        Me![CPT Elev Label].Caption = "(ft)"
        Me![CPT Depth Label].Caption = "(ft)"
        Me![Cone Resistance Label].Caption = "(Ton/ft²)"
        Me![Friction Resistance Label].Caption = "(Ton/ft²)"
        Me![DMT Elev Label].Caption = "(ft)"
        Me![DMT Depth Label].Caption = "(ft)"
    Case Else
        Me![Water Table Elev Label].Caption = "Water Table Elev"
        Me![Ground Surface Elev Label].Caption = "Ground Surface Elev"
        Me![SPT Elev Label].Caption = ""
        Me![SPT Depth Label].Caption = ""
        Me![CPT Elev Label].Caption = ""
        Me![CPT Depth Label].Caption = ""
        Me![Cone Resistance Label].Caption = ""
        Me![Friction Resistance Label].Caption = ""
        Me![DMT Elev Label].Caption = ""
        Me![DMT Depth Label].Caption = ""
    End Select
End Sub

'Moves to first record
Private Sub GoFirst_Click()
On Error Resume Next
    DoCmd.GoToRecord , , acFirst
End Sub

'Moves to previous record
Private Sub GoBack_Click()
On Error Resume Next
    DoCmd.GoToRecord , , acPrevious
End Sub

'Moves to next record
Private Sub GoForward_Click()
On Error Resume Next
    DoCmd.GoToRecord , , acNext
End Sub

'Moves to last record
Private Sub GoLast_Click()
On Error Resume Next
    DoCmd.GoToRecord , , acLast
End Sub
```

View_Load Tests

```
Option Compare Database
Option Explicit
Dim CurrentTest As String


Private Sub Form_Current()
   'Calls the ViewLT subprocedure
   ViewLT
   'Updates the CurrentRecordNumber control on the navigation box
   Me!CurrentRecordNumber.Value = Me.CurrentRecord
   'Updates the TotalRecords control on the navigation box
   Me![TotalRecords].Value = DCount("[Units]", "Load tests", " [ID Number] = (forms![View_General]![ID Number])
")
   'Updates the Units control
   Select Case Me![Textbox_Units]
   Case "SI"
   Me![Units].Value = 1
   Case "English"
   Me![Units].Value = 2
   Case Else
   Me![Units].Value = 0
   End Select
   'Updates the LoadTestUnits variable
   LoadTestUnits = Me![Units]
   'Calls the UnitsLT subprocedure
   UnitsLT
End Sub


Private Sub ViewLT()
   'Resets the visible properties of the tab control
   Me![Box].Visible = True
   Me![Statnamic test].Visible = False
   Me![Static test].Visible = False
   'Displays only tabs with information regarding the actual Insitu test
   Select Case Me![Type of Test]
         Case "Static"
         Me![Static test].Visible = True
         CurrentTest = "Static"
         Case "Statnamic"
         Me![Statnamic test].Visible = True
         CurrentTest = "Statnamic"
         Case Else
         Me![Box].Visible = False
         CurrentTest = "No Test"
   End Select
End Sub

Private Sub UnitsLT()
   'Changes the labels to the actual set of units
   Select Case LoadTestUnits
   Case 1
      Me![Static Time Label].Caption = "(min)"
      Me![Static Force Label].Caption = "(KN)"
      Me![Static Displacement Label].Caption = "(mm)"
      Me![Statnamic Time Label].Caption = "(ms)"
      Me![Statnamic Force Label].Caption = "(KN)"
      Me![Statnamic Displacement Label].Caption = "(mm)"
      Me![Velocity Label].Caption = "(m/s)"
      Me![Acceleration Label].Caption = "(m/s²)"
```

```
        Me![Max force Label].Caption = "Max. force (KN)"
        Me![Max Displacement Label].Caption = "Max. Displ. (mm)"
        Me![Permanent Displacement Label].Caption = "Permanent Displ.(mm)"
    Case 2
        Me![Static Time Label].Caption = "(min)"
        Me![Static Force Label].Caption = "(Ton)"
        Me![Static Displacement Label].Caption = "(in)"
        Me![Statnamic Time Label].Caption = "(ms)"
        Me![Statnamic Force Label].Caption = "(Tons)"
        Me![Statnamic Displacement Label].Caption = "(in)"
        Me![Velocity Label].Caption = "(ft/s)"
        Me![Acceleration Label].Caption = "(ft/s²)"
        Me![Max force Label].Caption = "Max. force (Ton)"
        Me![Max Displacement Label].Caption = "Max. Displ. (in)"
        Me![Permanent Displacement Label].Caption = "Permanent Displ.(in)"
    Case Else
        Me![Static Time Label].Caption = ""
        Me![Static Force Label].Caption = ""
        Me![Static Displacement Label].Caption = ""
        Me![Statnamic Time Label].Caption = ""
        Me![Statnamic Force Label].Caption = ""
        Me![Statnamic Displacement Label].Caption = ""
        Me![Velocity Label].Caption = ""
        Me![Acceleration Label].Caption = ""
        Me![Max force Label].Caption = "Max. force"
        Me![Max Displacement Label].Caption = "Max. Displ."
        Me![Permanent Displacement Label].Caption = "Permanent Displ."
    End Select
End Sub

'Moves to first record
Private Sub GoFirst_Click()
On Error Resume Next
      DoCmd.GoToRecord , , acFirst
End Sub

'Moves to previous record
Private Sub GoBack_Click()
On Error Resume Next
   DoCmd.GoToRecord , , acPrevious
End Sub

'Moves to next record
Private Sub GoForward_Click()
On Error Resume Next
   DoCmd.GoToRecord , , acNext
End Sub


'Moves to last record
Private Sub GoLast_Click()
On Error Resume Next
      DoCmd.GoToRecord , , acLast
End Sub
```

frmMenu

```
Option Compare Database
Option Explicit

Private Sub Form_Current()
DoCmd.Maximize
End Sub

'Opens frmGeneral and starts a new record
Private Sub InputButton_Click()
On Error Resume Next
   DoCmd.OpenForm "frmGeneral", acNormal, , , , acWindowNormal
   DoCmd.GoToRecord acActiveDataObject, , acNewRec
End Sub

'Opens frmSearch
Private Sub SearchButton_Click()
On Error Resume Next
   DoCmd.OpenForm "frmSearch", acNormal, , , , acDialog
End Sub

'Quits Access
Private Sub QuitButton_Click()
On Error Resume Next
   DoCmd.quit
End Sub
```

frmSearch

```
Option Compare Database
Option Explicit

Dim BasicSELECT As String
Dim BasicFROM As String
Dim BasicWHERE As String
Dim ProjectNameWHERE As String
Dim ElemWHERE As String
Dim PileP As String
Dim PileFROM As String
Dim PileWHERE As String
Dim InsituP As String
Dim InsituFROM As String
Dim InsituWHERE As String
Dim TestP As String
Dim TestFROM As String
Dim TestWHERE As String
Dim MaxTotalLengthWHERE As String
Dim MinTotalLengthWHERE As String
Dim MaxEmbLengthWHERE As String
Dim MinEmbLengthWHERE As String
Dim MaxDiameterWHERE As String
Dim MinDiameterWHERE As String
Dim DrivingFROM As String
Dim DrivingP As String
Dim GRLFROM As String
Dim GRLP As String
Dim IGMFROM As String
Dim IGMP As String
Dim QueryCount As Long


