Data-analysis Tool for
Metal Concentrations of the Gulf of Mexico

Graduate Project Report

by
Praveen C Gondra

Committee Members

Dr. M. Guimaraes, Chairman

Dr. Stephen Dannelly, Member

Dr. David R. Thomas, Member

Texas A&M University - Corpus Christi
Department of Computer Science
# Table of Contents

1. Title Page i

2. Table of contents ii

3. Abstract 1

4. Background and rationale 2

5. Narrative 4
   1. Database Component 4
   2. Graphical Presentation Component 8
   3. User Guide 9

6. Project Environment 11

7. Procedure 12
   1. Oracle Web Server 13
   2. Database Tables 16
   3. Packages, Procedures and Functions 19

8. Results and Future Work 26

9. References 27

10. List of Figures
    1. Oracle Web Server and the tool 28
    2. ER-Diagram 31
    3. Operational Flow Diagram 32
    4. Forms and Reports 34
    5. Matrix of Forms vs tables 54

11. Appendix 28
    1. Appendix A (Fig. Explaining ows and tool) 28
2. Appendix B (Tables and fields used) 29
3. Appendix C (ER-Diagram) 31
4. Appendix D (Operational Flow Diagram) 32
5. Appendix E (Figures of Forms and Reports) 33
6. Appendix F (Matrix of Procedures vs Tables) 54
7. Appendix G (Data dictionary) 55
8. Appendix H (Code Listing) 58
1. ABSTRACT

This project consists of the analysis, design and implementation of a data-analysis tool for environmental data for the coast of the Gulf of Mexico which can be accessed on Internet at URL “http://robin.tamu.edu:8891/apps/gondra.security.securitypage”. This system consists of a database to maintain a record of environmental parameters such as salinity, turbidity, temperature, pH, nickel concentration, etc., observed at different stations in the Gulf of Mexico coastal environment. It provides an interface for generic database manipulations. This system allows the user to analyze the data using graphical charts to establish correlations between different parameters so the user can establish possible causes for the relationships.
2. Background and Rationale

Terminos lagoon, located on the southeast coast of the Gulf of Mexico in Mexico, is ecologically and economically important because of its biological diversity and intensive recreational and fishing activities. In recent years, the lagoon has been subjected to high levels of contamination because of petroleum development and exploration, industrial effluent discharge, agricultural inputs and urban inputs. The biogeochemical processes associated with the lagoon can alter the characteristics of these contaminants, thus making it toxic to aquatic organisms. Heavy metals are also of concern. Some metals are required for growth of aquatic life, while others are highly toxic in trace amounts. Therefore, the need for environmental information to help protect marine resources is vital.

Relatively few studies have been conducted to gather data and determine the contamination levels along the coast of the Gulf of Mexico. A research group at the Texas A&M University – Corpus Christi (TAMU-CC) chemistry department consisting of Dr. Virender K Sharma, G.F. Vasquez and others are studying the hydrological parameters (such as salinity, temperature, turbidity, pH, dissolved oxygen, nitrite, etc..) and concentrations of total metals (such as Cd, Cu, Ni, Zn, Fe, etc..) at 24 stations in the Terminos lagoon. This study involves analyzing water samples gathered from different lagoons in the Gulf of Mexico. Every monsoon season they collect 20 samples each week from 15 different stations around the coast of Gulf of Mexico. There is a large amount of data generated due to continuous monitoring of these parameters. The researchers draw graphs using regression analysis to establish correlations between various parameters. From the analysis they infer the possible causes of these relationships.
Proper recording of data and storing it manually is a very time consuming procedure. Presently they are storing this data on paper sample sheets and analysing it by manually drawing charts. As the amount of data increases it is very difficult to keep track of when the data was generated and where it was recorded. Also, extracting data and drawing the graphs manually is a slow and error-prone process. Also this data is not available for all to view and analyze.

A computer-based database system is a very efficient way of recording and storing this type of data so it can be easily stored and modified. The present system draws the graphs so that user can compare the parameters visually, thus saving time. The data will be dynamic and is viewed and analyzed by more people than the person maintaining the database. It is available to all researchers and the general public.

This project involved designing and developing a data-analysis tool on the World Wide Web with a relational database system as the backend. This system allows authorized users to enter, modify and delete data, whereas other users can only view the data. It shows the comparison between parameters graphically using regression analysis depending on user requests. It establishes correlation between different parameters. This project serves the purpose of providing general awareness about contamination of lagoons to clients and the general public as well as government agencies by making data available on World Wide Web. This tool involves basic computer science concepts like programming, database design, graphical user interface and the Internet.
3. Narrative

The primary goal of the project was to provide a graphical presentation of a data-analysis system on the World Wide Web. This tool is developed with user-friendly features. This user-friendly, interactive tool provides immediate up-to-date information about environmental parameters to its users. This tool has two major components: a Database component and a Graphics component. The system also provides an Online User guide.

3.1 Database Component

This component consists of a relational database with PL/SQL procedures. This data-analysis system allows the users to do general database manipulations. The data related to environmental parameters is in the form of relational tables. There are four relational tables: Enviproperties, Metals, Stations, and Passwords. The Enviproperties table has data about chemical process properties (Such as Oxygen, pH, Temperature etc..) at different stations. The Metals table consists of data about different Metal concentrations observed at different stations. All the tables related to environmental parameters has data observed during various samplings. The Stations table has information about various sampling stations and their stationid's. The Passwords table has information about authorized-user passwords and their userid's.

The most important aspect of the system is the interaction between the database and the World Wide Web. PL/SQL procedures along with the Oracle Web Server toolkit are used as the Common Gateway Interface (CGI) between the database and the Internet. The PL/SQL procedures serve the following functions:

1. provide security to the database,
2. provide a graphical user interface,
3. provide an easy-to-use and friendly environment.

4. generate reports and forms.

