The United Nations University

GEOTHERMAL TRAINING PROGRAMME
Orkustofnun, Grensásvegur 9,
IS-108 Reykjavík, Iceland

Reports 2003
Number 9

APPLICATION OF DATABASE IN MONITORING THE PERFORMANCE OF GEOTHERMAL STEAM FIELDS AND POWER PLANTS IN THE PHILIPPINES

Ariel D. Fronda
Geothermal Division, Energy Resource Development Bureau
Department of Energy
Energy Center, Merritt Road
Fort Bonifacio, Metro Manila
PHILIPPINES
adfronda@yahoo.com

ABSTRACT

Data is vital for the efficient development and management of geothermal resources. The Philippines, with an installed capacity of 1931 MWe and still with a great potential for geothermal exploration and development, are generating an increasing volume of data. A computerized geothermal database was designed and developed for the storage, retrieval and manipulation of the collected data. The database will be used and applied by the Philippine Department of Energy (DOE) in monitoring the performance of geothermal steam fields and power plants to avoid over exploitation and assure sustainable geothermal resource management.

Well information, reservoir monitoring parameters, drilling information and power plant generation are the essential data considered for the database development using the Oracle Relational Database Management System (RDBMS). Oracle Developer tools and other related computer software were also utilized to provide fast, simple and easy data input, editing, viewing, reporting and graphing.

1. INTRODUCTION

The Philippine Department of Energy (DOE) is a government agency mandated among others, to administer programmes for exploration, development, transportation, marketing, and appropriate management of energy resources in all forms, including geothermal energy. It regulates and exercises full supervision and control over all government and private energy-related projects to ensure proper exploitation, utilization and sustainable development of the country's indigenous resources.

The Geothermal Division, which is under the Energy Resource Development Bureau of DOE, is mandated among others, to oversee, monitor, and regulate geothermal operations undertaken by both private and government entities to ensure strict compliance to the policies and standards of the government (Requejo, 1996; Salvania, 1995).

It is therefore important that the technical personnel of the Geothermal Division acquire the necessary training to improve their knowledge and skills in competently performing its mandate. A suitable training was provided by the United Nations University Geothermal Training Programme (UNU-GTP) in Iceland to the author to assist the Philippines, particularly the DOE, in accomplishing its functions.

Through this training, a relevant project was undertaken by the author that dealt with the development of a computerized database, which will be applied, in monitoring the performance of geothermal steam fields and power plants in the Philippines.

2. DATABASE DEVELOPMENT

2.1 Objective

The Philippines remain the second largest producer of geothermal energy for electricity generation. As of December 2002, an installed capacity of 1931 MWe in six geothermal production fields (Figure 1) had been reached after about 30 years of geothermal exploration, development and utilization. This is only 43% of the 4,537 MWe total geothermal resource potential of the country, in which the remaining 2,606 MWe or 57% are expected to be tapped in the future (Geothermal Division-DOE, 2002).

Along with these developments and the continuous exploitation of the geothermal resources to attain the country's energy self-reliance, large volumes of data will be produced and accumulated. These also include the existing data that have been generated through the years of geothermal experience. It is therefore very important that DOE should develop a computerized geothermal database that will efficiently store, retrieve and manipulate this information in performing their function effectively.

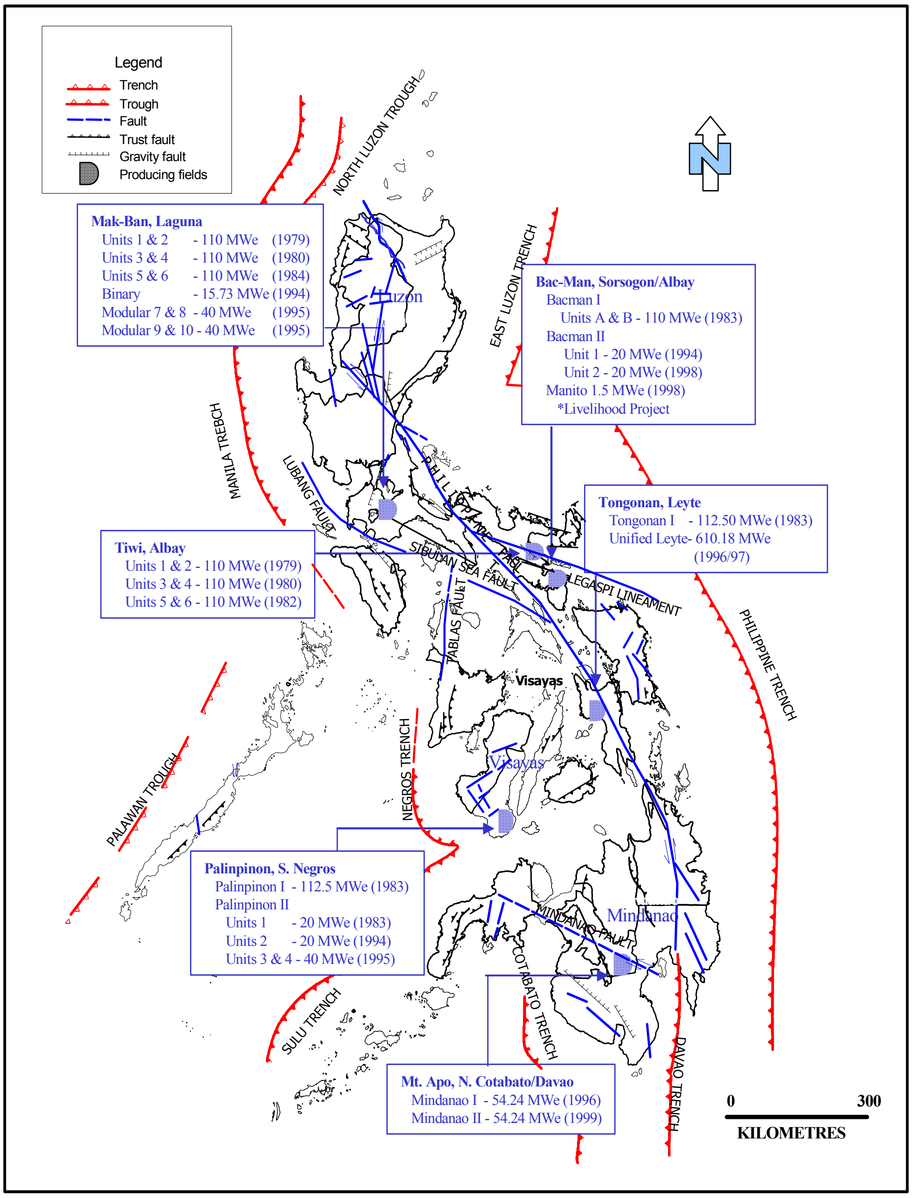


FIGURE 1: Map of the Philippines with the location of the six geothermal production fields

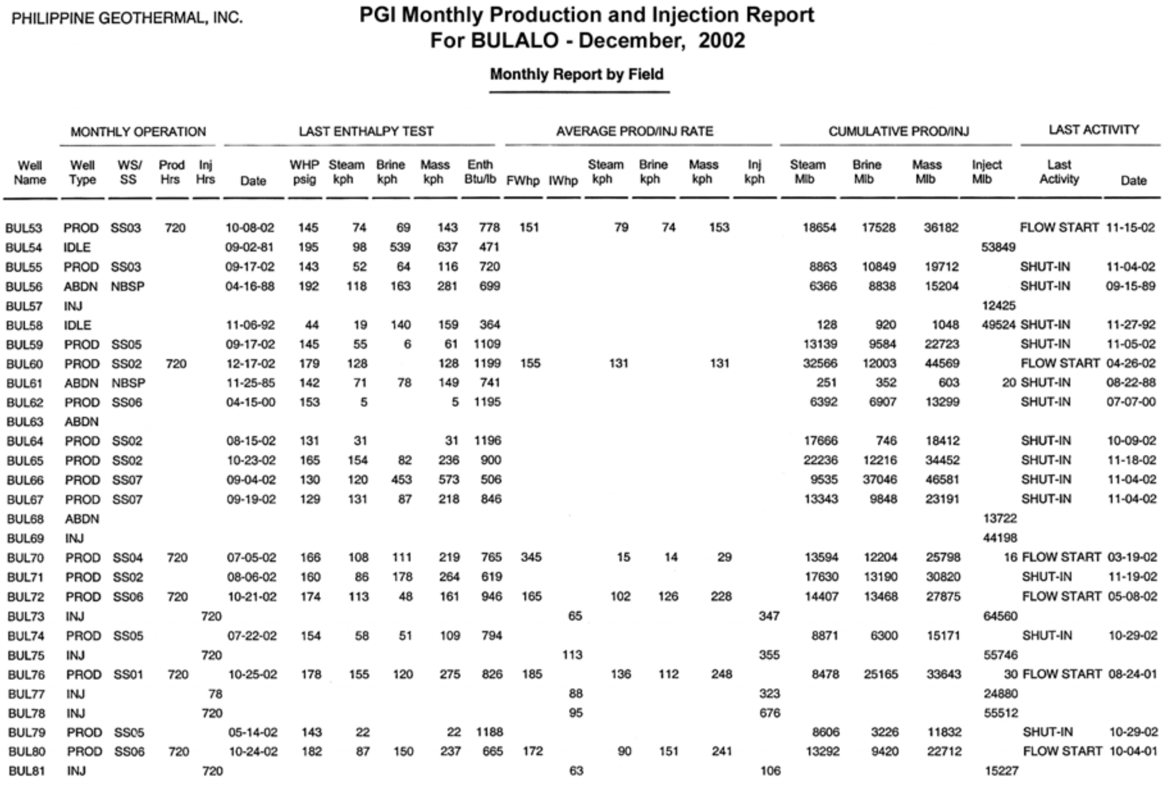
Hence, the aim of this project is to develop a database that can be used as a tool by DOE in monitoring and supervising the resource management strategies of the geothermal industry in the Philippines. The performance of the geothermal steam fields as well as the power plants need to be closely monitored to avoid over-exploitation of the resource. It is expected that this objective will be achieved with the implementation of this computerized geothermal database.