Private Sub Query()
On Error Resume Next
   Set db = CurrentDb
   'Deletes the existing Search Query
   DoCmd.DeleteObject acQuery, "qrySearch"
   On Error GoTo 0
   'Creates the new Search Query
   Set qdfSearch = db.CreateQueryDef("qrySearch", SQL)
End Sub

Private Sub Form_Current()
  DoCmd.Maximize
  'Disables the Pile control, the user is not able to select a pile type
  'unless "Driven Pile" is selected from the Element control
  Me!Pile.Enabled = False
    'Creates the basic Criteria for querying all records
    BasicSELECT = "SELECT DISTINCTROW General.[ID Number], General.[X coord of Site]," & _
    "General.[Y coord of Site], General.[X coord of Element]," & _
    "General.[Y coord of Element], General.Type, General.[Project Name]," & _
    "General.[Project Number], General.[Submitting Company]," & _
    "General.[Submitting Engineer], General.Comments "
  BasicFROM = "[General] "
  BasicWHERE = " WHERE ([General].[ID Number])>0"
End Sub
```

```
Private Sub Search_Click()
   'Concatenates all criteria to form the SQL statement
   SQL = BasicSELECT & "FROM" & PileP & InsituP & TestP & DrivingP & GRLP & IGMP & BasicFROM &
PileFROM & _
   InsituFROM & TestFROM & DrivingFROM & GRLFROM & IGMFROM & BasicWHERE &
ProjectNameWHERE & ElemWHERE & PileWHERE & _
   InsituWHERE & TestWHERE & MaxTotalLengthWHERE & MinTotalLengthWHERE & _
   MaxDiameterWHERE & MinDiameterWHERE & MaxEmbLengthWHERE & MinEmbLengthWHERE
  'Calls the Query subprocedure
  Call Query
     QueryCount = DCount("[ID Number]", "qrySearch")
   'Displays a message if no records matching the specified criteria are found
   If QueryCount = 0 Then
   MsgBox "No records were found. Please try a different search criteria"
   Exit Sub
   End If
  'Opens the View_General form to display the search results
  DoCmd.OpenForm "View_General", acNormal, , , , acWindowNormal
End Sub

Private Sub Reset_Click()
   'Resets the Units control
   Me!Units.Value = 0
   'Resets all labels
   Me![Max Total Length Label].Caption = "Max Total Length"
   Me![Min Total Length Label].Caption = "Min Total Length"
   Me![Max Embedded Length Label].Caption = "Max Embedded Length"
   Me![Min Embedded Length Label].Caption = "Min Embedded Length"
   Me![Max Diameter Label].Caption = "Max Diameter"
   Me![Min Diameter label].Caption = "Min Diameter"
   'Resets all criteria
   ProjectNameWHERE = ""
   ElemWHERE = ""
   PileP = ""
   PileFROM = ""
   PileWHERE = ""
   InsituP = ""
   InsituFROM = ""
   InsituWHERE = ""
   TestP = ""
   TestFROM = ""
   TestWHERE = ""
   MaxTotalLengthWHERE = ""
   MinTotalLengthWHERE = ""
   MaxEmbLengthWHERE = ""
   MinEmbLengthWHERE = ""
   MaxDiameterWHERE = ""
   MinDiameterWHERE = ""
   DrivingFROM = ""
   DrivingP = ""
   GRLFROM = ""
   GRLP = ""
   IGMFROM = ""
   IGMP = ""
   'Resets all controls
   Me!Element.Value = 1
   Me!Pile.Value = 1
   Me!Insitu.Value = 1
   Me!Load.Value = 1
   Me![Max Total Length].Value = Null
   Me![Min Total Length].Value = Null
   Me![Max Embedded Length].Value = Null
```

```
    Me![Min Embedded Length].Value = Null
    Me![Max Diameter].Value = Null
    Me![Min Diameter].Value = Null
    Me![Project Name] = ""
    Me!Driving = False
    Me!GRL = False
    Me!IGM = False
    'Disables the controls involving units to force user to select units
    Me![Min Total Length].Enabled = False
    Me![Max Total Length].Enabled = False
    Me![Min Embedded Length].Enabled = False
    Me![Max Embedded Length].Enabled = False
    Me![Min Diameter].Enabled = False
    Me![Max Diameter].Enabled = False
End Sub


Private Sub Project_Name_AfterUpdate()
'Creates the Project Name criteria
    ProjectNameWHERE = " AND [General].[Project Name] Like '*" & Me![Project Name] & "*'"
End Sub

Private Sub Element_AfterUpdate()
    'Resets the Element Type criteria
    Me!Pile.Value = 1
    PileFROM = ""
    PileP = ""
    PileWHERE = ""
      'Creates the Element Type  Criteria
      Select Case Element
        Case 1
        Me!Pile.Enabled = False
        ElemWHERE = ""
        Case 2
        'Enables the Pile control to allow the user to choose a Pile Type
        Me!Pile.Enabled = True
        ElemWHERE = " AND ([General].[Type])= " & "'Driven Pile'"
        Case 3
        Me!Pile.Enabled = False
        ElemWHERE = " AND ([General].[Type])= " & "'Drilled Shaft'"
      End Select
End Sub

Private Sub Pile_AfterUpdate()
    'Creates the basic Pile type Criteria
    PileFROM = " INNER JOIN Piles ON [General].[ID Number] = Piles.[ID Number])"
    PileP = "("
    PileWHERE = ""
      'Completes pile criteria depending on the selected pile type
      Select Case Pile
        Case 1
        PileFROM = ""
        PileP = ""
        Case 2
        PileWHERE = " AND (Piles.[Pile Type])='Concrete Pile'"
        Case 3
        PileWHERE = " AND (Piles.[Pile Type])='H Steel Pile'"
        Case 4
        PileWHERE = " AND (Piles.[Pile Type])='Pipe Steel Pile'"
      End Select
End Sub
```

```
Private Sub Insitu_AfterUpdate()
   'Creates the basic Insitu test criteria
   InsituFROM = " INNER JOIN [InSitu Tests] ON [General].[ID Number] = [InSitu Tests].[ID Number])"
   InsituP = "("
   InsituWHERE = ""
    'Completes the Insitu test criteria depending on the selected test
    Select Case Insitu
      Case 1
      InsituFROM = ""
      InsituP = ""
      Case 2
      InsituWHERE = " AND ([InSitu Tests].[Test Type])='SPT'"
      Case 3
      InsituWHERE = " AND ([InSitu Tests].[Test Type])='CPT'"
      Case 4
      InsituWHERE = " AND ([InSitu Tests].[Test Type])='DMT'"
    End Select
End Sub