### 3.1.1 Web Interface

Access to the data through World Wide Web utilizes 3 HTML pages:

1. Security page
2. Main page
3. Database Manipulations page

An overview of the system design and operational flow diagram is shown in Appendix D.

#### 1. Security page

When a user attempts to enter the URL of any page in the system for the first time, he or she will be directed to a security check page, where he/she is prompted to enter his/her userID and password (see Figure E.1). This security is necessary to avoid unauthorized database manipulation by unreliable people. The program administrator assigns the userID and passwords to the certified users. This allows a user to enter as a guest, but he/she will be restricted to do only certain operations like viewing the charts.

#### 2. Main page

When a user passes the security check, he/she will get this page where they are provided with options to link to database manipulations or the chart section (see Figure E.3).

#### 3. Database manipulations page

When a user chooses the submit button with the label "Database Manipulations" he will be directed to this page. This page has different options related to database manipulations on the relational tables (see Figure E.4). The user can choose the operation
he would like to perform from the submit buttons provided. According to the user’s access level he/she will be able to do only certain operations. If a user is a guest, he is able to only view information, otherwise he is able to do all of the database operations. The program administrator assigns userIDs and passwords to the certified users of each table so that unauthorised users are not allowed to modify the table.

3.1.2 Data Manipulations

The Database Manipulations page has four submit buttons, of which each takes the user to the corresponding page.

1. Select

When a user selects this button, he/she will be directed to a page which consists of option buttons for each table to be manipulated (see Figure E.5). The user can select any table to view the data in a HTML table. When a user chooses a particular option he will be directed a page which consists of data from the selected table (see Figure E.6). This data is generated dynamically in HTML table format. The user can go to the database operations page if he selects the “go back to database” link, or can go back to mainpage if he selects the “go back to mainpage” link.

2. Add

When a user selects this button, he/she is directed to a page, which consists of option buttons for each table (see Figure E.7). Here the user can add new data to the selected table. But, the user will not have access to the passwords table. Only the program administrator can change or add passwords. When the user chooses the number of rows he wants to insert and selects the table in which he wants to insert, he reaches a page which has a series of textboxes (see Figure E.8). Here user can enter the data into the textboxes. If the user submits this data, it is entered into the corresponding table and he/she is directed to a page which displays the entire table. The user can go back to the database operations page if he/she selects the “go back to database” link.
3. Delete

Here a user can delete any row from the table he/she selected. The user has choice of selecting multiple rows to be deleted (see Figure E.13). When a user selects a check box related to a particular row and chooses the submit button, that row is deleted. The entire data is displayed after deletion in an HTML table. The user can go back to the database operations page if he/she selects the “go back to database” link, else he/she can choose the “go back to mainpage” link which takes him to the mainpage.

4. Update

This page has option buttons for the tables to be selected (see Figure E.9). When a user chooses a particular table and submits it for update, he/she gets a new page with checkboxes for each row of the selected table along with the data. Here the user has options to select multiple rows (see Figure E.10). After a user selects a particular row he/she is presented with the data present in that particular row, as well as textboxes to enter the data to be updated (see Figure E.11). When the user enters the data in the textboxes and submits it, he is directed to a page which will display the data contained in that table after the update. The user can go back to the database operations page if he/she selects the “go back to database” link, else he/she can choose the “go back to mainpage” link which takes him to the mainpage.

Each of the above pages have a link to the User Guide page, so that whenever a user has some problem he/she can go to the help pages.

3.2 Graphical Presentation Component

This component consists of pages which guide the user to different input forms until the data is presented graphically and data analysis is completed. This interface consists of various options like pull down menus and option buttons to allow the user to
select different attribute pairs to assign to the co-ordinate axes to monitor the relationship between them. This tool provides one chart type, the line chart, to present the data according to the user requests. It presents the data as a linear regression line chart or just a plain line chart.

1. Chart Section page:

This form is reached when a user (authorized or guest) enters the main page and selects the chart-section option. Here a user can select an option and submit it which takes him to the chart page, or he can select the “go back to mainpage” link which takes him/her to the mainpage.

2. Chart page:

This page has options for the user to select a graph type he wants displayed. When the user selects the graph type and submits it, it takes the user to the Line Chart page, or the user can go back to the mainpage by selecting the “go back to mainpage” link.

3. Line chart page:

This page consists of different line chart option buttons like a linear regression chart or a simple chart. When the user selects an option and clicks the submit button, it takes the user to the Attributes page (see Figure E.19), or he/she can choose the “go back to mainpage” link to go to the mainpage.

4. Attributes page: Here the user is provided with option menus where he/she can select x-axis and y-axis attributes (see Figure E.19) and correspondingly view the charts when he/she submits them. When a user wants to go back to the mainpage he can choose the “go back to mainpage” link.
Since a user generally is a researcher, he may want a chart to view how parameters are related. In order to accomplish this task, the plain line chart as well as linear regression chart are provided. Charts generated are two dimensional charts with x and y axes. A line chart is provided so that the user can view the trends in data with respect to time or with some other parameter.

3.3 User Guide

The On-line user guide consists of simple step-by-step instructions to the user as to how to use the software. It has two parts, Database Help and Chart Help.

3.3.1 Database Help

This section explains to the users how to achieve a particular task related to the database. It gives accurate instructions to the user as to how to do a particular operation on the data. It gives details about which options a user needs to take him from page to page to perform a given operation. This section reminds the user which operations he/she can do, and which operations he is not authorized to do. After the user has browsed the database help section, he can choose the “go back” link to go to the previous page, or he can choose the “go back to database” link which will take him to database manipulations page.

3.3.2 Chart Help

This section explains to the users how to draw a chart between parameters requested. It consists of details about which options a user has to navigate from page to page to get desired results. After the user has browsed the database help section, he can choose the “go back” link to go to the previous page, or he can choose the “go back to chart section” link which will take him to the chart section page.
4. Project Environment

This Data-analysis system must run on a Java capable browser like Netscape Navigator, Microsoft Internet Explorer 3.0, or Sun's Hotjava browser. All the data is in the relational database Oracle 8.0. All the coding is done in PL/SQL, with HTML and Java used to generate reports and charts. Oracle Web Server is used to host all the code. CGI script is generated from PL/SQL using the Oracle Web Server tool kit.