2.2 Data collection

For DOE to fully accomplish its mandated function, it requires submission of reports and information in paper format from the geothermal steam field developers and power plant operators regarding their operations. These include the following:

- Production and injection reports that contains data from wells in production or injection which give information on mass flow rate, steam flow rate, water (brine) flow rate, and injection rate of each well (Figure 2).
- Report on well test activities and downhole measurements (pressure and temperature) conducted (Figure 3).
- Mass and heat extraction and injection report (Figure 4).
- Report on power generation of the power plants.

These data are evaluated, analyzed and used for a separate report prepared by DOE as for monitoring. Eventually, the submitted reports are compiled in a manual filing system.



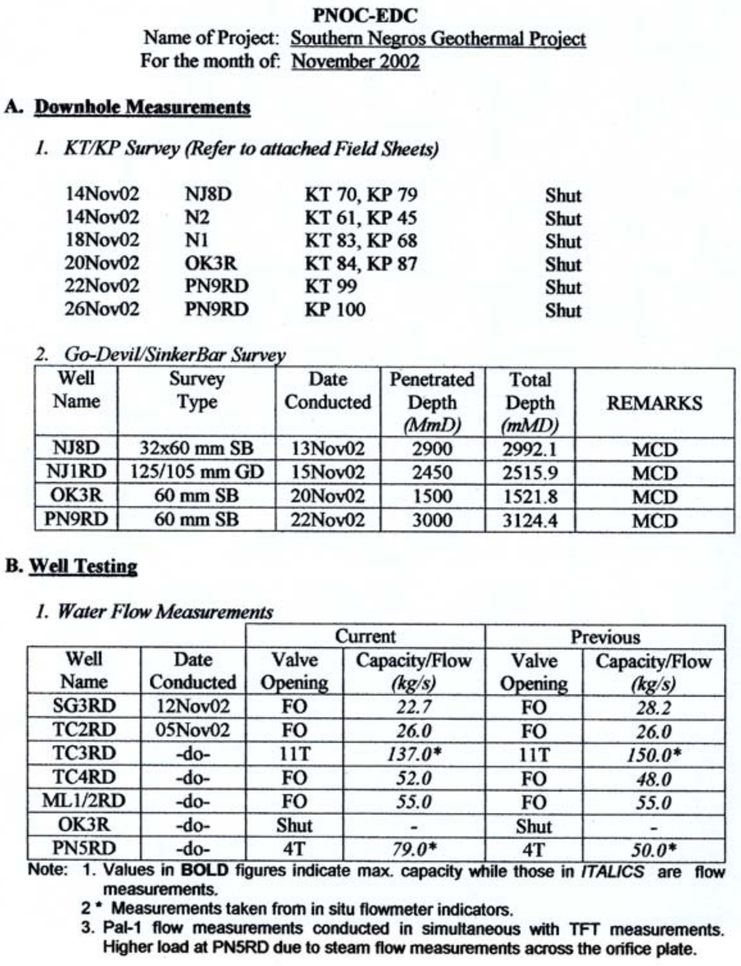
PHILIPPINE GEOTHERMAL, INC.

PGI Monthly Production and Injection Report
For BULALO - December, 2002

Monthly Report by Field

	MONTHLY OPERATION				LAST ENTHALPY TEST						AVERAGE PROD/INJ RATE						CUMULATIVE PROD/INJ				LAST ACTIVITY	
Well Name	Well Type	WS/ SS	Prod Hrs	Inj Hrs	Date	WHP psig	Steam kph	Brine kph	Mass kph	Enth Btu/lb	FWhp	IWhp	Steam kph	Brine kph	Mass kph	Inj kph	Steam Mlb	Brine Mlb	Mass Mlb	Inject Mlb	Last Activity	Date
BUL53	PROD	SS03	720		10-08-02	145	74	69	143	778	151		79	74	153		18654	17528	36182		FLOW START	11-15-02
BUL54	IDLE				09-02-81	195	98	539	637	471										53849		
BUL55	PROD	SS03			09-17-02	143	52	64	116	720							8863	10849	19712		SHUT-IN	11-04-02
BUL56	ABDN	NBSP			04-16-88	192	118	163	281	699							6366	8838	15204		SHUT-IN	09-15-89
BUL57	INJ																			12425		
BUL58	IDLE				11-06-92	44	19	140	159	364							128	920	1048	49524	SHUT-IN	11-27-92
BUL59	PROD	SS05			09-17-02	145	55	6	61	1109							13139	9584	22723		SHUT-IN	11-05-02
BUL60	PROD	SS02	720		12-17-02	179	128		128	1199	155		131		131		32566	12003	44569		FLOW START	04-26-02
BUL61	ABDN	NBSP			11-25-85	142	71	78	149	741							251	352	603	20	SHUT-IN	08-22-88
BUL62	PROD	SS06			04-15-00	153	5		5	1195							6392	6907	13299		SHUT-IN	07-07-00
BUL63	ABDN																					
BUL64	PROD	SS02			08-15-02	131	31		31	1196							17666	746	18412		SHUT-IN	10-09-02
BUL65	PROD	SS02			10-23-02	165	154	82	236	900							22236	12216	34452		SHUT-IN	11-18-02
BUL66	PROD	SS07			09-04-02	130	120	453	573	506							9535	37046	46581		SHUT-IN	11-04-02
BUL67	PROD	SS07			09-19-02	129	131	87	218	846							13343	9848	23191		SHUT-IN	11-04-02
BUL68	ABDN																			13722		
BUL69	INJ																			44198		
BUL70	PROD	SS04	720		07-05-02	166	108	111	219	765	345		15	14	29		13594	12204	25798	16	FLOW START	03-19-02
BUL71	PROD	SS02			08-06-02	160	86	178	264	619							17630	13190	30820		SHUT-IN	11-19-02
BUL72	PROD	SS06	720		10-21-02	174	113	48	161	946	165		102	126	228		14407	13468	27875		FLOW START	05-08-02
BUL73	INJ			720								65				347				64560		
BUL74	PROD	SS05			07-22-02	154	58	51	109	794							8871	6300	15171		SHUT-IN	10-29-02
BUL75	INJ			720								113				355				55746		
BUL76	PROD	SS01	720		10-25-02	178	155	120	275	826	185		136	112	248		8478	25165	33643	30	FLOW START	08-24-01
BUL77	INJ			78								88				323				24880		
BUL78	INJ			720								95				676				55512		
BUL79	PROD	SS05			05-14-02	143	22		22	1188							8606	3226	11832		SHUT-IN	10-29-02
BUL80	PROD	SS06	720		10-24-02	182	87	150	237	665	172		90	151	241		13292	9420	22712		FLOW START	10-04-01
BUL81	INJ			720								63				106				15227		

FIGURE 2: An example of a production and injection report



PNOC-EDC
Name of Project: Southern Negros Geothermal Project
For the month of: November 2002

A. Downhole Measurements

1. KT/KP Survey (Refer to attached Field Sheets)

14Nov02	NJ8D	KT 70, KP 79	Shut
14Nov02	N2	KT 61, KP 45	Shut
18Nov02	N1	KT 83, KP 68	Shut
20Nov02	OK3R	KT 84, KP 87	Shut
22Nov02	PN9RD	KT 99	Shut
26Nov02	PN9RD	KP 100	Shut

2. Go-Devil/SinkerBar Survey

Well Name	Survey Type	Date Conducted	Penetrated Depth *(MmD)*	Total Depth *(mMD)*	REMARKS
NJ8D	32x60 mm SB	13Nov02	2900	2992.1	MCD
NJ1RD	125/105 mm GD	15Nov02	2450	2515.9	MCD
OK3R	60 mm SB	20Nov02	1500	1521.8	MCD
PN9RD	60 mm SB	22Nov02	3000	3124.4	MCD

B. Well Testing

1. Water Flow Measurements

		Current		Previous	
Well Name	Date Conducted	Valve Opening	Capacity/Flow *(kg/s)*	Valve Opening	Capacity/Flow *(kg/s)*
SG3RD	12Nov02	FO	*22.7*	FO	*28.2*
TC2RD	05Nov02	FO	*26.0*	FO	*26.0*
TC3RD	-do-	11T	*137.0**	11T	*150.0**
TC4RD	-do-	FO	*52.0*	FO	*48.0*
ML1/2RD	-do-	FO	*55.0*	FO	*55.0*
OK3R	-do-	Shut	-	Shut	-
PN5RD	-do-	4T	*79.0**	4T	*50.0**