Private Sub Load_AfterUpdate()
   'Creates the basic Load test criteria
   TestFROM = "INNER JOIN [Load Tests] ON [General].[ID Number] = [Load Tests].[ID Number]) "
   TestP = "("
   TestWHERE = ""
   'Completes the Load test criteria depending on the selected test
    Select Case Load
      Case 1
      TestFROM = ""
      TestP = ""
      Case 2
      TestWHERE = " AND ([Load Tests].[Type of test])='Static'"
      Case 3
      TestWHERE = " AND ([Load Tests].[Type of test])='Statnamic'"
    End Select
End Sub

Private Sub Max_Total_Length_AfterUpdate()
Dim Length As String
   'Calculates the Max Total Length in SI units
   Select Case (Me![Textbox_Units])
   Case "SI"
      Length = Me![Max Total Length]
   Case "English"
      Length = Me![Max Total Length] * 0.3048
   End Select
   'Creates Max Total Length Criteria
   MaxTotalLengthWHERE = " AND [General].[Element Total Length (m)] <= " & Length
End Sub

Private Sub Min_Total_Length_AfterUpdate()
Dim Length As String
   'Calculates the Min Total Length in SI units
   Select Case (Me![Textbox_Units])
   Case "SI"
      Length = Me![Min Total Length]
   Case "English"
      Length = Me![Min Total Length] * 0.3048
   End Select
   'Creates Min Total Length Criteria
   MinTotalLengthWHERE = " AND [General].[Element Total Length (m)] >= " & Length
End Sub
Private Sub Max_Embedded_Length_AfterUpdate()
```

```
Dim Length As String
   'Calculates the Max Embedded Length in SI units
   Select Case (Me![Textbox_Units])
   Case "SI"
      Length = Me![Max Embedded Length]
   Case "English"
      Length = Me![Max Embedded Length] * 0.3048
   End Select
   'Creates Max Embedded Length Criteria
   MaxEmbLengthWHERE = " AND [General].[Element Embedded Length (m)] <= " & Length
End Sub
Private Sub Min_Embedded_Length_AfterUpdate()
Dim Length As String
   'Calculates the Min Embedded Length in SI units
   Select Case (Me![Textbox_Units])
   Case "SI"
      Length = Me![Min Embedded Length]
   Case "English"
      Length = Me![Min Embedded Length] * 0.3048
   End Select
   'Creates Min Embedded Length Criteria
   MinEmbLengthWHERE = " AND [General].[Element Embedded Length (m)] >= " & Length
End Sub
Private Sub Max_Diameter_AfterUpdate()
Dim Diameter As String
   'Calculates the Max Diameter in SI units
   Select Case (Me![Textbox_Units])
   Case "SI"
      Diameter = Me![Max Diameter]
   Case "English"
      Diameter = Me![Max Diameter] * 2.54
   End Select
   'Creates Max Diameter Criteria
   MaxDiameterWHERE = " AND [General].[Element Diameter (cm)] <= " & Diameter
End Sub

Private Sub Min_Diameter_AfterUpdate()
Dim Diameter As String
   'Calculates the Min Diameter in SI units
   Select Case (Me![Textbox_Units])
   Case "SI"
      Diameter = Me![Min Diameter]
   Case "English"
      Diameter = Me![Min Diameter] * 2.54
   End Select
   'Creates Min Diameter Criteria
   MinDiameterWHERE = " AND [General].[Element Diameter (cm)] >= " & Diameter
End Sub

Private Sub Driving_Click()
   If Me![Driving] = True Then
      'Creates Driving Criteria
      DrivingP = "("
      DrivingFROM = "INNER JOIN [Piles Penetration] ON [General].[ID Number] = [Piles Penetration].[ID
Number]) "
   Else
      'Resets Driving Criteria
      DrivingP = ""
      DrivingFROM = ""
   End If
End Sub
```

```
Private Sub GRL_Click()
   If Me![GRL] = True Then
      'Creates GRL Criteria
      GRLP = "("
      GRLFROM = "INNER JOIN [GRL DATA] ON [General].[ID Number] = [GRL Data].[ID Number]) "
   Else
      'Resets GRL Criteria
      GRLP = ""
      GRLFROM = ""
   End If
End Sub

Private Sub IGM_Click()
   If Me![IGM] = True Then
      'Creates IGM Criteria
      IGMP = "("
      IGMFROM = "INNER JOIN [Shafts IGM] ON [General].[ID Number] = [Shafts IGM].[ID Number]) "
   Else
      'Resets IGM Criteria
      IGMP = ""
      IGMFROM = ""
   End If
End Sub

'Closes Form
Private Sub MainMenuButton_Click()
On Error Resume Next
      DoCmd.Close acForm, "frmSearch"
End Sub


Private Sub Units_AfterUpdate()
   'Enables the controls involving units
   Me![Min Total Length].Enabled = True
   Me![Max Total Length].Enabled = True
   Me![Min Embedded Length].Enabled = True
   Me![Max Embedded Length].Enabled = True
   Me![Min Diameter].Enabled = True
   Me![Max Diameter].Enabled = True
   'Resets all controls involving Units
   Me![Max Total Length] = Null
   Me![Min Total Length] = Null
   Me![Max Embedded Length] = Null
   Me![Min Embedded Length] = Null
   Me![Max Diameter] = Null
   Me![Min Diameter] = Null
    'Displays all labels with the selected set of units
    Select Case Me![Units]
      Case 1
         Me![Textbox_Units].Value = "SI"
         Me![Max Total Length Label].Caption = "Max Total Length (m)"
         Me![Min Total Length Label].Caption = "Min Total Length (m)"
         Me![Max Embedded Length Label].Caption = "Max Embedded Length (m)"
         Me![Min Embedded Length Label].Caption = "Min Embedded Length (m)"
         Me![Max Diameter Label].Caption = "Max Diameter (cm)"
         Me![Min Diameter label].Caption = "Min Diameter (cm)"
      Case 2
         Me![Textbox_Units].Value = "English"
         Me![Max Total Length Label].Caption = "Max Total Length (ft)"
         Me![Min Total Length Label].Caption = "Min Total Length (ft)"
         Me![Max Embedded Length Label].Caption = "Max Embedded Length (ft)"
         Me![Min Embedded Length Label].Caption = "Min Embedded Length (ft)"
```

```
        Me![Max Diameter Label].Caption = "Max Diameter (in)"
        Me![Min Diameter label].Caption = "Min Diameter (in)"
    End Select
End Sub
```

frmPrint