Java applets are used to display the dynamic data graphically. The Hypertext Markup Language is used to generate reports. Common Gateway Interface is used to access the Oracle database. The minimum hardware configurations of the client side computer are 16MBRAM, 800MB Hard Drive, CPU clock cycle of 133MHZ whereas server requires atleast 64MBRAM and 12GB Hard Drive.
5. Procedure

The general description of the components that are used in this project are explained in the narrative section. This section explains how the project is analyzed and designed. The detailed description of the various components are also explained.

The following steps were taken before the design of the data-analysis tool for the environmental parameters of the Gulf of Mexico.

1. Interviewed Dr. Sharma, Associate Professor of Chemistry at TAMU-CC, and Dr. Vasquez, Associate Professor of New Mexico University, to determine the requirements of the project.

2. Interviewed the above mentioned professors numerous times to get the final view of what the data-analysis tool would be, what it would do and how it would deal with sample data.

3. Researched the various databases and software, which can be used to keep the database online. Also studied the security issues for the different web supporting software.

4. Researched the different programming languages and the chart-making software that satisfies the needs of the data-analysis tool.

5. Optimized the cost of the software used by evaluating the cost of the different software available.
6. Designed the database system to keep track of sample data, users and to do database manipulations and chart supporting.

The final design of the data-analysis tool is explained in this section.

As described in the previous section, this tool has two major components: Database and Charts.

The Database component mainly deals with database manipulations, queries, forms and reports. This is implemented using the Oracle Web Server toolkit, CGI scripts and PL/SQL procedures. The built-in procedures of PL/SQL are also used for various functions. These procedures made the tool user-friendly and provide security to the system. The detailed description is discussed in next sections.

The Charts component consists of java applets. They are customised by PL/SQL procedures which will receive the parameters requested by the users.

5.1 Oracle Web Server:

The Oracle Web Server is an HTTP server with a tightly integrated Oracle 8.0 Server that enables the creation of distributed and dynamic applications. It creates dynamic HTML documents from the data stored in the Oracle database. When the data changes, the HTML documents are updated automatically with no further effort on the part of the site administrator. It provides a framework that encompasses a modular distributed architecture, an open API that enables you to create portable applications, and different application models or paradigms.

The Oracle Web Server and its toolkit are the main software products that are used by the data-analysis tool. It consists of the following components.
**Oracle Web Listener:**

The Oracle Web Listener receives requests from users using any Web browser. Upon receiving the request from the client, Oracle Web Listener first determines whether the request is for a static document or a dynamic document. If the request is for a static document, the Web Listener sends the file and associated type information directly to the client. If the request is for a dynamic document, it is created dynamically by a program invoked by the Web Listener, in compliance with the Common Gateway Interface (CGI). CGI is an interface that enables HTTP servers to run a program and use the output of that program in the document that is sent to the user. The Web Listener uses it to invoke the Oracle Web Agent when a database procedure is requested.

**Oracle Web Agent:**

The Oracle Web Agent handles requests from users for dynamic pages. It handles the connection to the ORACLE 8 Server, invocation of requested procedures, and transmission of the resulting HTML document back to the Web browser.

**Oracle Web Server Developers Toolkit:**

The Oracle Web Server Developers toolkit is set of PL/SQL routines and procedures to help clients create dynamic HTML documents. It contains the procedures that can execute client requests. The graphical description of Oracle Web Server is shown in Appendix A.

The following events occur when the Web Application Server receives a request.

1. The Listener component of the Web Application Server receives the request from
a client, and determines who should handle it. In this case, it forwards the request to the Web Request Broker (WRB), since the request is for a cartridge.

2. The WRB routes the request to an available PL/SQL cartridge.

3. The PL/SQL cartridge retrieves the name of the PL/SQL Agent from the request, and uses the agent's configuration values to determine to which database server to connect and how to set up the PL/SQL client configuration. You can define many PL/SQL Agents, each with different configuration information.

4. Using the PL/SQL Agent's configuration values, the PL/SQL cartridge connects to the database, prepares the call parameters, and invokes the procedure in the database.

5. The procedure generates the HTML page, which can include dynamic data accessed from tables in the database as well as static data.

6. The output from the procedure is returned via the response buffer back to the PL/SQL cartridge and the client.

When connecting to a database, the PL/SQL cartridge uses configuration information from two sources: a PL/SQL Agent and a Database Access Descriptor (DAD). Like the PL/SQL Agent, a DAD is a named set of configuration values used for database access. Each PL/SQL Agent is associated with a DAD. Appendix A explains how the Data analysis tool works in conjunction with the Oracle Web Server.

A DAD specifies information such as the database name or the SQL*Net V2 service name, the ORACLE_HOME directory, and NLS configuration information such as language, sort type, and date language. You can also specify username and password
information in a DAD; if they are not specified, the user will be prompted to enter a username and password when the URL is invoked.

The connection information is divided into PL/SQL Agents and DADs so that multiple agents can use the same DAD. This enables one to define a DAD for each database to which they want to connect, since it is the DAD that specifies the database. The only difference is in the configuration of the PL/SQL Agent.

5.2 Database Tables:

The database tables that are used and the relationships between them are explained in this section. There are four tables:

1. Metals
2. Enviproperties
3. Station
4. Passwords

The Metals table has the data about metal concentrations (like lead, vanadium, copper, iron, manganese). These concentration units depend on the type of metal that is stored. Users are provided with a Help Guide to know which concentration is expressed and in what units. It has station number, sample, sample location (deep or surface water) and metals as its columns. This table has a many-to-one relationship with the station table.