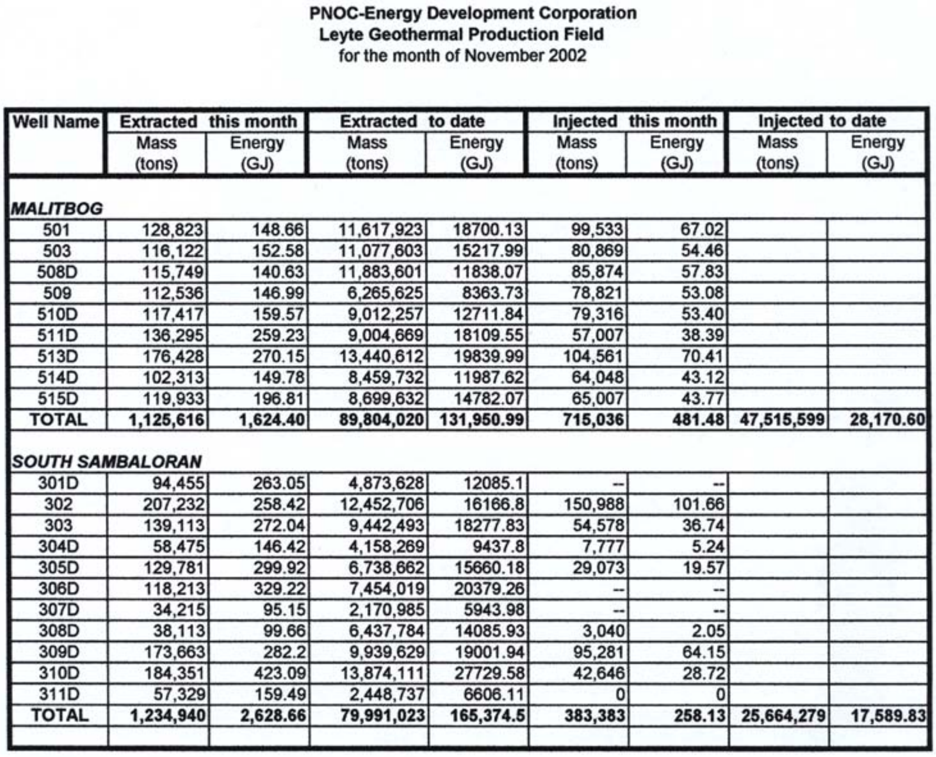
Note: 1. Values in **BOLD** figures indicate max. capacity while those in *ITALICS* are flow measurements.
2 * Measurements taken from in situ flowmeter indicators.
3. Pal-1 flow measurements conducted in simultaneous with TFT measurements. Higher load at PN5RD due to steam flow measurements across the orifice plate.

FIGURE 3: An example of a report showing well test activities and downhole measurements

This is the procedure on how DOE handles data. However, based on experience, if these data are not properly kept or managed, misplacement or even loss of these paper reports may happen because of natural human mistakes. This inefficiency is inherent in a manual filing system. This is basically the reason why the author recognizes the importance of developing a computerized database to avoid the effects of the manual filing system.

2.3 Database design

The emphasis of the database design is on the aspects of geothermal reservoir and resource management. In order to design a reliable and accurate computerized database on this specialized subject, it is an advantage for the designer to have an ample knowledge of geothermal reservoir engineering, and very good experience on designing databases. Although the author has little exposure on geothermal reservoir engineering and database designing,



PNOC-Energy Development Corporation
Leyte Geothermal Production Field
for the month of November 2002

Well Name	Extracted this month		Extracted to date		Injected this month		Injected to date	
	Mass (tons)	Energy (GJ)	Mass (tons)	Energy (GJ)	Mass (tons)	Energy (GJ)	Mass (tons)	Energy (GJ)
MALITBOG								
501	128,823	148.66	11,617,923	18700.13	99,533	67.02		
503	116,122	152.58	11,077,603	15217.99	80,869	54.46		
508D	115,749	140.63	11,883,601	11838.07	85,874	57.83		
509	112,536	146.99	6,265,625	8363.73	78,821	53.08		
510D	117,417	159.57	9,012,257	12711.84	79,316	53.40		
511D	136,295	259.23	9,004,669	18109.55	57,007	38.39		
513D	176,428	270.15	13,440,612	19839.99	104,561	70.41		
514D	102,313	149.78	8,459,732	11987.62	64,048	43.12		
515D	119,933	196.81	8,699,632	14782.07	65,007	43.77		
TOTAL	**1,125,616**	**1,624.40**	**89,804,020**	**131,950.99**	**715,036**	**481.48**	**47,515,599**	**28,170.60**
SOUTH SAMBALORAN								
301D	94,455	263.05	4,873,628	12085.1	--	--		
302	207,232	258.42	12,452,706	16166.8	150,988	101.66		
303	139,113	272.04	9,442,493	18277.83	54,578	36.74		
304D	58,475	146.42	4,158,269	9437.8	7,777	5.24		
305D	129,781	299.92	6,738,662	15660.18	29,073	19.57		
306D	118,213	329.22	7,454,019	20379.26	--	--		
307D	34,215	95.15	2,170,985	5943.98	--	--		
308D	38,113	99.66	6,437,784	14085.93	3,040	2.05		
309D	173,663	282.2	9,939,629	19001.94	95,281	64.15		
310D	184,351	423.09	13,874,111	27729.58	42,646	28.72		
311D	57,329	159.49	2,448,737	6606.11	0	0		
TOTAL	**1,234,940**	**2,628.66**	**79,991,023**	**165,374.5**	**383,383**	**258.13**	**25,664,279**	**17,589.83**

FIGURE 4: An example of a mass and heat extraction and injection report

the experts of Orkustofnun - the National Energy Authority of Iceland through the Geothermal Training Programme provided the required knowledge and experience.

To start with, a database is a structured collection of data. Data in this design refers to the information gathered from paper reports submitted to DOE. Then, an Oracle Relational Database Management System (RDBMS) was considered in this project to provide the structure and mechanism to process the available data. This RDBMS allows access to the stored data to be retrieved and manipulated according to the specific needs of the users for an intended application (Ensor and Stevenson, 1997).

The database developed in this project is made up of tables. Information that describes or illustrates a certain subject was combined and stored in a table. Each table is composed of rows (horizontal) and columns (vertical). Rows are made up of records (Ellison, 1989). Each record contains a data value at the intersection of a row and a column. Different tables were created for the various sets of information. Grouping of information to form a table is considered as a basic requirement for RDBMS but needs careful study, analysis and know-how of the business process; in this case, this refers to familiarity with geothermal operations.

2.3.1 Data modelling

Analyzing DOE's requirements in performing its functions was the first thing that was done in the database design. From this, grouping of information that formed the tables was identified. Below is a brief description of the tables that were created:

Well table describes the characteristics of each geothermal well. It contains information about the type of the well drilled in a specific location of a geothermal steam field.

Field table gives information about a producing geothermal steam field in the Philippines. It indicates the location, coverage area of the steam field, field's installed capacity and the date when the service contract was awarded for the development of the field.

Company table gives information about the company that manages the development of a steam field or the operation of a power plant.

Prod_Inj table stores data that are being monitored by DOE. These are the wellhead pressure, enthalpy, steam flow rate, water flow rate, mass flow rate, injection rate, heat extraction and heat injection of production and injection wells.

Log table describes the logs conducted on a well. Hence, the type of logs and the tools used in the logs are the information in this table.

Tdata table stores the values obtained from the logs according to the date and time that the logs were performed.

Ddata table stores the values gathered from the logs according to the measured depths that the logs were performed.

Drilling table provides information about the drilling activity and the drill rig used. It has the dates when drilling started and ended as well as the starting and ending depth.

Drillact table stores the information about the status of a drilling activity such as drilling of a new well or work-over on a well in a certain period, and the drill rig used. This will be used for monitoring the progress of drilling activity according to the schedule of the well's drilling programme.

Comptest table refers to the results of a completion test conducted on a well after drilling. These are important data used in reservoir modelling. Injectivity index, maximum wellhead pressure, permeability-thickness and the measured depth of circulation losses are the available information.

Casing table holds the description of the casing installed in a well. The type of casing, cement, type of pipe with its diameter, connection, and the length of casing are the information in this table.

Rig table gives the specifications of the rigs used to drill wells.

Powerplant table gives the names of the geothermal power plants in the Philippines with the installed capacity, and their corresponding turbine-generator unit capacity.

Plantspecs table provides the information about the technical specifications of the facilities of the power plant such as the turbine, generator, condenser, cooling tower and pumps.