```
Option Compare Database
Option Explicit
Dim Element As String
Dim ShaftSQL As String
Dim IGMSQL As String
Dim ConcreteSQL As String
Dim HSQL As String
Dim PipeSQL As String
Dim DrivingSQL As String
Dim GRLSQL As String
Dim SPTSQL As String
Dim CPTSQL As String
Dim DMTSQL As String
Dim StaticSQL As String
Dim StatnamicSQL As String
Private qdfShaft As QueryDef
Private qdfIGM As QueryDef
Private qdfConcrete As QueryDef
Private qdfH As QueryDef
Private qdfPipe As QueryDef
Private qdfDriving As QueryDef
Private qdfGRL As QueryDef
Private qdfSPT As QueryDef
Private qdfCPT As QueryDef
Private qdfDMT As QueryDef
Private qdfStatic As QueryDef
Private qdfStatnamic As QueryDef
Dim RecordCheck As Long

Private Sub Form_Current()
DoCmd.Maximize
End Sub

Private Sub CancelButton_Click()
'Closes the form
DoCmd.Close
End Sub

Private Sub OKButton_Click()
On Error Resume Next
   Set db = CurrentDb
      'Deletes all the existing queries related to the reports
      DoCmd.DeleteObject acQuery, "qry_Shafts Report"
      DoCmd.DeleteObject acQuery, "qry_IGM Report"
      DoCmd.DeleteObject acQuery, "qry_Concrete Pile Report"
      DoCmd.DeleteObject acQuery, "qry_H Pile Report"
      DoCmd.DeleteObject acQuery, "qry_Pipe Pile Report"
      DoCmd.DeleteObject acQuery, "qry_Driving Report"
      DoCmd.DeleteObject acQuery, "qry_GRL Data Report"
      DoCmd.DeleteObject acQuery, "qry_SPT Report"
      DoCmd.DeleteObject acQuery, "qry_CPT Report"
      DoCmd.DeleteObject acQuery, "qry_DMT Report"
      DoCmd.DeleteObject acQuery, "qry_Static Report"
      DoCmd.DeleteObject acQuery, "qry_Statnamic Report"
   On Error GoTo 0

   If Me![General] = True Then
      Select Case Element
         Case "Drilled Shaft"
            'Creates the query related to the Shafts reports
```

```
'If the current element type is "Drilled Shaft"
Set qdfShaft = db.CreateQueryDef("qry_Shafts Report", ShaftSQL)
'Creates the query related to the IGM report
'If the current element type is "Drilled Shaft"
Set qdfIGM = db.CreateQueryDef("qry_IGM Report", IGMSQL)
   'Prints the Shafts or English_Shafts report or export it to
   'excel or word, depending on the selected option and units
   Select Case (Me![Textbox_Units])
      Case "SI"
         Select Case (Me![Textbox_Option])
            Case "Print"
               DoCmd.OpenReport "Shafts Report"
            Case "Excel"
               DoCmd.OutputTo acOutputReport, "Shafts Report", acFormatXLS, , False
            Case "Word"
               DoCmd.OutputTo acOutputReport, "Shafts Report", acFormatRTF, , False
            End Select
      Case "English"
         Select Case (Me![Textbox_Option])
            Case "Print"
               DoCmd.OpenReport "English_Shafts Report"
            Case "Excel"
               DoCmd.OutputTo acOutputReport, "English_Shafts Report", acFormatXLS, , False
            Case "Word"
               DoCmd.OutputTo acOutputReport, "English_Shafts Report", acFormatRTF, , False
          End Select
   End Select

'Deletes the IGM query if no IGM data are found
RecordCheck = DCount("[ID Number]", "qry_IGM Report")
   If RecordCheck = 0 Then
    DoCmd.DeleteObject acQuery, "qry_IGM Report"
   Else
   'Prints the IGM or English_IGM report or export it to
   'excel or word, depending on the selected option and units
    Select Case (Me![Textbox_Units])
      Case "SI"
        Select Case (Me![Textbox_Option])
            Case "Print"
               DoCmd.OpenReport "IGM Report"
            Case "Excel"
               DoCmd.OutputTo acOutputReport, "IGM Report", acFormatXLS, , False
            Case "Word"
               DoCmd.OutputTo acOutputReport, "IGM Report", acFormatRTF, , False
         End Select
      Case "English"
        Select Case (Me![Textbox_Option])
            Case "Print"
               DoCmd.OpenReport "English_IGM Report"
            Case "Excel"
               DoCmd.OutputTo acOutputReport, "English_IGM Report", acFormatXLS, , False
            Case "Word"
               DoCmd.OutputTo acOutputReport, "English_IGM Report", acFormatRTF, , False
         End Select
     End Select
    End If

Case "Concrete Pile"
   'Creates the query related to the Concrete Pile reports
   'If the current pile type is "Concrete Pile"
   Set qdfConcrete = db.CreateQueryDef("qry_Concrete Pile Report", ConcreteSQL)
```

```
        'Prints the Concrete Pile or English_Concrete Pile report or export it to
        'excel or word, depending on the selected option and units
        Select Case (Me![Textbox_Units])
          Case "SI"
            Select Case (Me![Textbox_Option])
              Case "Print"
                DoCmd.OpenReport "Concrete Pile Report"
              Case "Excel"
                DoCmd.OutputTo acOutputReport, "Concrete Pile Report", acFormatXLS, , False
              Case "Word"
                DoCmd.OutputTo acOutputReport, "Concrete Pile Report", acFormatRTF, , False
            End Select
          Case "English"
            Select Case (Me![Textbox_Option])
              Case "Print"
                DoCmd.OpenReport "English_Concrete Pile Report"
              Case "Excel"
                DoCmd.OutputTo acOutputReport, "English_Concrete Pile Report", acFormatXLS, , False
              Case "Word"
                DoCmd.OutputTo acOutputReport, "English_Concrete Pile Report", acFormatRTF, , False
            End Select
        End Select

    Case "H Steel Pile"
      'Creates the query related to the H Pile reports
      'If the current pile type is "H Steel Pile"
      Set qdfH = db.CreateQueryDef("qry_H Pile Report", HSQL)
        'Prints the H Pile or English_H Pile report or export it to
        'excel or word, depending on the selected option and units
        Select Case (Me![Textbox_Units])
          Case "SI"
            Select Case (Me![Textbox_Option])
              Case "Print"
                DoCmd.OpenReport "H Pile Report"
              Case "Excel"
                DoCmd.OutputTo acOutputReport, "H Pile Report", acFormatXLS, , False
              Case "Word"
                DoCmd.OutputTo acOutputReport, "H Pile Report", acFormatRTF, , False
            End Select
          Case "English"
            Select Case (Me![Textbox_Option])
              Case "Print"
                DoCmd.OpenReport "English_H Pile Report"
              Case "Excel"
                DoCmd.OutputTo acOutputReport, "English_H Pile Report", acFormatXLS, , False
              Case "Word"
                DoCmd.OutputTo acOutputReport, "English_H Pile Report", acFormatRTF, , False
            End Select
        End Select