The Station table has the data about the different monitoring stations which are observed by the Chemistry Department. It has the station name and station number as its columns. Stationno is the primary key for this table.

The Enviproperties table has the data about the different physical and environmental properties of ocean water as well as some important chemical
compositions. Concentrations are expressed in standard units depending on the type of chemical.

It has data obtained for each station, depth of water, type of water, etc... This table is a many-to-one relationship with the station table.

The **Passwords** table has information about the authorised users of the system. Userid is the primary key for this table. It has password, user name and office as its fields. Attributes of all these tables is described in Appendix B.

After careful analysis of the project about the different columns that the database tables should have, the following columns have been decided upon.

**STATION**:

<table>
<thead>
<tr>
<th>Stationno</th>
<th>stationid primary key</th>
<th>not null char(5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stationname</td>
<td>station name</td>
<td>varchar2(50)</td>
</tr>
</tbody>
</table>

**METALS**:

<table>
<thead>
<tr>
<th>stationno</th>
<th>stationid (foreign key)</th>
<th>char(5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>sample</td>
<td>sample name</td>
<td>varchar2(18)</td>
</tr>
<tr>
<td>takenin</td>
<td>where it was taken</td>
<td>varchar2(30)</td>
</tr>
<tr>
<td>sampledate</td>
<td>sample (duration)</td>
<td>varchar2(30)</td>
</tr>
<tr>
<td>depth</td>
<td>depth of ocean water</td>
<td>float(126)</td>
</tr>
<tr>
<td>zinc</td>
<td>zinc concentration</td>
<td>float(126)</td>
</tr>
<tr>
<td>cadmium</td>
<td>cadmium conc.</td>
<td>float(126)</td>
</tr>
<tr>
<td>nickel</td>
<td>nickel conc.</td>
<td>float(126)</td>
</tr>
<tr>
<td>iron</td>
<td>iron conc.</td>
<td>float(126)</td>
</tr>
<tr>
<td>copper</td>
<td>copper conc.</td>
<td>float(126)</td>
</tr>
<tr>
<td>manganese</td>
<td>manganese conc.</td>
<td>float(126)</td>
</tr>
<tr>
<td>chromium</td>
<td>chromium conc.</td>
<td>float(126)</td>
</tr>
<tr>
<td>cobalt</td>
<td>cobalt conc.</td>
<td>float(126)</td>
</tr>
<tr>
<td>lead</td>
<td>lead conc.</td>
<td>float(126)</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
<td>Type</td>
</tr>
<tr>
<td>------------------</td>
<td>--------------------------------------</td>
<td>------------</td>
</tr>
<tr>
<td>vanadium</td>
<td>vanadium conc.,</td>
<td>float(126)</td>
</tr>
<tr>
<td>zirconium</td>
<td>zirconium conc.,</td>
<td>float(126)</td>
</tr>
<tr>
<td>strontium</td>
<td>strontium conc.,</td>
<td>float(126)</td>
</tr>
<tr>
<td>barium</td>
<td>barium conc.,</td>
<td>float(126)</td>
</tr>
<tr>
<td>organicmatter</td>
<td>organic matter conc.,</td>
<td>float(126)</td>
</tr>
</tbody>
</table>

**ENVI Properties:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>stationno</td>
<td>stationid (foreign key)</td>
<td>char(5)</td>
</tr>
<tr>
<td>sample</td>
<td>sample name</td>
<td>varchar2(18)</td>
</tr>
<tr>
<td>type</td>
<td>where it was taken</td>
<td>varchar2(30)</td>
</tr>
<tr>
<td>sampledate</td>
<td>sample date duration</td>
<td>varchar2(21)</td>
</tr>
<tr>
<td>depth</td>
<td>depth of ocean water</td>
<td>float(126)</td>
</tr>
<tr>
<td>temp</td>
<td>temperature of water</td>
<td>float(126)</td>
</tr>
<tr>
<td>salinity</td>
<td>salinity</td>
<td>float(126)</td>
</tr>
<tr>
<td>turbidity</td>
<td>turbidity</td>
<td>float(126)</td>
</tr>
<tr>
<td>ph</td>
<td>ph of water</td>
<td>float(126)</td>
</tr>
<tr>
<td>oxygen</td>
<td>oxygen % in water</td>
<td>float(126)</td>
</tr>
<tr>
<td>soxigen</td>
<td>saturated oxygen%</td>
<td>float(126)</td>
</tr>
<tr>
<td>silicate</td>
<td>silicate conc.,</td>
<td>float(126)</td>
</tr>
<tr>
<td>nitrate</td>
<td>nitrate conc.,</td>
<td>float(126)</td>
</tr>
<tr>
<td>nitrite</td>
<td>nitrite</td>
<td>float(126)</td>
</tr>
<tr>
<td>ammonia</td>
<td>ammonia conc.,</td>
<td>float(126)</td>
</tr>
<tr>
<td>phosphate</td>
<td>phosphate conc.,</td>
<td>float(126)</td>
</tr>
<tr>
<td>totalcarbono2</td>
<td>carbondioxide conc.,</td>
<td>float(126)</td>
</tr>
</tbody>
</table>

**Passwords:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>userid</td>
<td>userid of the user (primary key)</td>
<td>not null char(10)</td>
</tr>
<tr>
<td>password</td>
<td>passwords of user</td>
<td>not null varchar2(10)</td>
</tr>
<tr>
<td>lastname</td>
<td>last name of user</td>
<td>varchar2(18)</td>
</tr>
<tr>
<td>firstname</td>
<td>first name of user</td>
<td>varchar2(18)</td>
</tr>
<tr>
<td>office</td>
<td>office name</td>
<td>varchar2(18)</td>
</tr>
</tbody>
</table>
**Packages, procedures and functions:**

The data-analysis system has PL/SQL procedures, PL/SQL functions and Java code for applets. This section discusses about these procedures.

**Package: security**

This package has all the necessary procedures and functions for checking the users authenticity as well as for prompting and validating.