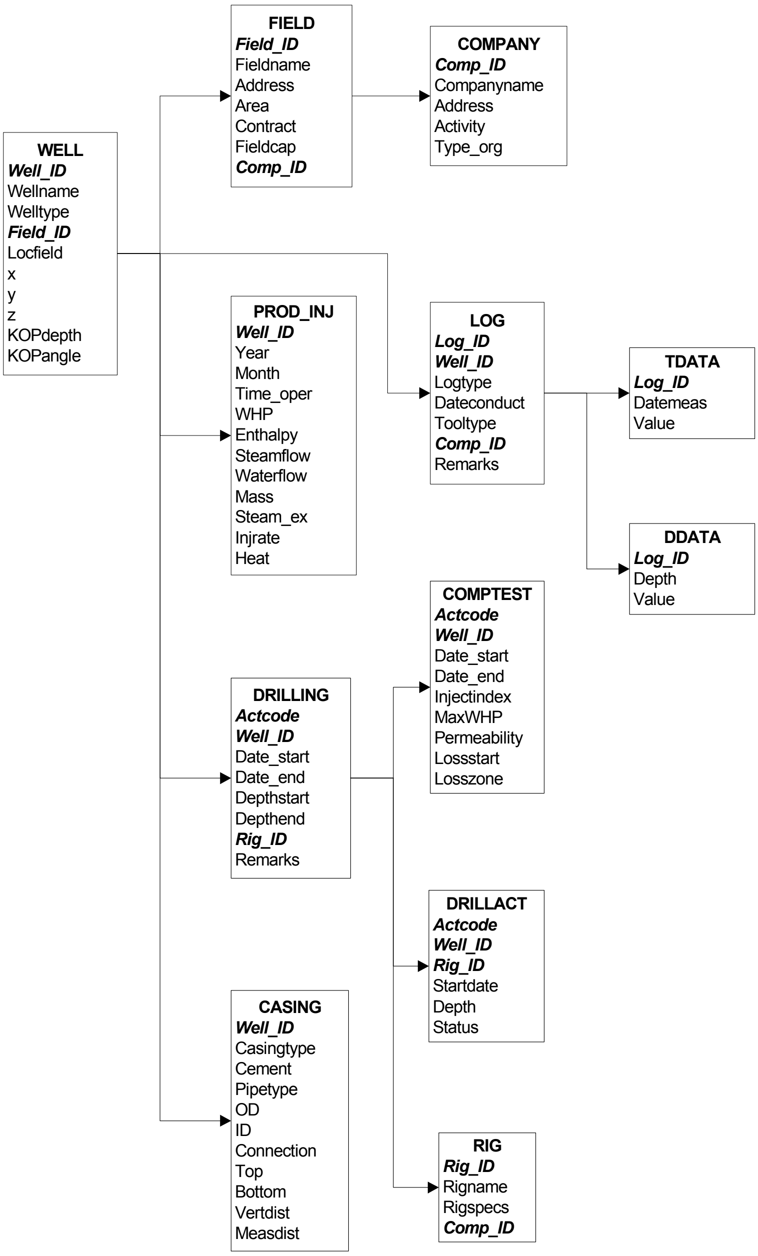


FIGURE 5: Steam field data model

Generation table keeps and stores the electricity generation data of each power plant. This table is used for the geothermal sector reports of DOE that are open for public information.

Oilprice table has the average fuel oil price per month that is needed in the computation of the savings derived from utilizing geothermal energy instead of fuel oil.

These tables were used for the construction of the data models. The purpose of data modelling is to develop graphical representation of the structure of the database design. Two data models were organized. The first one is the structure of a database for monitoring the steam field performance which is composed of the WELL, FIELD, COMPANY, PROD_INJ, LOG, TDATA, DDATA, DRILLING, COMPTEST, CASING, and RIG tables (Figure 5). The second data model comprises POWERPLANT, PLANTSPECS, GENERATION and OILPRICE tables (Figure 6). The idea of making two data models is that, DOE at present, receives and gives information in two separate ways because geothermal energy development in the Philippines is divided and managed also by two entities, the steam field developers and the power plant operators. The name given to each table is the main subject that the field pertain to. The column names given are the specific categories of information that the table contains, and the records are the data relating to the column.

In Figures 5 and 6, table names are in bold capital letters while the primary key(s) are in bold italics. Primary key(s) were selected to uniquely define the occurrence of each record. Choosing a primary key(s) is important because the relationships in the database rely on it. Information from related tables can be retrieved through the primary key(s). It allows the user and the database system to identify, locate, and refer to one particular record in the table. Tables that show relationships between each other are connected by arrows.

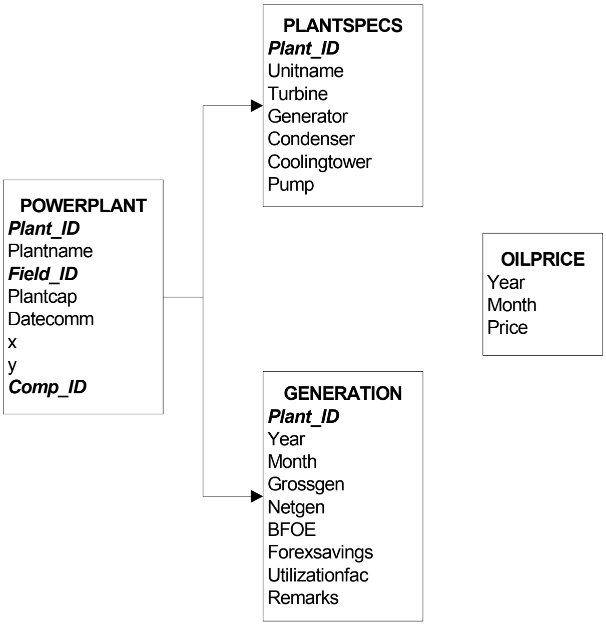


FIGURE 6: Power plant data model

2.3.2 Creation of tables

In the creation of the tables, Structured Query Language (SQL*Plus) was used. It is a standard relational language for database access and manipulation. The language consists of statements or commands to insert, update, delete, query and protect data (Van Der Lans, 1988). An example of a command statement in creating table WELL is shown below. The creation for the other tables with description or comments on each column is given in Appendix I.

```
SQL> create table well (
Well_ID       number(5)     primary key,
Wellname      varchar2(10)  not null,
Welltype      varchar2(10),
Field_ID      number(5),
Locfield      varchar2(30),
x             number(10,2),
y             number(10,2),
z             number(10,2),
KOPdepth      number(6,2),
KOPangle      number(6,2)
)
/
```

After the create table statement, the fields or columns inside the WELL table were specified. Beside each column name is the data type and if necessary, a not null should be specified. Not null will check the data values being entered into the database to ensure that they conform to the data integrity specifications.

2.3.3 Data input, loading and manipulation

Entering rows of data for the tables created was done in several ways. First was by using the insert command. Information for the table Company was entered as shown below:

```
SQL> insert into ops$arfr.company (comp_id, companyname,address,activity,type_org)
values  (10,'PNOC-Energy  Development  Corporation','PNPC  Complex,Merritt  Rd.,Fort
Bonifacio,Makati City','Steam field developer', 'Govt. owned and controlled corp.');

insert into ops$arfr.company (comp_id, companyname,address,activity,type_org)
values  (20,'Philippine  Geothermal,  Inc.','12th  floor,  Citibank  tower,  Paseo  de
Roxas,Makati City','Steam field developer', 'Private company');

insert into ops$arfr.company (comp_id, companyname,address,activity,type_org)
values  (30,'National  Power  Corporation','NPC  Building,Quezon  City','Power  plant
operator', 'Government');

3 records created.
```

The second way was by using the SQL* Loader. This is advisable for entering large volumes of data from external files in a variety of formats such as data in spreadsheets. The SQL* Loader performs filtering by selectively loading records based on the data values, and can load multiple tables simultaneously (Moran and Dimmick, 1987). The production and injection report submitted to DOE is an example of this, because the information here is the mass flow rate, steam flow rate, and water flow rate of each production well, and the injection flow rate of each injection well of a geothermal steam field. Likewise, the AWK programming language in the Unix system was also used to make the loading of data fast and easy. AWK makes data manipulation simple by scanning the input files and splits each input file into columns automatically with very short programs (Aho et al., 1988). Below is an illustration of a script on how the production and injection report was loaded into the database.

```
# MAK-BAN field - data for input in prod_inj table

# 2003-08-29
# The input data came from excel-file (12 sheets)

A=/os/jhskoli/arfr/paperdata/text/makban02.txt
B=/os/jhskoli/arfr/input_in_tables/macban02_prod.ctl

# Creating the header
echo "load data
infile *
into table ops\$arfr.temp_prod_inj
fields terminated by \"    \"
trailing nullcols
(wellname,whp,enthalpy,steamflow,waterflow,mass,injrate,year,month)
begindata" > $B

# Get the data from the textfile
awk 'BEGIN{FS="     "; OFS="     "}
NR >    5 && NR <  109 {print $1,$7,$11,$14,$15,$16,$17,2002,1}
NR >  115 && NR <  219 {print $1,$7,$11,$14,$15,$16,$17,2002,2}
NR >  225 && NR <  340 {print $1,$7,$11,$14,$15,$16,$17,2002,3}
NR >  346 && NR <  450 {print $1,$7,$11,$14,$15,$16,$17,2002,4}
NR >  456 && NR <  571 {print $1,$7,$11,$14,$15,$16,$17,2002,5}
NR >  577 && NR <  692 {print $1,$7,$11,$14,$15,$16,$17,2002,6}
NR >  698 && NR <  813 {print $1,$7,$11,$14,$15,$16,$17,2002,7}
NR >  819 && NR <  924 {print $1,$7,$11,$14,$15,$16,$17,2002,8}
NR >  930 && NR < 1034 {print $1,$7,$11,$14,$15,$16,$17,2002,9}
NR > 1040 && NR < 1145 {print $1,$7,$11,$14,$15,$16,$17,2002,10}
NR > 1151 && NR < 1255 {print $1,$7,$11,$14,$15,$16,$17,2002,11}
NR > 1261 && NR < 1366 {print $1,$7,$11,$14,$15,$16,$17,2002,12}
' $A >> $B
```

The above script created the file `macban02_prod.ctl` which is used as input file for the SQL*Loader command that loaded only the needed column from the production and injection report into the PROD_INJ table. This is a convenient way of data input and manipulation considering that DOE monitors six geothermal production fields in the Philippines.