    Case "Pipe Steel Pile"
      'Creates the query related to the Pipe Pile reports
      'If the current pile type is "Steel Pipe Pile"
      Set qdfPipe = db.CreateQueryDef("qry_Pipe Pile Report", PipeSQL)
      'Prints the Pipe Pile or English_Pipe Pile report or export it to
      'excel or word, depending on the selected option and units
      Select Case (Me![Textbox_Units])
          Case "SI"
            Select Case (Me![Textbox_Option])
              Case "Print"
                DoCmd.OpenReport "Pipe Pile Report"
              Case "Excel"
```

```
                DoCmd.OutputTo acOutputReport, "Pipe Pile Report", acFormatXLS, , False
              Case "Word"
                DoCmd.OutputTo acOutputReport, "Pipe Pile Report", acFormatRTF, , False
            End Select
          Case "English"
            Select Case (Me![Textbox_Option])
              Case "Print"
                DoCmd.OpenReport "English_Pipe Pile Report"
              Case "Excel"
                DoCmd.OutputTo acOutputReport, "English_Pipe Pile Report", acFormatXLS, , False
              Case "Word"
                DoCmd.OutputTo acOutputReport, "English_Pipe Pile Report", acFormatRTF, , False
            End Select
        End Select
    End Select
End If

If Me![Driving] = True Then
    'Creates the query related to the Driving reports
    'If the current element type is "Driven Pile"
    Set qdfDriving = db.CreateQueryDef("qry_Driving Report", DrivingSQL)
    'Deletes the Driving query if no Driving data are found
    RecordCheck = DCount("[ID Number]", "qry_Driving Report")
      If RecordCheck = 0 Then
        DoCmd.DeleteObject acQuery, "qry_Driving Report"
      Else
        'Prints the Driving or English_Driving report or export it to
        'excel or word, depending on the selected option and units
        Select Case (Me![Textbox_Units])
          Case "SI"
            Select Case (Me![Textbox_Option])
                Case "Print"
                  DoCmd.OpenReport "Driving Report"
                Case "Excel"
                  DoCmd.OutputTo acOutputReport, "Driving Report", acFormatXLS, , False
                Case "Word"
                  DoCmd.OutputTo acOutputReport, "Driving Report", acFormatRTF, , False
            End Select
          Case "English"
            Select Case (Me![Textbox_Option])
                Case "Print"
                  DoCmd.OpenReport "English_Driving Report"
                Case "Excel"
                  DoCmd.OutputTo acOutputReport, "English_Driving Report", acFormatXLS, , False
                Case "Word"
                  DoCmd.OutputTo acOutputReport, "English_Driving Report", acFormatRTF, , False
            End Select
        End Select
      End If
End If

If Me![GRL] = True Then
    'Creates the query related to the GRL reports
    'If the current element type is "Driven Pile"
    Set qdfGRL = db.CreateQueryDef("qry_GRL Data Report", GRLSQL)
    'Deletes the GRL query if no GRL data are found
    RecordCheck = DCount("[ID Number]", "qry_GRL Data Report")
      If RecordCheck = 0 Then
        DoCmd.DeleteObject acQuery, "qry_GRL Data Report"
      Else
        'Prints the GRL or English_GRL data report or export it to
        'excel or word, depending on the selected option and units
```

```
        Select Case (Me![Textbox_Units])
          Case "SI"
            Select Case (Me![Textbox_Option])
                Case "Print"
                  DoCmd.OpenReport "GRL Data Report"
                Case "Excel"
                  DoCmd.OutputTo acOutputReport, "GRL Data Report", acFormatXLS, , False
                Case "Word"
                  DoCmd.OutputTo acOutputReport, "GRL Data Report", acFormatRTF, , False
            End Select
          Case "English"
            Select Case (Me![Textbox_Option])
                Case "Print"
                  DoCmd.OpenReport "English_GRL Data Report"
                Case "Excel"
                  DoCmd.OutputTo acOutputReport, "English_GRL Data Report", acFormatXLS, , False
                Case "Word"
                  DoCmd.OutputTo acOutputReport, "English_GRL Data Report", acFormatRTF, , False
            End Select
        End Select
      End If
End If

If Me![SPT] = True Then
   'Creates the query related to the SPT reports
   Set qdfSPT = db.CreateQueryDef("qry_SPT Report", SPTSQL)
   'Deletes the SPT query if no SPT data are found
   RecordCheck = DCount("[ID Number]", "qry_SPT Report")
     If RecordCheck = 0 Then
       DoCmd.DeleteObject acQuery, "qry_SPT Report"
     Else

       'Prints the SPT or English_SPT report or export it to
       'excel or word, depending on the selected option and units
       Select Case (Me![Textbox_Units])
         Case "SI"
           Select Case (Me![Textbox_Option])
               Case "Print"
                 DoCmd.OpenReport "SPT Report"
               Case "Excel"
                 DoCmd.OutputTo acOutputReport, "SPT Report", acFormatXLS, , False
               Case "Word"
                 DoCmd.OutputTo acOutputReport, "SPT Report", acFormatRTF, , False
           End Select
         Case "English"
           Select Case (Me![Textbox_Option])
               Case "Print"
                 DoCmd.OpenReport "English_SPT Report"
               Case "Excel"
                 DoCmd.OutputTo acOutputReport, "English_SPT Report", acFormatXLS, , False
               Case "Word"
                 DoCmd.OutputTo acOutputReport, "English_SPT Report", acFormatRTF, , False
           End Select
       End Select
     End If
End If

If Me![CPT] = True Then
   'Creates the query related to the CPT reports
   Set qdfCPT = db.CreateQueryDef("qry_CPT Report", CPTSQL)
   'Deletes the CPT query if no CPT data are found
   RecordCheck = DCount("[ID Number]", "qry_CPT Report")
```

```
    If RecordCheck = 0 Then
      DoCmd.DeleteObject acQuery, "qry_CPT Report"
    Else
      'Prints the CPT or English_CPT report or export it to
      'excel or word, depending on the selected option and units
      Select Case (Me![Textbox_Units])
        Case "SI"
          Select Case (Me![Textbox_Option])
              Case "Print"
                DoCmd.OpenReport "CPT Report"
              Case "Excel"
                DoCmd.OutputTo acOutputReport, "CPT Report", acFormatXLS, , False
              Case "Word"
                DoCmd.OutputTo acOutputReport, "CPT Report", acFormatRTF, , False
          End Select
        Case "English"
          Select Case (Me![Textbox_Option])
              Case "Print"
                DoCmd.OpenReport "English_CPT Report"
              Case "Excel"
                DoCmd.OutputTo acOutputReport, "English_CPT Report", acFormatXLS, , False
              Case "Word"
                DoCmd.OutputTo acOutputReport, "English_CPT Report", acFormatRTF, , False
          End Select
      End Select
    End If
End If