**Securitypage**

**Function-type: PL/SQL procedure.**

**Input:** When the user first enters the URL of the project site on the Internet browser, he will be calling this function. The input is null in the above case or the target procedure name, if the user enters the target procedure name for the first time on the web without entering the password.

**Function:** It generates a form with the userid textbox and the password textbox, so that for security purposes only authorised users can enter the site.

**Output:** Userid and password request form.

**Isuser**

**Function-type: PL/SQL function.**

**Input:** Userid and password entered in the security page.

**Function:** It does search on the passwords table for the userid and password input parameters, if they exist it returns true else false.

**Output:** An integer value 1(true) or 0(false).
**Isadmin**

Function-type: PL/SQL function.

**Input:** Userid and password entered in the security page.

**Function:** It does search on the passwords table for the userid and password input parameters, if they exist it will return true, else false.

**Output:** An integer value 1(true) or 0(false).

**Isguest**

Function-type: PL/SQL function.

**Input:** Userid and password entered in the security page.

**Function:** It does search on the passwords table for the userid and password input parameters, if they exist it will return true else false.

**Output:** An integer value 1(true) or 0(false).

**Insage**

Function-type: PL/SQL procedure.

**Input:** Userid (as hidden field), Password (as hidden field) and Target Procedure.

**Function:** Depending on the type of user, it gives instructions to the user as to what privileges the user has on the database manipulations and on viewing charts.

**Output:** A form explaining the privileges, with a submit button to continue.
Package : peg

This package deals with data manipulations and queries. It aids in generating forms and reports. All the procedures and functions are coded in the PL/SQL language. It is of much help for constructing complex queries and printing data. Some of the procedures use the built-in procedures of Oracle Web Server tool kit.

Proj

Function type: PL/SQL procedure.

Input: When the user enters the URL of the project site on the Internet Browser he/she will be prompted to enter the password. So the userid and password are the input parameters.

Function: The function of this procedure is to check whether the user is an authorised user by calling isuser, isadmin and isguest functions. Then, if he/she is authorised, they are provided with two radio buttons, chart and database, to select the database-manipulations page or chart-selection page.

Output: A form with two radio buttons and a submit button.

Mainpage

Function-type: PL/SQL procedure.

Input: User selected option (chart or database from proj procedure). Userid and password (hidden form from security procedure).

Function: The main function of this procedure is to generate a form based on the User’s input. If the user selects the database option, this procedure generates a list of submit buttons for inserting, deleting, updating, customised selection and data querying. If he/she selects a chart option, this procedure accesses the database based on the built-in PL/SQL procedures, generates dynamic attributes (x and y co-ordinates) from parameters data. To generate a location from the pull-down menu there is a cursor which selects the monitoring site from the station table. It also generates option buttons for selecting the type of chart the user wants.

Output: Appropriate request form based on database or chart option.
Options

Function-type: PL/SQL procedure.

Input: Userid, password (in hidden form) and the submit variable. The submit variable may be ‘SELECT’ or ‘INSERT’ or ‘UPDATE’ or ‘DELETE’ or ‘CUSTOMISED_SELECT’ or ‘DATAQUERY’.

Function: This procedure based on the options from mainpage, generates a form. For the ‘select’ option it prints a HTML table with option buttons for each table in the database. In the case of ‘insert’ option, it prints the HTML table as well as a pull-down menu for selecting the number of rows the user wants to insert into the database. If the user selects the ‘data-query’ option, this procedure generates a query form requesting the user to select the appropriate fields which the user is interested in examining at the data. The CUSTOMISES SELECT option generates a text box for the user to enter the SQL select statements.

Output: Appropriate form.

SEARCH

Function-type: PL/SQL procedure.

Input: The table name from the options procedure.

Function: To print all the data in the table in an HTML table format.

Output: A form with all the rows of the table in an HTML table format.

INSI

Function-type: PL/SQL procedure.

Input: The table name from the options procedure and the variable containing the number of rows the user wants to insert.

Function: To generate rows of textboxes for the selected table so that the user can
enter the data into those textboxes and submit it using submit button.

**Output:** A form with of rows textboxes for a particular table.

### UPDATE

**Function-type:** PL/SQL procedure.

**Input:** The Userid, password and table name from the pull-down menu of the Options procedure and the option variable which specifies what type of update the user wants (customised or row-by-row update).

**Function:** This procedure generates a check box for each row of the table the user selected along with the original data. In the case of the user selecting customised update it generates a textbox. The User can enter the SQL update statements in the textbox and submit the textbox.

**Output:** An HTML table with check boxes for each row or just a textbox.

### DELROW

**Function-type:** PL/SQL procedure.

**Input:** The Userid, password and table name from the Options procedure.

**Function:** This procedure generates a check box for each row of the table the user selected along with the original data. The user can select multiple rows for deletion from the database.

**Output:** An HTML table with check boxes for each row of the selected table.

### QUERY

**Function-type:** PL/SQL procedure.

**Input:** The Userid and password.

Stationame, from the pull-down menu in the Options procedure.
Columns for each of the parameters and the metalscone table.

**Function:** This function queries the database for the options the user selects in the Options procedure. These options might be the station name, table names, the columns from the table. It builds the queries from the dynamic data selected by the user, executes them and prints the data.

**Output:** It prints the data in the form of a HTML table.

**INSERROWST OR INSERROWP OR INSERROW**

**Function-type:** PL/SQL procedure.

**Input:** The Userid, password, table name number of rows to be inserted and an array of each column value for all the rows inserted.

**Function:** This function uses some built-in packages as well as loops for inserting data into the database and for generating reports.

**Output:** It prints the data of the inserted table in an HTML table format.

**Updemp**

**Function-type:** PL/SQL procedure.

**Input:** The Userid, password, table name and an array of rowids which the user has checked for update in Upd procedure.