After all of the available data were loaded into the database, test runs were made to ensure that the database structure works using the SQL relational language. Retrieval of data is the most common SQL operation and therefore, a very good test was necessary to check the database. Database retrieval is called a query and to issue a query, the select command is used. Following are some demonstrations:

```
SQL> select * from field;

FIELD_ID FIELDNAME       ADDRESS                           AREA CONTRACT     FIELDCAP    COMP_ID
-------- --------------- ----------------------------- ---------- ---------- ---------- ---
       1 Tongonan        Tongonan,Leyte                107625 1981-03-14     722.68         10
       2 Mak-Ban         Bay,Laguna                    162000 1973-02-23     425.73         20
       3 Tiwi            Tiwi,Albay                    17661  1971-09-10     330            20
       4 Palinpinon      Palinpinon,Negros Oriental    133000 1981-10-16     192.5          10
       5 Bacman          Albay/Sorsogon                31300  1981-10-16     151.5          10
       6 Mindanao        Kidapawan, North Cotabato     701    1992-03-24     108.48         10

6 rows selected.
```

In querying, the select clause is always followed by the from clause and the asterix (select *) means that all the columns in the FIELD table are selected. To select information in specific columns and specific rows, the statement below can be issued for retrieval:

```
SQL> select well_id, year, month, whp, enthalpy, steamflow, waterflow
  2  from prod_inj
  3  where month = 12
  4  and well_id between 10000 and 10500;

WELL_ID       YEAR      MONTH        WHP   ENTHALPY  STEAMFLOW  WATERFLOW
---------- ---------- ---------- ---------- ---------- ---------- --------
  10000       2002         12        155        720        179        217
  10020       2002         12        217        639        232        460
  10040       2002         12        187        756         85        112
  10060       2002         12        165       1088          0          0
  10080       2002         12        170       1198        107          0
  10100       2002         12        150        481          0          0
  10120       2002         12        160        658         73        131
  10140       2002         12        133        725          0          0
  10160       2002         12        153        985         48         24
  10180       2002         12        167        583         54        136
  10200       2002         12         55        586          0          0
  10220       2002         12        131       1195         13          0
  10240       2002         12        143       1003          0          0
  10260       2002         12        144        542          0          0
  10280       2002         12        139       1195         64          0
  10300       2002         12        126       1194         18          0
  10320       2002         12          0          0          0          0
  10340       2002         12        156       1197          0          0
  10360       2002         12        185        753        159        116
  10380       2002         12        244        551         52        198
  10400       2002         12        158        827          0          0
  10420       2002         12          0          0          0          0
  10440       2002         12          0          0          0          0
  10460       2002         12        147        431          0          0
  10480       2002         12          0          0          0          0
  10500       2002         12        158        700          0          0

26 rows selected.
```

The above statement is searching for a number of specific wells through their unique Well_Id, listing only the columns of interest and data that fall in the month of December. Another good test for the database is the join query. Usually, information that we are looking for may not all be stored in the same table. In the actual scenario, two different tables need to be queried, one after the other, to get the information needed. The join feature selects data from two or more tables and combines the selected data in a single result table. A select statement can be called a join, if the from clause names two or more table identifiers and the where clause specifies at least one condition that compares columns from different tables. The columns in a select statement that define the join are called the join columns. Below is an example that satisfies the conditions of a join query.

```
SQL> column grossgen format 999999.99
SQL> select a.plantname, a.plantcap, b.year, b.month, b.grossgen
  2  from powerplant a, generation b
  3  where a.plantname ='Palinpinon I'
  4  and a.plant_id = b.plant_id
  5  order by a.plantname, a plantcap, b.year, b.month, b.grossgen;
```

```
PLANTNAME                        PLANTCAP       YEAR      MONTH   GROSSGEN
------------------------------ ---------- ---------- ---------- ----------
Palinpinon I                        112.5       2002          1   60357.60
Palinpinon I                        112.5       2002          2   61711.20
Palinpinon I                        112.5       2002          3   55922.40
Palinpinon I                        112.5       2002          4   52012.80
Palinpinon I                        112.5       2002          5   45403.20
Palinpinon I                        112.5       2002          6   43207.20
Palinpinon I                        112.5       2002          7   43142.40
Palinpinon I                        112.5       2002          8   55533.60
Palinpinon I                        112.5       2002          9   62193.60
Palinpinon I                        112.5       2002         10   57477.60
Palinpinon I                        112.5       2002         11   63720.00
Palinpinon I                        112.5       2002         12   63948.80
Palinpinon I                        112.5       2001          1   62676.00
Palinpinon I                        112.5       2001          2   59544.00
Palinpinon I                        112.5       2001          3   46202.40
Palinpinon I                        112.5       2001          4   51739.20
Palinpinon I                        112.5       2001          5   52480.80
Palinpinon I                        112.5       2001          6   55080.00
Palinpinon I                        112.5       2001          7   52776.00
Palinpinon I                        112.5       2001          8   57722.40
Palinpinon I                        112.5       2001          9   62272.80
Palinpinon I                        112.5       2001         10   50601.60
Palinpinon I                        112.5       2001         11   54957.00
Palinpinon I                        112.5       2001         12   63640.80

24 rows selected.
```

In the above table, the columns Plantname and Plantcap come from the POWERPLANT table while columns Year, Month and Grossgen come from the GENERATION table in the database. The two tables were joined because the Plant_ID column from the POWERPLANT table matches the Plant_ID column in the GENERATION table `(a.plant_id = b.plant_id).` In addition to the join-condition, the WHERE clause contains the search condition `(a.plantname ='Palinpinon I').`

3. APPLICATION OF DATABASE IN MONITORING GEOTHERMAL STEAM FIELD AND POWER PLANT PERFORMANCE

The database developed in this project is not only in fulfilment of the requirement to complete the UNU Geothermal Training Programme, but the foremost purpose is the application of this database in DOE's function to monitor the management and operation of the geothermal fields and power plants in the Philippines.

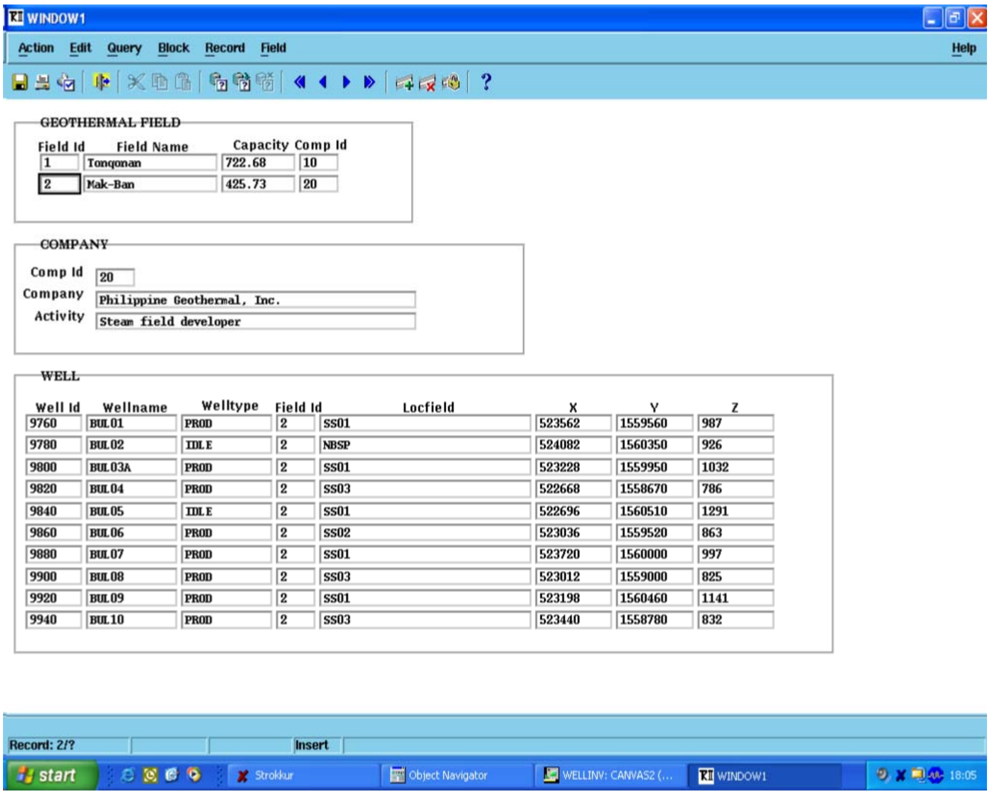
The importance of database can only be felt and realized, if besides storage and retrieval of valuable information, it can also be used in various applications to give meaning to the data inside the database. The Oracle Developer software modules are powerful tools that provide the appropriate database applications that the users require. Here, the author presents the Oracle Developer forms, reports, and graphical output, with the aid of the Gnuplot software, that are used in the monitoring responsibility of DOE.

3.1 Data forms

The Forms Builder is an Oracle Developer tool that enables the end users to retrieve, view, enter, modify, and save information in the database directly into the created on-screen forms. Several data forms were designed in this project to show how this tool can be applied for monitoring geothermal steam fields and power plant operations. All data forms were created through the Forms Builder Data Block Wizard. This wizard allows the quick and easy creation of data blocks based on tables, views or a set of stored procedures (Chu and Lim, 1994). The three essential parts of the Forms Builder are the following:

- Layout Editor that shows the on-screen view of the design form
- Object Navigator that describes the content of the design form such as the data blocks, items and triggers, etc., and
- Property Palette for making the adjustments or changes in the physical properties of the design form.

A data form for the inventory of wells in a specific geothermal steam field as shown in Figure 7 is composed of three data blocks from FIELD, COMPANY and WELL tables. Relationships among these data blocks were created based on join conditions. Similar columns from different tables satisfy the join condition. Usually, the primary key(s) of the tables (Figures 5 and 6) are chosen for the join condition. The form for the inventory of wells can be used to view all the wells drilled in a selected geothermal steam field with their corresponding location in the field. Additional wells that will be drilled can be entered directly into the form simply by typing the necessary information on the empty box after the last record.