If Me![DMT] = True Then
  'Creates the query related to the DMT reports
  Set qdfDMT = db.CreateQueryDef("qry_DMT Report", DMTSQL)
  'Deletes the DMT query if no DMT data are found
  RecordCheck = DCount("[ID Number]", "qry_DMT Report")
    If RecordCheck = 0 Then
      DoCmd.DeleteObject acQuery, "qry_DMT Report"
    Else
      'Prints the DMT or English_DMT report or export it to
      'excel or word, depending on the selected option and units
      Select Case (Me![Textbox_Units])
        Case "SI"
          Select Case (Me![Textbox_Option])
              Case "Print"
                DoCmd.OpenReport "DMT Report"
              Case "Excel"
                DoCmd.OutputTo acOutputReport, "DMT Report", acFormatXLS, , False
              Case "Word"
                DoCmd.OutputTo acOutputReport, "DMT Report", acFormatRTF, , False
          End Select
        Case "English"
          Select Case (Me![Textbox_Option])
              Case "Print"
                DoCmd.OpenReport "English_DMT Report"
              Case "Excel"
                DoCmd.OutputTo acOutputReport, "English_DMT Report", acFormatXLS, , False
              Case "Word"
                DoCmd.OutputTo acOutputReport, "English_DMT Report", acFormatRTF, , False
          End Select
      End Select
    End If
End If

If Me![Static] = True Then
```

```
    'Creates the query related to the Static Test reports
    Set qdfStatic = db.CreateQueryDef("qry_Static Report", StaticSQL)
    'Deletes the Static Test query if no Static Test data are found
    RecordCheck = DCount("[ID Number]", "qry_Static Report")
      If RecordCheck = 0 Then
        DoCmd.DeleteObject acQuery, "qry_Static Report"
      Else
        'Prints the Static or English_Static Test report or export it to
        'excel or word, depending on the selected option and units
        Select Case (Me![Textbox_Units])
          Case "SI"
            Select Case (Me![Textbox_Option])
                Case "Print"
                  DoCmd.OpenReport "Static Test Report"
                Case "Excel"
                  DoCmd.OutputTo acOutputReport, "Static Test Report", acFormatXLS, , False
                Case "Word"
                  DoCmd.OutputTo acOutputReport, "Static Test Report", acFormatRTF, , False
            End Select
          Case "English"
            Select Case (Me![Textbox_Option])
                Case "Print"
                  DoCmd.OpenReport "English_Static Test Report"
                Case "Excel"
                  DoCmd.OutputTo acOutputReport, "English_Static Test Report", acFormatXLS, , False
                Case "Word"
                  DoCmd.OutputTo acOutputReport, "English_Static Test Report", acFormatRTF, , False
            End Select
        End Select
      End If
End If

If Me![Statnamic] = True Then
    'Creates the query related to the Statnamic Test reports
    Set qdfStatnamic = db.CreateQueryDef("qry_Statnamic Report", StatnamicSQL)
    'Deletes the Statnamic Test query if no Statnamic Test data are found
    RecordCheck = DCount("[ID Number]", "qry_Statnamic Report")
      If RecordCheck = 0 Then
        DoCmd.DeleteObject acQuery, "qry_Statnamic Report"
      Else
        'Prints the Statnamic or English_Statnamic Test report or export it to
        'excel or word, depending on the selected option and units
        Select Case (Me![Textbox_Units])
          Case "SI"
            Select Case (Me![Textbox_Option])
                Case "Print"
                  DoCmd.OpenReport "Statnamic Test Report"
                Case "Excel"
                  DoCmd.OutputTo acOutputReport, "Statnamic Test Report", acFormatXLS, , False
                Case "Word"
                  DoCmd.OutputTo acOutputReport, "Statnamic Test Report", acFormatRTF, , False
            End Select
          Case "English"
            Select Case (Me![Textbox_Option])
                Case "Print"
                  DoCmd.OpenReport "English_Statnamic Test Report"
                Case "Excel"
                  DoCmd.OutputTo acOutputReport, "English_Statnamic Test Report", acFormatXLS, , False
                Case "Word"
                  DoCmd.OutputTo acOutputReport, "English_Statnamic Test Report", acFormatRTF, , False
            End Select
        End Select
```

```
        End If
    End If
End Sub

Private Sub General_Click()
    'Resets the SQL statements
    ShaftSQL = ""
    ConcreteSQL = ""
    HSQL = ""
    PipeSQL = ""

    If Me![General] = True Then
        'Creates the SQL statements for the Shafts and IGM queries if the Element Type is "Drilled Shaft"
        If (Forms![View_General].Form![Type]) = "Drilled Shaft" Then
            Element = "Drilled Shaft"
            ShaftSQL = "SELECT General.[ID Number], General.[X coord of Site], General.[Y coord of Site], General.[X coord of Element], General.[Y coord of Element], General.[Project Name], General.[Project Number], General.[Submitting Company], General.[Submitting Engineer], General.Comments, Shafts.[Construction Method], Shafts.[Type of Base], Shafts.[Diameter (cm)], Shafts.[Total Length (m)], Shafts.[Embedded Length (m)], Shafts.[Casing Length (m)], Shafts.[Bell Diameter (cm)], Shafts.[Bell Length (m)], Shafts.[Slump of Concrete(mm)], Shafts.[Concrete strength f'c (KN/m²)], Shafts.[Core Modulus Ec (KN/m²)], Shafts.[Top Elev (m)], Shafts.[Tip Elev (m)], Shafts.[Scour Elev (m)], Shafts.[Rock Socket Length (m)], Shafts.[Design Load (kN)], Shafts.[As Built Diameter (cm)], Shafts.[As Built Tip Elev (m)], Shafts.[As Built Total length (m)], Shafts.[As Built Rock Socket Length (m)], Shafts.[As built Design Load (kN)]" & _
                " FROM [General] INNER JOIN Shafts ON General.[ID Number] = Shafts.[ID Number]" & _
                " WHERE (((General.[ID Number])=" & ID & "))"