**Function:** Since the user has already selected the rows to be updated, this procedure then prints textboxes for each of the rows along with the original data in the row above it.

**Output:** It generates a form with textboxes for the user to update the data corresponding to original column.

**Updrows, Updrowp and Updrowm**

**Function-type:** PL/SQL procedure.

**Input:** The Userid, password, an array of rowids which the user has
checked for update in upd procedure and an array of column values for each column.

**Function**: This procedure updates the column values which was changed in the Updemp procedure and prints the data of the particular table.

**Output**: Generates a form containing the data for the table which was updated.

---

**Delid**

**Function-type**: PL/SQL procedure.

**Input**: The Userid, password, an array of rowids which the user has checked for delete in Delrow procedure.

**Function**: This procedure deletes the rows which were checked in the Delrow procedure and prints the data from the table in an HTML table format.

**Output**: Generates a form containing the data for the table whose rows are deleted.

---

**Custupd (customised update)**

**Function-type**: PL/SQL procedure.

**Input**: The Userid, password, tablename and text from the Upd procedure.

**Function**: This procedure executes the text entered into the textbox provided in the Upd procedure form using PL/SQL built-in procedures and prints the data in the form of a table.

**Output**: Generates a form containing the data.

---

**Custselect (customised select)**

**Function-type**: PL/SQL procedure.
**Input:** The Userid, password, table name and text from the textbox containing SQL statements, text from another textbox containing column name and value. These parameters sent by the Options procedure.

**Function:** This procedure executes the text entered into the textbox provided in the Options procedure form using PL/SQL built-in procedures and prints the data in the form of a table. It also takes the table name, column name, column value and constructs SQL select statements, executes them and prints the data.

**Output:** Generates a form containing the data.

---

**Chart package**

This package takes care of all chart-related procedures and functions. This package is separated from the other packages, mainly because it contains java applets and to maintain the design of the software in a modular fashion. And also at any time only a part of the code which is used will be in the memory, thus decreasing the load on the server because of graphics.

It consists of mainly the following procedure.

---

**Chartsect (Chart Section)**

**Function-type:** PL/SQL procedure and java code(applets).

**Input:** The Userid, password, stationcode, x co-ordinate, y co-ordinate, chart type and where taken (surface or deep) from the Mainpage procedure.

**Function:** This procedure builds the queries from the attributes selected from the Mainpage, executes them. The resultant data is stored in the form of strings delimited by comma. These strings will be sent to java applets as parameters. The java applet tokenizes the strings, stores them in vectors, then selects them one-by-one from the vectors and draws the charts.

**Output:** Generates the graphs using java applets, displays the data queried from the database in an easy-to-understand and graphical manner.
6. Results and Future Work

A data-analysis system with user-friendly features has been implemented for the Gulf of Mexico Environmental pollution program. This system can access the Environmental Database through World Wide Web. It presents the environmental data on the internet as well as present the data in a graphical format. The user can navigate through the system web pages by clicking on the submit buttons. This system allows the authorized users to update, insert and delete the database through World Wide Web. It is easy to use and needs little training thus saving a lot of time for the researchers and general public.
7. References


Appendix A

This figure explains how the data analysis tool works in conjunction with the Oracle Web Server.
Appendix B

Tables and their Columns

The following are the tables and their attributes which are used in this project.

**STATION:**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stationno</td>
<td>stationid primary key</td>
<td>not null char(5)</td>
</tr>
<tr>
<td>Stationname</td>
<td>station name</td>
<td>varchar2(50)</td>
</tr>
</tbody>
</table>

**METALS:**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>stationno</td>
<td>stationid (foreign key)</td>
<td>char(5)</td>
</tr>
<tr>
<td>sample</td>
<td>sample name</td>
<td>varchar2(18)</td>
</tr>
<tr>
<td>takenin</td>
<td>where it was taken</td>
<td>varchar2(30)</td>
</tr>
<tr>
<td>sampledate</td>
<td>sample (duration)</td>
<td>varchar2(30)</td>
</tr>
<tr>
<td>depth</td>
<td>depth of ocean water</td>
<td>float(126)</td>
</tr>
<tr>
<td>zinc</td>
<td>zinc concentration</td>
<td>float(126)</td>
</tr>
<tr>
<td>cadmium</td>
<td>cadmium conc.</td>
<td>float(126)</td>
</tr>
<tr>
<td>nickel</td>
<td>nickel conc.</td>
<td>float(126)</td>
</tr>
<tr>
<td>iron</td>
<td>iron conc.</td>
<td>float(126)</td>
</tr>
<tr>
<td>copper</td>
<td>copper conc.</td>
<td>float(126)</td>
</tr>
<tr>
<td>manganese</td>
<td>manganese conc.</td>
<td>float(126)</td>
</tr>
<tr>
<td>chromium</td>
<td>chromium conc.</td>
<td>float(126)</td>
</tr>
<tr>
<td>cobalt</td>
<td>cobalt conc.</td>
<td>float(126)</td>
</tr>
<tr>
<td>lead</td>
<td>lead conc.</td>
<td>float(126)</td>
</tr>
<tr>
<td>vanadium</td>
<td>vanadium conc.</td>
<td>float(126)</td>
</tr>
<tr>
<td>zirconium</td>
<td>zirconium conc.</td>
<td>float(126)</td>
</tr>
<tr>
<td>strontium</td>
<td>strontium conc.</td>
<td>float(126)</td>
</tr>
<tr>
<td>barium</td>
<td>barium conc.</td>
<td>float(126)</td>
</tr>
<tr>
<td>organicmatter</td>
<td>organic matter conc.</td>
<td>float(126)</td>
</tr>
</tbody>
</table>

**ENVIPROPERTIES:**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>stationno</td>
<td>stationid (foreign key)</td>
<td>char(5)</td>
</tr>
<tr>
<td>sample</td>
<td>sample name</td>
<td>varchar2(18)</td>
</tr>
</tbody>
</table>