GEOTHERMAL FIELD

Field Id	Field Name	Capacity	Comp Id
1	Tonqonan	722.68	10
2	Mak-Ban	425.73	20

COMPANY

Comp Id: 20
Company: Philippine Geothermal, Inc.
Activity: Steam field developer

WELL

Well Id	Wellname	Welltype	Field Id	Locfield	X	Y	Z
9760	BUL01	PROD	2	SS01	523562	1559560	987
9780	BUL02	IDLE	2	NBSP	524082	1560350	926
9800	BUL03A	PROD	2	SS01	523228	1559950	1032
9820	BUL04	PROD	2	SS03	522668	1558670	786
9840	BUL05	IDLE	2	SS01	522696	1560510	1291
9860	BUL06	PROD	2	SS02	523036	1559520	863
9880	BUL07	PROD	2	SS01	523720	1560000	997
9900	BUL08	PROD	2	SS03	523012	1559000	825
9920	BUL09	PROD	2	SS01	523198	1560460	1141
9940	BUL10	PROD	2	SS03	523440	1558780	832

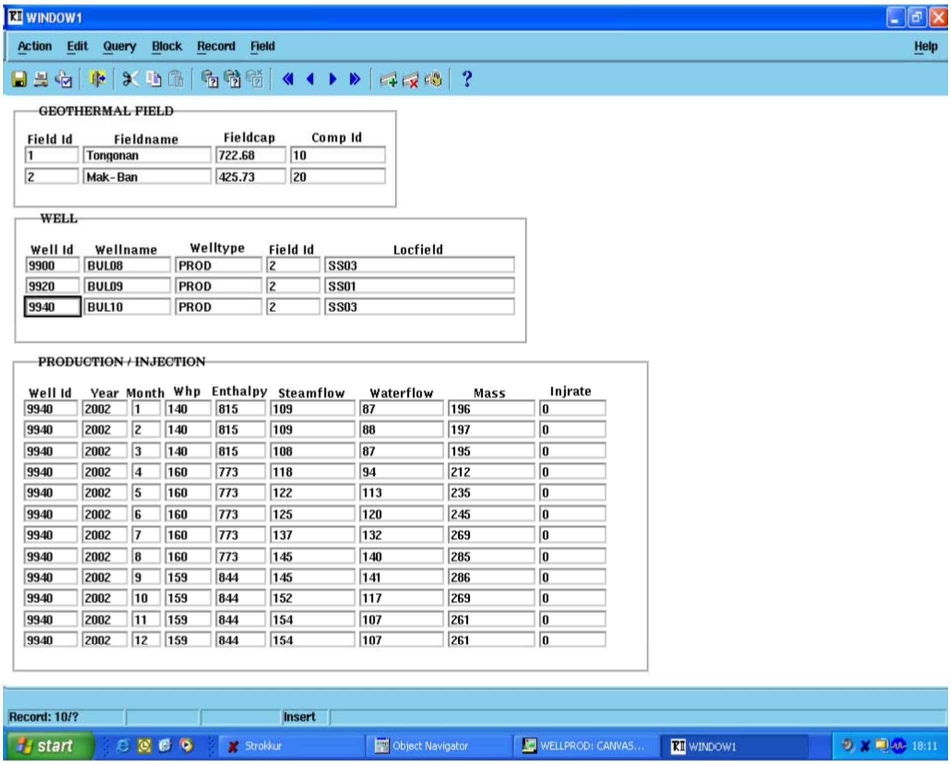
Record: 2/? Insert

FIGURE 7: Form for the inventory of wells

A data form for the production and injection rate of each well in a specific geothermal field is shown in Figure 8. The design of this form is composed of FIELD, WELL and PROD_INJ tables, that enables the user or viewer to easily compare the performance of a well in each month. This is a good application of the database to monitor the steam field performance through a sudden change or big difference of all the well's production or injection rates. Input of new records for the succeeding months can be done in the next empty record.

A log information and log result conducted in a well is shown in the data form in Figure 9. Data blocks consist of the FIELD, WELL, LOG, TDATA and DDATA tables. Data from this form will be used in plotting temperature and pressure profiles of wells with respect to time or depth.

Figure 10 shows a data form for drilling information and the status of a drilling activity in a well. The design of this form, which includes the FIELD, WELL, DRILLING and DRILLACT tables, will be used for monitoring the progress of a drilling activity as compared with the drilling programme.



GEOTHERMAL FIELD

Field Id	Fieldname	Fieldcap	Comp Id
1	Tongonan	722.68	10
2	Mak-Ban	425.73	20

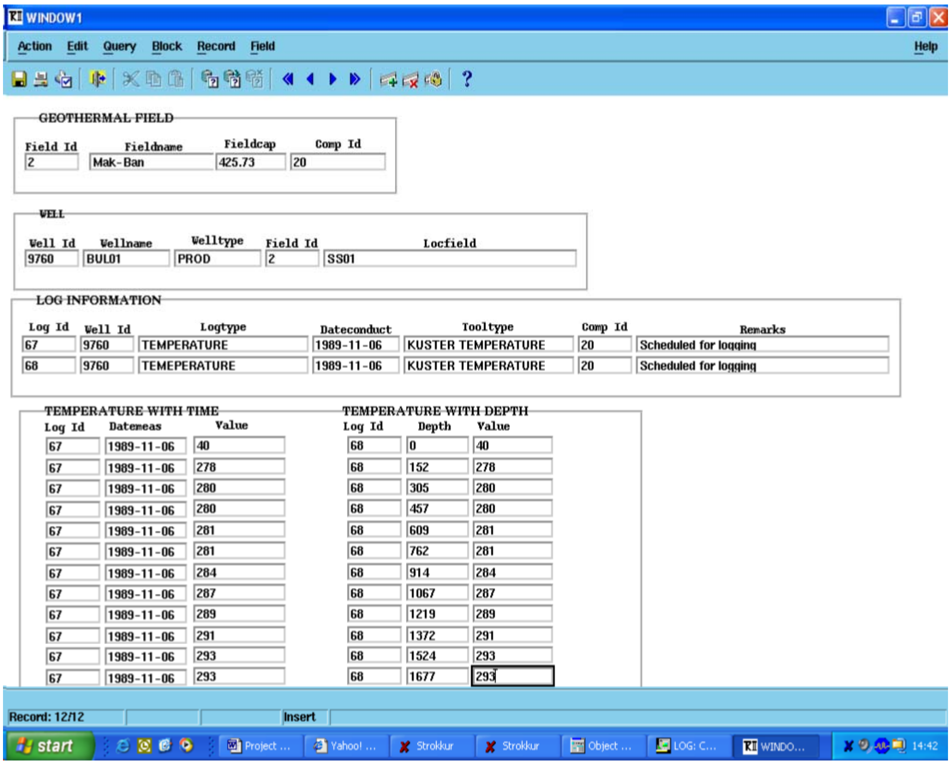
WELL

Well Id	Wellname	Welltype	Field Id	Locfield
9900	BUL08	PROD	2	SS03
9920	BUL09	PROD	2	SS01
9940	BUL10	PROD	2	SS03

PRODUCTION / INJECTION

Well Id	Year	Month	Whp	Enthalpy	Steamflow	Waterflow	Mass	Injrate
9940	2002	1	140	815	109	87	196	0
9940	2002	2	140	815	109	88	197	0
9940	2002	3	140	815	108	87	195	0
9940	2002	4	160	773	118	94	212	0
9940	2002	5	160	773	122	113	235	0
9940	2002	6	160	773	125	120	245	0
9940	2002	7	160	773	137	132	269	0
9940	2002	8	160	773	145	140	285	0
9940	2002	9	159	844	145	141	286	0
9940	2002	10	159	844	152	117	269	0
9940	2002	11	159	844	154	107	261	0
9940	2002	12	159	844	154	107	261	0

FIGURE 8: Form for production or injection rates of each well



GEOTHERMAL FIELD

Field Id	Fieldname	Fieldcap	Comp Id
2	Mak-Ban	425.73	20

WELL

Well Id	Wellname	Welltype	Field Id	Locfield
9760	BUL01	PROD	2	SS01

LOG INFORMATION

Log Id	Well Id	Logtype	Dateconduct	Tooltype	Comp Id	Remarks
67	9760	TEMPERATURE	1989-11-06	KUSTER TEMPERATURE	20	Scheduled for logging
68	9760	TEMEPERATURE	1989-11-06	KUSTER TEMPERATURE	20	Scheduled for logging