            IGMSQL = "SELECT General.[ID Number], General.[Project Name], General.[Project Number], [Shafts IGM].[Elev (m)], [Shafts IGM].[Depth (m)], [Shafts IGM].[qu (KN/m²)], [Shafts IGM].[qt (KN/m²)], [Shafts IGM].[Em (KN/m²)], [Shafts IGM].[Ei (KN/m²)], [Shafts IGM].[Surface Type]" & _
                " FROM [General] INNER JOIN [Shafts IGM] ON General.[ID Number] = [Shafts IGM].[ID Number]" & _
                " WHERE (((General.[ID Number])=" & ID & "))"
        Else
            'Creates the SQL statements for Concrete, H or Pipe Piles depending on the pile Type
            Select Case (Forms![View_General]![Driven Piles].Form.[Pile Type])
                Case "Concrete Pile"
                    Element = "Concrete Pile"
                    ConcreteSQL = "SELECT General.[ID Number], General.[X coord of Site], General.[Y coord of Site], General.[X coord of Element], General.[Y coord of Element], General.[Project Name], General.[Project Number], General.[Submitting Company], General.[Submitting Engineer], General.Comments, Piles.[Pile Type], Piles.[Type of Load test], Piles.[Pile Elastic Modulus (KN/m²)], Piles.[Capwap Elastic Modulus (KN/m²)], Piles.[Pile Wave Speed c (m/s)], Piles.[Pile Impedance E*A/c (KN-s/m)], [Piles Concrete].[Pile Description], [Piles Concrete].[Width (cm)], [Piles Concrete].[Void Diameter (cm)], [Piles Concrete].[Cross Sectional Area (cm²)], [Piles Concrete].[Total Length (m)], [Piles Concrete].[Embedded Length (m)], [Piles Concrete].[Total Weight (KN)], [Piles Concrete].[Unit Weight of Concrete (KN/m^3)], [Piles Concrete].[Concrete Strength (KN/m²)]" & _
                        " FROM ([General] INNER JOIN [Piles Concrete] ON General.[ID Number] = [Piles Concrete].[ID Number]) INNER JOIN Piles ON (Piles.[ID Number] = [Piles Concrete].[ID Number]) AND (General.[ID Number] = Piles.[ID Number])" & _
                        " WHERE (((General.[ID Number])=" & ID & "))"
                Case "H Steel Pile"
                    Element = "H Steel Pile"
                    HSQL = "SELECT General.[ID Number], General.[X coord of Site], General.[Y coord of Site], General.[X coord of Element], General.[Y coord of Element], General.[Project Name], General.[Project Number], General.[Submitting Company], General.[Submitting Engineer], General.Comments, Piles.[Pile Type], Piles.[Type of Load test], Piles.[Pile Elastic Modulus (KN/m²)], Piles.[Capwap Elastic Modulus (KN/m²)], Piles.[Pile Wave Speed c (m/s)], Piles.[Pile Impedance E*A/c (KN-s/m)], [Piles H Steel].[Pile Description], [Piles H Steel].[HP Depth Designation Number], [Piles H Steel].[Hp Weight Designation Number], [Piles H Steel].[Steel Cross Sectional Area (cm²)], [Piles H Steel].[Pile Cross Sectional Area (cm²)], [Piles H Steel].[Steel Section Perimeter (cm)], [Piles H Steel].[Pile Perimeter (cm)], [Piles H Steel].[Weight of Steel (KN/m)], [Piles H Steel].[Weight of pile (KN)], [Piles H Steel].[Total Length (m)], [Piles H Steel].[Embedded Length (m)], " & _
                        "[Piles H Steel].[Unit Weight of Steel (KN/m^3)], [Piles H Steel].[Concrete Strength (KN/m²)]" & _
```

```
                " FROM ([General] INNER JOIN [Piles H Steel] ON General.[ID Number] = [Piles H Steel].[ID
Number]) INNER JOIN Piles ON (Piles.[ID Number] = [Piles H Steel].[ID Number]) AND (General.[ID Number] =
Piles.[ID Number])" & _
                " WHERE (((General.[ID Number])=" & ID & "))"
            Case "Pipe Steel Pile"
              Element = "Pipe Steel Pile"
              PipeSQL = "SELECT General.[ID Number], General.[X coord of Site], General.[Y coord of Site],
General.[X coord of Element], General.[Y coord of Element], General.[Project Name], General.[Project Number],
General.[Submitting Company], General.[Submitting Engineer], General.Comments, Piles.[Pile Type], Piles.[Type of
Load test], Piles.[Pile Elastic Modulus (KN/m²)], Piles.[Capwap Elastic Modulus (KN/m²)], Piles.[Pile Wave Speed c
(m/s)], Piles.[Pile Impedance E*A/c (KN-s/m)], [Piles Pipe Steel].[End Conditon], [Piles Pipe Steel].[Outside Diameter
(cm)], [Piles Pipe Steel].[Thickness of Steel (cm)], [Piles Pipe Steel].[Steel Cross Sectional Area (cm²)], [Piles Pipe
Steel].[Pile Cross Sectional Area (cm²)], [Piles Pipe Steel].[Weight of Steel (KN/m)], [Piles Pipe Steel].[Weight of pile
(KN)], [Piles Pipe Steel].[Total Length (m)], [Piles Pipe Steel].[Embedded Length (m)], [Piles Pipe Steel].[Unit Weight
of Steel (KN/m^3)], [Piles Pipe Steel].[Concrete Strength (KN/m²)]" & _
                " FROM ([General] INNER JOIN [Piles Pipe Steel] ON General.[ID Number] = [Piles Pipe
Steel].[ID Number]) INNER JOIN Piles ON (Piles.[ID Number] = [Piles Pipe Steel].[ID Number]) AND (General.[ID
Number] = Piles.[ID Number])" & _
                " WHERE (((General.[ID Number])=" & ID & "))"
          End Select
        End If
      Else
        'Resets the SQL statements if the general option is turned off
        ShaftSQL = ""
        ConcreteSQL = ""
        HSQL = ""
        PipeSQL = ""
      End If
End Sub

Private Sub Driving_Click()
      If Me![Driving] = True Then
        'Creates the SQL statement for the Driving query
        DrivingSQL = "SELECT General.[Project Name], General.[Project Number], [Piles Driving].[ID Number], [Piles
Driving].[Hammer Type], [Piles Driving].[Weight (N)], [Piles Driving].[Energy (KN-m)], [Piles Driving].[Pre Bored
Depth (m)], [Piles Driving].[Last Blow (cm)], [Piles Driving].[End of Driving (cm)], [Piles Driving].[Start of Restrike
(cm)], [Piles Penetration].[Depth (m)], [Piles Penetration].[Penetration (Blows /30,48cm)]" & _
                " FROM ([Piles Penetration] INNER JOIN [General] ON [Piles Penetration].[ID Number] = General.[ID
Number]) INNER JOIN [Piles Driving] ON ([Piles Driving].[ID Number] = [Piles Penetration].[ID Number]) AND
(General.[ID Number] = [Piles Driving].[ID Number])" & _
                " WHERE ((([Piles Driving].[ID Number])=" & ID & "))"
      Else
        'Resets the SQL statement if the Driving option is turned off
        DrivingSQL = ""
      End If
End Sub

Private Sub GRL_Click()
      If Me![GRL] = True Then
        'Creates the SQL statement for the GRL query
        GRLSQL = "SELECT General.[Project Name], General.[Project Number], [GRL Data].[ID Number], [GRL
Data].[L (m)], [GRL Data].[Le (m)], [GRL Data].[Area (m²)], [GRL Data].[c (m/s)], [GRL Data].[dt (sec)], [GRL
Data].[Ec (GN/m²)], [GRL Data].[EA/c  (KN-s/m)], [GRL PDA Trace].[Time (s)], [GRL PDA Trace].[dv_v
(volt/100)], [GRL PDA Trace].[v-v (volt/100)], [GRL PDA Trace].[dv-F (volt/100)], [GRL PDA Trace].[v-F
(volt/100)]" & _
              " FROM ([General] INNER JOIN [GRL PDA Trace] ON General.[ID Number] = [GRL PDA Trace].[ID
Number]) INNER JOIN [GRL Data] ON ([GRL Data].[ID Number] = [GRL PDA Trace].[ID Number]) AND
(General.[ID Number] = [GRL Data].[ID Number])" & _
              " WHERE ((([GRL Data].[ID Number])=" & ID & "))"
      Else
        'Resets the SQL statement if the GRL option is turned off
        GRLSQL = ""
```