29
<table>
<thead>
<tr>
<th>type</th>
<th>where it was taken</th>
<th>varchar2(30)</th>
</tr>
</thead>
<tbody>
<tr>
<td>sampledate</td>
<td>sample date duration</td>
<td>varchar2(21)</td>
</tr>
<tr>
<td>depth</td>
<td>depth of ocean water</td>
<td>float(126)</td>
</tr>
<tr>
<td>temp</td>
<td>temperature of water</td>
<td>float(126)</td>
</tr>
<tr>
<td>salinity</td>
<td>salinity</td>
<td>float(126)</td>
</tr>
<tr>
<td>turbidity</td>
<td>turbidity</td>
<td>float(126)</td>
</tr>
<tr>
<td>ph</td>
<td>ph of water</td>
<td>float(126)</td>
</tr>
<tr>
<td>oxygen</td>
<td>oxygen % in water</td>
<td>float(126)</td>
</tr>
<tr>
<td>soxygen</td>
<td>saturated oxygen%</td>
<td>float(126)</td>
</tr>
<tr>
<td>silicate</td>
<td>silicate conc.,</td>
<td>float(126)</td>
</tr>
<tr>
<td>nitrate</td>
<td>nitrate conc.</td>
<td>float(126)</td>
</tr>
<tr>
<td>nitrite</td>
<td>nitrite</td>
<td>float(126)</td>
</tr>
<tr>
<td>ammonia</td>
<td>ammonia conc.,</td>
<td>float(126)</td>
</tr>
<tr>
<td>phosphate</td>
<td>phosphate conc.,</td>
<td>float(126)</td>
</tr>
<tr>
<td>totalcarbono2</td>
<td>carbodioxide conc.,</td>
<td>float(126)</td>
</tr>
</tbody>
</table>

**PASSWORDS:**

<table>
<thead>
<tr>
<th>userid</th>
<th>userid of the user(primaryKey)</th>
<th>not null char(10)</th>
</tr>
</thead>
<tbody>
<tr>
<td>password</td>
<td>passwords of user</td>
<td>not null varchar2(10)</td>
</tr>
<tr>
<td>lastname</td>
<td>last name of user</td>
<td>varchar2(18)</td>
</tr>
<tr>
<td>firstname</td>
<td>first name of user</td>
<td>varchar2(18)</td>
</tr>
<tr>
<td>office</td>
<td>office name</td>
<td>varchar2(18)</td>
</tr>
</tbody>
</table>
Appendix C

ER-Diagram
The figure explains how the procedures and functions are related to user interface.

**APPENDIX D**
The figure explains how the procedures and functions are related to user interface.

**APPENDIX D**
Appendix E

FIGURES OF SAMPLE FORMS AND REPORTS
If you want to enter as a guest, please enter "guest" at both Loginname and Password.

---

Figure E.1
Login page of the data-analysis system.
If you are an administrator then you can do database manipulations to all tables including passwords table as well as view charts.

**Figure E.2**
Form which explains the User's access levels.
Choose the option you want to do on operations on it

- Database Section
- Chart Section

submit

**Figure E.3**
Form giving the user the options to choose from database or chart section.
Figure E.4
Database options.
Figure E.5
Form providing the search options.
All the data in the table ENVIPROPERTIES

Note: All Concentrations are in mg/l (milligrams/Litre).
Temperature is in degrees celsius (°C). Depth in metres. Turbidity in NTU.

Figure E.6
Results of searching based on select all search.
Figure E.7
A form allowing a user to select options for insertion into the database.
Figure E.8

Multiple rows can be inserted as per the number of rows selected in Fig E.7.
Figure E.9
Update Form with select to update and customize update options.
Select the rows to update

<table>
<thead>
<tr>
<th>Check</th>
<th>USERID</th>
<th>PASSWORD</th>
<th>LASTNAME</th>
<th>FIRSTNAME</th>
<th>OFFICE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>admin</td>
<td>CCBAYW</td>
<td>Gondra</td>
<td>Praveen</td>
<td>uscenter</td>
</tr>
<tr>
<td></td>
<td>joe</td>
<td>bonkers</td>
<td>joe</td>
<td>martin</td>
<td>ci400tamucc</td>
</tr>
</tbody>
</table>

update  Reset

Note: You can select multiple rows to update simultaneously

Back to Main  Help

---

**FigureE.10**

Update selection where a user can select multiple rows to update.
Update form with the textboxes to update the corresponding values.
NOTE: Enter any Update SQL statements with no semicolon() at the end and owner of the table before its name. For example: Enter something like `update gondra.metas set inc=10`.

Figure E.12
Customize Update textarea where a user can enter SQL Update statements.
<table>
<thead>
<tr>
<th>ID</th>
<th>Type</th>
<th>Zone</th>
<th>Date</th>
<th>Concentration</th>
</tr>
</thead>
<tbody>
<tr>
<td>s001</td>
<td>MUD</td>
<td>a</td>
<td>3</td>
<td>10</td>
</tr>
<tr>
<td>s001</td>
<td>s</td>
<td>s</td>
<td>3</td>
<td>10</td>
</tr>
<tr>
<td>s001</td>
<td>ss006</td>
<td>deepwater</td>
<td>01/01/1999</td>
<td>3</td>
</tr>
<tr>
<td>s001</td>
<td>ss</td>
<td>ss</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

Note: You can select multiple rows to update simultaneously. All Concentrations are in mg/l (milligrams/Litre).

Temperature is in degrees celsius (°C). Depth in metres. Turbidity in NTU.

Figure E.13
Deleting multiple rows from the Metal Concentrations.
Note:
In case of first option: Enter SQL SELECT statements of the form

```
SELECT * from gondra.metals where zinc=2.0
```

with no semicolon at the end.

In case of second option: Enter table name in first text area (like gondra.metals), enter column name in the second text area (like salinity or zinc) and enter the value on the last text area (like -- 9.0 or -- 9.0 and vanadium=2.0).