TEMPERATURE WITH TIME

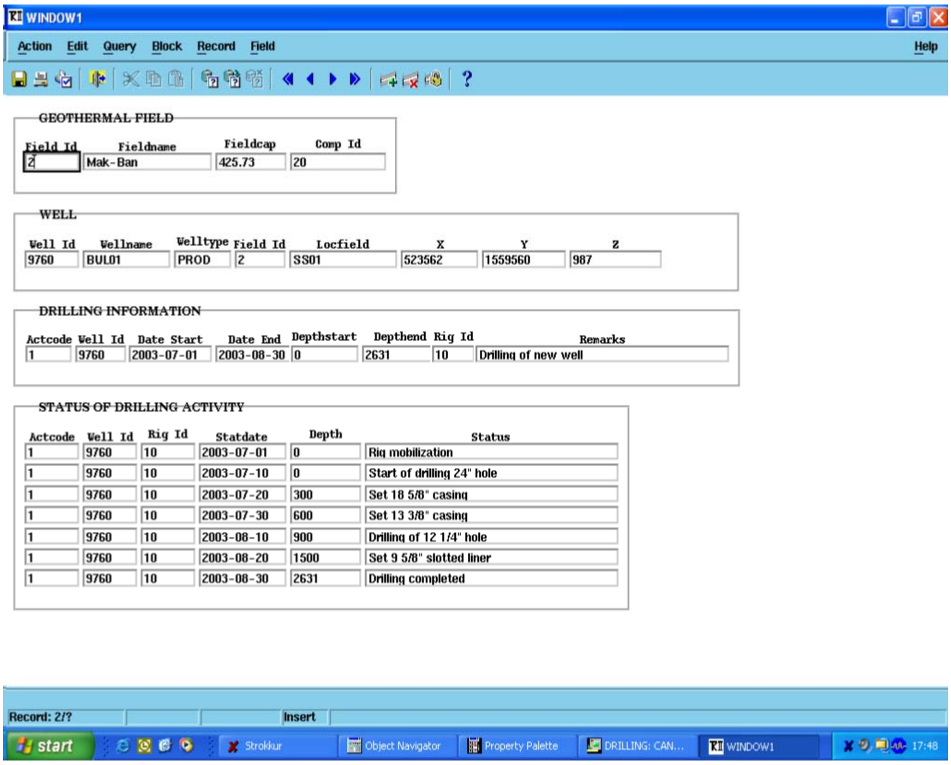
Log Id	Datemeas	Value
67	1989-11-06	40
67	1989-11-06	278
67	1989-11-06	280
67	1989-11-06	280
67	1989-11-06	281
67	1989-11-06	281
67	1989-11-06	284
67	1989-11-06	287
67	1989-11-06	289
67	1989-11-06	291
67	1989-11-06	293
67	1989-11-06	293

TEMPERATURE WITH DEPTH

Log Id	Depth	Value
68	0	40
68	152	278
68	305	280
68	457	280
68	609	281
68	762	281
68	914	284
68	1067	287
68	1219	289
68	1372	291
68	1524	293
68	1677	293

FIGURE 9: Form for log information and log results

Data on power generation of the geothermal power plants is shown in the data form in Figure 11. Comprising the form are data blocks from the POWERPLANT, COMPANY and GENERATION tables. The design of the form provides a clear picture of the each power plant's generation every month, which is a good application for monitoring the performance of the power plant. A big decrease in generation in a certain month can be easily noticed, as DOE should know the reason for such change.



GEOTHERMAL FIELD

Field Id	Fieldname	Fieldcap	Comp Id
2	Mak-Ban	425.73	20

WELL

Well Id	Wellname	Welltype	Field Id	Locfield	X	Y	Z
9760	BUL01	PROD	2	SS01	523562	1559560	987

DRILLING INFORMATION

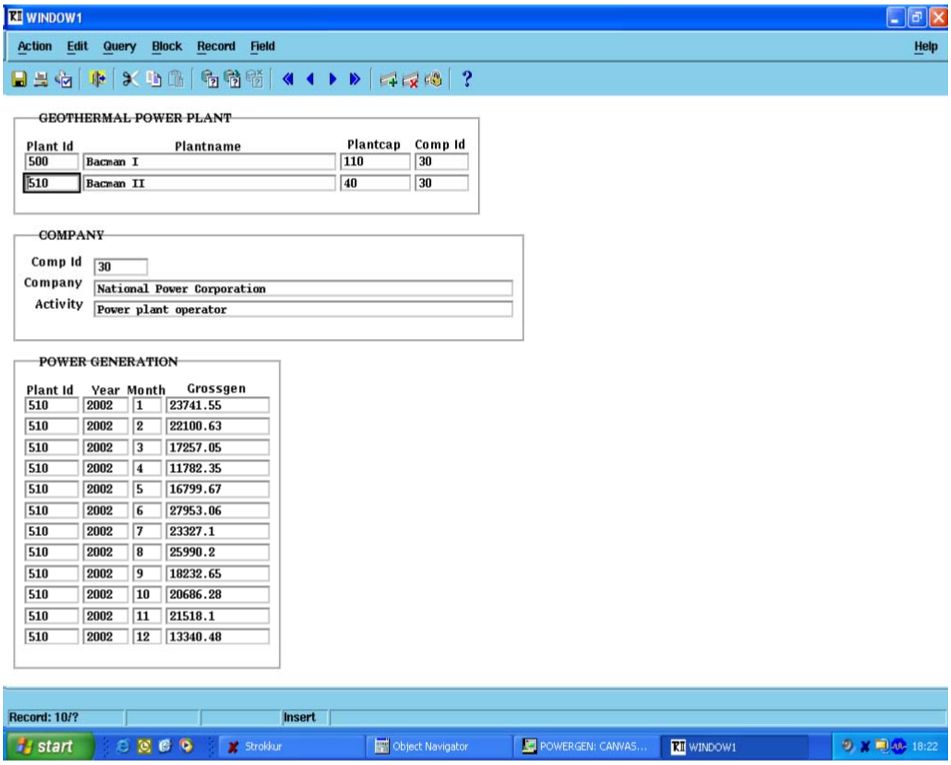
Actcode	Well Id	Date Start	Date End	Depthstart	Depthend	Rig Id	Remarks
1	9760	2003-07-01	2003-08-30	0	2631	10	Drilling of new well

STATUS OF DRILLING ACTIVITY

Actcode	Well Id	Rig Id	Statdate	Depth	Status
1	9760	10	2003-07-01	0	Rig mobilization
1	9760	10	2003-07-10	0	Start of drilling 24" hole
1	9760	10	2003-07-20	300	Set 18 5/8" casing
1	9760	10	2003-07-30	600	Set 13 3/8" casing
1	9760	10	2003-08-10	900	Drilling of 12 1/4" hole
1	9760	10	2003-08-20	1500	Set 9 5/8" slotted liner
1	9760	10	2003-08-30	2631	Drilling completed

Record: 2/? Insert

FIGURE 10: Form for drilling information and status of drilling activity



GEOTHERMAL POWER PLANT

Plant Id	Plantname	Plantcap	Comp Id
500	Bacman I	110	30
510	Bacman II	40	30

COMPANY

Comp Id: 30
Company: National Power Corporation
Activity: Power plant operator

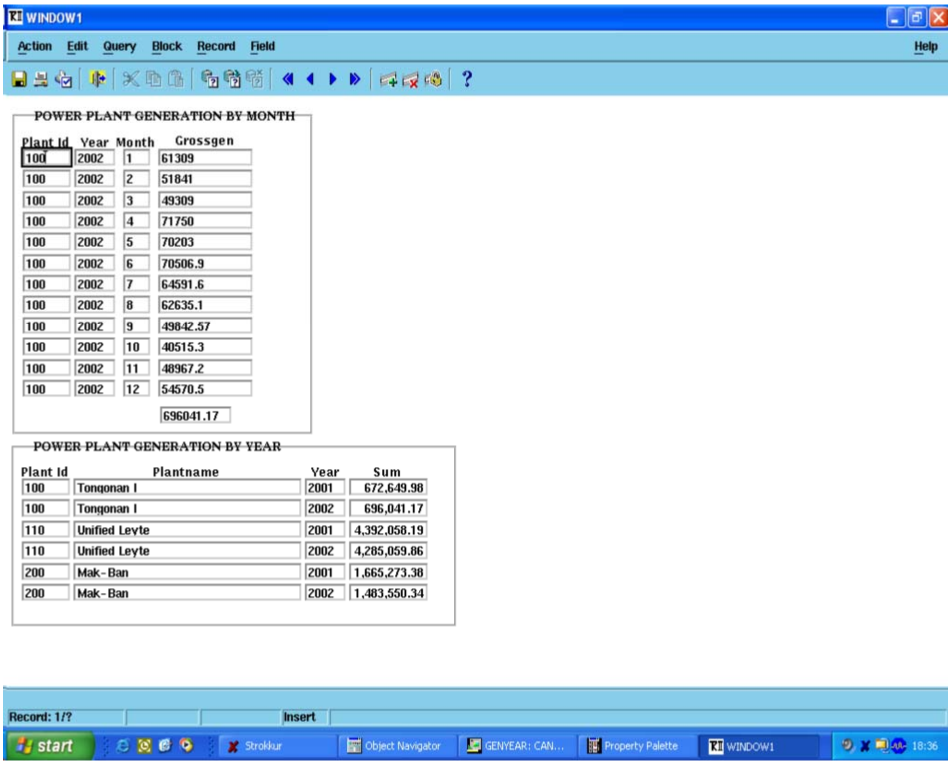
POWER GENERATION

Plant Id	Year	Month	Grossgen
510	2002	1	23741.55
510	2002	2	22100.63
510	2002	3	17257.05
510	2002	4	11782.35
510	2002	5	16799.67
510	2002	6	27953.06
510	2002	7	23327.1
510	2002	8	25990.2
510	2002	9	18232.65
510	2002	10	20686.28
510	2002	11	21518.1
510	2002	12	13340.48

Record: 10/? Insert

FIGURE 11: Form for the power generation of the power plants

The capability of utilizing the form for simple computations is illustrated in Figure 12. The first data block that comes from the GENERATION table shows the monthly generation of a selected power plant. Then, a new data block was created for the computation of the sum of the generation through a post-query trigger using a PL/SQL command. Finally, the third data block shows the sum of the generation of certain power plants in chosen years. It was created from ALL_GENERATION view (Appendix II) having the columns Plant_ID, Plantname, Year and Sum. This form is designed to easily view and compare the generation of a power plant in two succeeding years. Performance of the power plants can therefore quickly be assessed.