```
  End If
End Sub

Private Sub SPT_Click()
  If Me![SPT] = True Then
    'Creates the SQL statement for the SPT query
    SPTSQL = "SELECT General.[Project Name], General.[Project Number], [InSitu Tests].[ID Number], [InSitu
Tests].[Insitu test ID], [InSitu Tests].[X coord of Test], [InSitu Tests].[Y coord of Test], [InSitu Tests].[Station #],
[InSitu Tests].Comments, [InSitu Tests].[Ground Surface Elev (m)], [InSitu Tests].[Water Table Elev (m)], [Insitu
SPT].[Elev (m)], [Insitu SPT].[Depth (m)], [Insitu SPT].[SPT N], [Insitu SPT].[Soil Description], [Insitu SPT].USCS "
& _
        " FROM ([General] INNER JOIN [InSitu Tests] ON General.[ID Number] = [InSitu Tests].[ID Number])
INNER JOIN [Insitu SPT] ON [InSitu Tests].[Insitu test ID] = [Insitu SPT].[Insitu test ID]" & _
        " WHERE ((([InSitu Tests].[ID Number])=" & ID & "))"
  Else
    'Resets the SQL statement if the SPT option is turned off
    SPTSQL = ""
  End If
End Sub

Private Sub CPT_Click()
  If Me![CPT] = True Then
    'Creates the SQL statement for the CPT query
    CPTSQL = "SELECT General.[Project Name], General.[Project Number], [InSitu Tests].[ID Number], [InSitu
Tests].[Insitu test ID], [InSitu Tests].[X coord of Test], [InSitu Tests].[Y coord of Test], [InSitu Tests].[Station #],
[InSitu Tests].[Ground Surface Elev (m)], [InSitu Tests].[Water Table Elev (m)], [InSitu Tests].Comments, [Insitu
CPT].[Elev (m)], [Insitu CPT].[Depth (m)], [Insitu CPT].[Cone resistance qc (KN/m²)], [Insitu CPT].[Friction
resistance fs (KN/m²)], [Insitu CPT].[Friction ratio Rf]" & _
        " FROM ([General] INNER JOIN [InSitu Tests] ON General.[ID Number] = [InSitu Tests].[ID Number])
INNER JOIN [Insitu CPT] ON [InSitu Tests].[Insitu test ID] = [Insitu CPT].[Insitu test ID]" & _
        " WHERE ((([InSitu Tests].[ID Number])=" & ID & "))"
  Else
    'Resets the SQL statement if the CPT option is turned off
    CPTSQL = ""
  End If
End Sub

Private Sub DMT_Click()
  If Me![DMT] = True Then
    'Creates the SQL statement for the DMT query
    DMTSQL = "SELECT General.[Project Name], General.[Project Number], [InSitu Tests].[ID Number], [InSitu
Tests].[Insitu test ID], [InSitu Tests].[X coord of Test], [InSitu Tests].[Y coord of Test], [InSitu Tests].[Station #],
[InSitu Tests].[Ground Surface Elev (m)], [InSitu Tests].[Water Table Elev (m)], [InSitu Tests].Comments, [InSitu
Tests].[DMT Delta A (bars)], [InSitu Tests].[DMT Delta B (bars)], [InSitu Tests].[DMT Gauge Zero (bars)], [Insitu
DMT].[Elev (m)], [Insitu DMT].[Depth (m)], [Insitu DMT].[Thrust (Kg)], [Insitu DMT].[Reading A (bars)], [Insitu
DMT].[Reading B (bars)], [Insitu DMT].[Reading C (bars)]" & _
        " FROM ([General] INNER JOIN [InSitu Tests] ON General.[ID Number] = [InSitu Tests].[ID Number])
INNER JOIN [Insitu DMT] ON [InSitu Tests].[Insitu test ID] = [Insitu DMT].[Insitu test ID]" & _
        " WHERE ((([InSitu Tests].[ID Number])=" & ID & "))"
  Else
    'Resets the SQL statement if the DMT option is turned off
    DMTSQL = ""
  End If
End Sub

Private Sub Static_Click()
  If Me![Static] = True Then
  'Creates the SQL statement for the Static Test query
    StaticSQL = "SELECT General.[Project Name], General.[Project Number], [Load Tests].[ID Number], [Load
Tests].[Load test ID], [Load Tests].Comments, [Load Test Static].[Time (min)], [Load Test Static].[Force (KN)],
[Load Test Static].[Displacement (mm)], [Load Test Static].Counter" & _
```

```
            " FROM ([General] INNER JOIN [Load Tests] ON General.[ID Number] = [Load Tests].[ID Number])
INNER JOIN [Load Test Static] ON [Load Tests].[Load test ID] = [Load Test Static].[Load test ID]" & _
            " WHERE ((([Load Tests].[ID Number])=" & ID & "))"
  Else
    'Resets the SQL statement if the Static Test option is turned off
    StaticSQL = ""
  End If
End Sub

Private Sub Statnamic_Click()
  If Me![Statnamic] = True Then
    'Creates the SQL statement for the Statnamic Test query
    StatnamicSQL = "SELECT General.[Project Name], General.[Project Number], [Load Tests].[ID Number], [Load
Tests].[Load test ID], [Load Tests].Comments, [Load Tests].[Statnamic Max Force (KN)], [Load Tests].[Statnamic
Max Displacement (mm)], [Load Tests].[Statnamic Permanent Displacement (mm)], [Load Test Statnamic].[Time
(ms)], [Load Test Statnamic].[Force (KN)], [Load Test Statnamic].[Displacement (mm)], [Load Test
Statnamic].[Velocity (m/s)], [Load Test Statnamic].[Acceleration (m/s²)]" & _
              " FROM ([General] INNER JOIN [Load Tests] ON General.[ID Number] = [Load Tests].[ID Number])
INNER JOIN [Load Test Statnamic] ON [Load Tests].[Load test ID] = [Load Test Statnamic].[Load test ID]" & _
              " WHERE ((([Load Tests].[ID Number])=" & ID & "))"
  Else
    'Resets the SQL statement if the Statnamic Test option is turned off
    StatnamicSQL = ""
  End If
End Sub

Private Sub Units_AfterUpdate()
  'Updates the Textbox_Units control
  Select Case (Me!Units)
    Case 1
      Me![Textbox_Units].Value = "SI"
    Case 2
      Me![Textbox_Units].Value = "English"
  End Select
End Sub

Private Sub Option_AfterUpdate()
  'Updates the Textbox_Option control
  Select Case (Me!Option)
    Case 1
      Me![Textbox_Option].Value = "Print"
    Case 2
      Me![Textbox_Option].Value = "Excel"
    Case 3
      Me![Textbox_Option].Value = "Word"
  End Select
End Sub
```