FigureE.14
Customized Select Form.
Figure E.15
Data Query page for querying the database.
Figure E.16
Queries based on specific criteria.
Query the Metal Concentrations

Enter the date Here...  Enter and Click...

Enter the date in the format: mm/yy/yyyy.
For Example: 04/08/1999 for April 08 1999.

- Back to Main - Help

Figure E.17
Query based on date.
Query the Metal Concentrations

Enter the Metal name Here... ZINC
Enter the Search Criteria...

Enter the Search Criteria as...
For Example: (< or > or <= or >= or = 30.0

Figure E.18
Query based on specific column name.
Figure E.19
Chart parameters selection Form.
Figure E.20
Line chart along with linear regression chart using java.
### Appendix F
Procedures (Forms & Reports) vs Tables

<table>
<thead>
<tr>
<th>Tablename</th>
<th>Station</th>
<th>Enviproperties</th>
<th>Metals</th>
<th>Passwords</th>
</tr>
</thead>
<tbody>
<tr>
<td>search1</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>S</td>
</tr>
<tr>
<td>options</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>S</td>
</tr>
<tr>
<td>ins1</td>
<td>S, I</td>
<td>S, I</td>
<td>S, I</td>
<td>S, I</td>
</tr>
<tr>
<td>inserwp</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>upd</td>
<td>S, U</td>
<td>S, U</td>
<td>U, S</td>
<td>U, S</td>
</tr>
<tr>
<td>updrowm</td>
<td></td>
<td></td>
<td>U, S</td>
<td></td>
</tr>
<tr>
<td>updrowp</td>
<td></td>
<td></td>
<td>U, S</td>
<td></td>
</tr>
<tr>
<td>delrow</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>S</td>
</tr>
<tr>
<td>delid</td>
<td>S, D</td>
<td>S, D</td>
<td>S, D</td>
<td>S, D</td>
</tr>
<tr>
<td>query</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>S</td>
</tr>
<tr>
<td>mainpage</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>S</td>
</tr>
</tbody>
</table>

I = Insert rows into the table.  
D = Delete the rows from the table.  
U = Update the rows the table.  
S = Select the rows from the table.
APPENDIX G
System Tables and their dependency and independency Description

1) Station :

This table is created to name the stations and code them.

1. Stationno: Station identification number --- Primary key.

2. Stationname: Name of the station where samplings are taken corresponding to the stationno.

2) Enviproperties :

This table contains all the information about different chemical concentrations and the properties of the Gulf of Mexico waters. It is created to describe those parameters.

1. Stationno: Station identification number. --- Foreign key references Station Table.

2. Sample: Sample name, this tells about when the sample was taken, which season etc.

3. Takenin: Type of water, that is whether deep waters or shallow waters.

4. Sampledate: Date when the sample taken.

5. Depth: This field tells you the depth where the sample was taken.

6. Temp: The temperature of the water at the location where the sample was taken.
   It is expressed in degrees celsius.

7. Salinity: Salt content in the water.

8. Turbidity: This is the property of the water expressed in units NTU.

9. Ph: This is the property of the water which tells whether the water is acid of base.

10. Oxygen: This is the dissolved oxygen content in water expressed in mg/l.

11. Soxygen: Saturated oxygen content in water, expressed in mg/l.

12. Silicate: Content of silicate salts in the water in mg/l.
13. Nitrate: Nitrate salts content in the water samples in mg/l.
14. Nitrite: Nitrite salts content in the water samples in mg/l.
15. Ammonia: Gas Concentration of Ammonia in water in mg/l.
16. Phosphate: Salts of phosphate in ocean water samples expressed in mg/l.
17. Total Carbon O2: Concentration of Carbon-Dioxide in samples expressed in mg/l.

3) Metals Table:

This table relates Metals concentrations with the Station table.

1. Station no: Station identification number. ---Foreign key references Station Table.
2. Sample: Sample name, this tells about when the sample was taken, which season etc.,
3. Taken in: Type of water, that is whether deep waters or shallow waters.
4. Sample date: Date when the sample taken.
5. Depth: This field tells you the depth where the sample was taken.
6. Zinc: A sort metal concentration of zinc in natural ocean water samples expressed mg/l (Milligram of the metal per litre of the solution).
7. Cadmium: Concentration of Cadmium in sample in mg/l.
8. Nickel: Nickel concentration in water samples expressed in milligrams per litre.
9. Iron: Metal concentration of iron in units of mg/l.
10. Copper: Copper metal concentration in ocean water samples.
11. Manganese: Manganese presence in water in standard units.
13. Cobalt: Cobalt in mg/l in samples.
15. Vanadium: Concentration of Vanadium in samples in standard units.

16. Zirconium: Concentration of Zirconium in samples in standard units of mg/l.

17. Strontium: Concentration of Strontium in samples in standard units.


4) **Web User Passwords:**

   This table has information about the system users userid’s and passwords. It also has users last name, first name and office address.

1. **Userid:** This is the user's userid given by the program tool administrator.  
   --- *Primary Key*.

2. **Password:** Passwords of the users assigned to a particular userid by the system Administrator. --- not null

3. **Lastname:** Last name of the authorised tool user.

4. **Firstname:** First name of the web user.

5. **Office:** Address of the user.
APPENDIX H

PROGRAM LISTINGS

Packages Used

PCG ........................................PCG.SQL
( Database manipulations package)

CHART .......................................CHART.SQL
( Preparing the data for chart routines and calling java applets )

SECURITY .................................SECURITY.SQL
( Security for user identification and safe usage of tool )

GRAD code .................................GRAD.JAVA & GRAD1.JAVA
( Java code for taking the data and drawing the charts )

Code URL:
{http://robin.tamucc.edu:8888/owa_dba/owa/browse.user_plsql?cuser=GONDRA}.

System Access URL:
{ http://robin.tamucc.edu:8891/apps/gondra.security.securitypage }. 