POWER PLANT GENERATION BY MONTH

Plant Id	Year	Month	Grossgen
100	2002	1	61309
100	2002	2	51841
100	2002	3	49309
100	2002	4	71750
100	2002	5	70203
100	2002	6	70506.9
100	2002	7	64591.6
100	2002	8	62635.1
100	2002	9	49842.57
100	2002	10	40515.3
100	2002	11	48967.2
100	2002	12	54570.5
			696041.17

POWER PLANT GENERATION BY YEAR

Plant Id	Plantname	Year	Sum
100	Tongonan I	2001	672,649.98
100	Tongonan I	2002	696,041.17
110	Unified Leyte	2001	4,392,058.19
110	Unified Leyte	2002	4,285,059.86
200	Mak-Ban	2001	1,665,273.38
200	Mak-Ban	2002	1,483,550.34

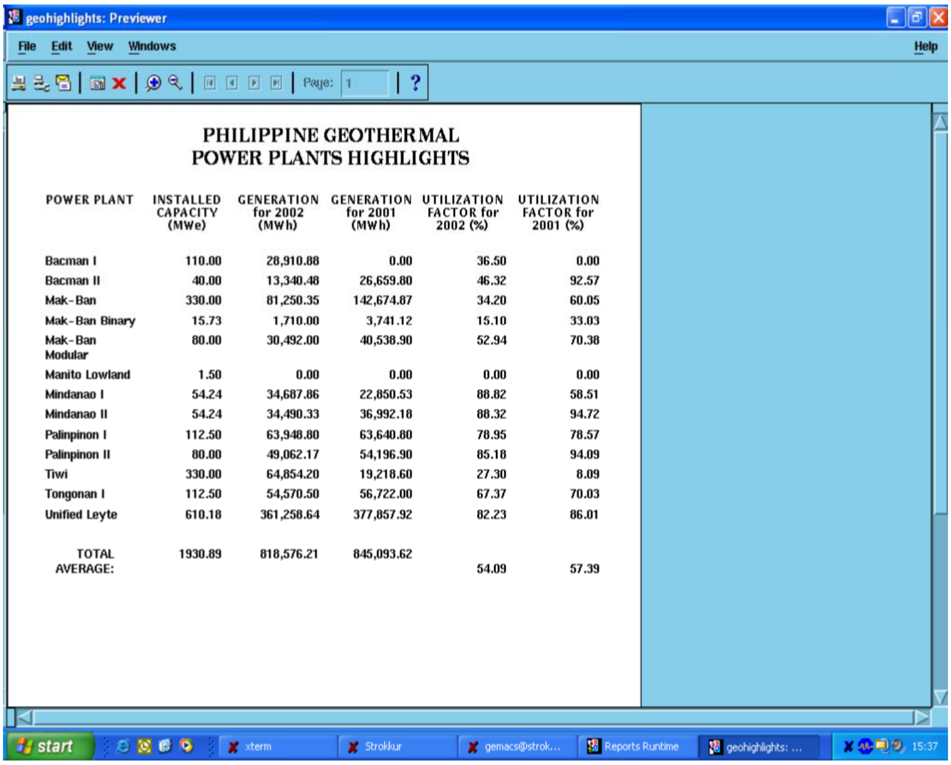
FIGURE 12: Form showing computation (sum) of the power generation of a selected power plant

The data forms presented in this project are only some of the illustrations from the author's perception on how this tool can be applied in monitoring the performance of geothermal steam fields and power plants in the Philippines. Evaluation of these forms still need to go through necessary improvements.

3.2 Report layout

Organizing reports about the situation of the Philippine geothermal sector is one of the main tasks of DOE. These reports are usually provided for public information. A sample report layout was constructed using the Oracle Developer Report Builder software. This is a database application tool for building standard and customized reports from simple tabular to complex matrix report styles readily accessible for a better view as compared with the forms.

The report "Philippine geothermal power plants highlights" presented in Figure 13 was created using the Report Wizard of the Report Builder. The wizard follows a systematic procedure but requires knowledge of the SQL commands. Adjustments or changes of the design report are also done in the Report Editor, Object Navigator, and Property Palette of the Report Builder. The columns of the said report are chosen from ALL_GENERATION view (Appendix II). The format of the report gives the electricity generation of the power plants with their corresponding computed utilization factor as compared with the previous year's performance. Usually, low generation, which is equal to low utilization factor, indicates problems in the facilities of the power plant.



PHILIPPINE GEOTHERMAL
POWER PLANTS HIGHLIGHTS

POWER PLANT	INSTALLED CAPACITY (MWe)	GENERATION for 2002 (MWh)	GENERATION for 2001 (MWh)	UTILIZATION FACTOR for 2002 (%)	UTILIZATION FACTOR for 2001 (%)
Bacman I	110.00	28,910.88	0.00	36.50	0.00
Bacman II	40.00	13,340.48	26,659.80	46.32	92.57
Mak-Ban	330.00	81,250.35	142,674.87	34.20	60.05
Mak-Ban Binary	15.73	1,710.00	3,741.12	15.10	33.03
Mak-Ban Modular	80.00	30,492.00	40,538.90	52.94	70.38
Manito Lowland	1.50	0.00	0.00	0.00	0.00
Mindanao I	54.24	34,687.86	22,850.53	88.82	58.51
Mindanao II	54.24	34,490.33	36,992.18	88.32	94.72
Palinpinon I	112.50	63,948.80	63,640.80	78.95	78.57
Palinpinon II	80.00	49,062.17	54,196.90	85.18	94.09
Tiwi	330.00	64,854.20	19,218.60	27.30	8.09
Tongonan I	112.50	54,570.50	56,722.00	67.37	70.03
Unified Leyte	610.18	361,258.64	377,857.92	82.23	86.01
TOTAL	1930.89	818,576.21	845,093.62		
AVERAGE:				54.09	57.39

FIGURE 13: Report on the Philippine geothermal power plants highlights

3.3 Graphical output

Graphical output is the third application of the database undertaken in this project. For the purpose of monitoring the steam fields, graphing the wellhead pressure, enthalpy, and mass flow rate of individual wells are some of the parameters being determined. This will provide information on the well performance such as decline in production, which may require decision of improving the well. The graphs are created in two command files, the SQL commands and the Gnuplot commands. The SQL commands import the data from the database, and the Gnuplot commands draw the data into an X-Y graph (Williams and Kelley, 1993) as presented in Appendix III.

The graph of mass flow rate with time (Figure 14) of a well may show a constant production rate but might demonstrate decline in the future due to mineral deposition.

Graph of wellhead pressure with time (Figure 15) is also used to accompany an evaluation of well performance. A drop of the wellhead pressure at the same wellhead opening may be a result of well blockages. Work-over of wells are usually the solution to this problem that includes mechanical drilling out of blockages and/or acid treatment.

Figure 16 shows a graph of enthalpy with time, which is another requirement for a sound judgement in improving a well's production. Increase in discharge enthalpy with a decrease in mass flow but without change in steam flow (Figure 17) is possibly caused by reservoir pressure drawdown. On the other hand, decline in enthalpy perhaps may be associated with reservoir cooling induced by inflow of natural recharge or injection returns (Sarmiento, 2000). Assessment of the production and injection strategies is mostly suggested for making the necessary resource management changes. Whatever actions and decisions that the steam field developers will carry out, it is obligation that DOE be informed.

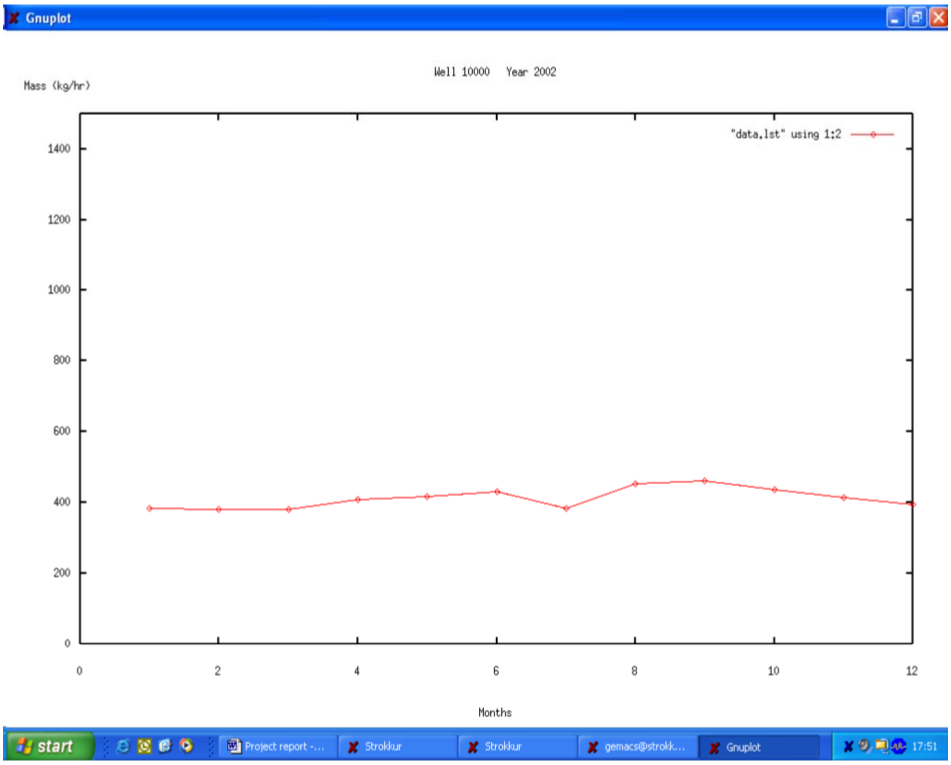


FIGURE 14: Graph of mass flow rate in a well vs. time

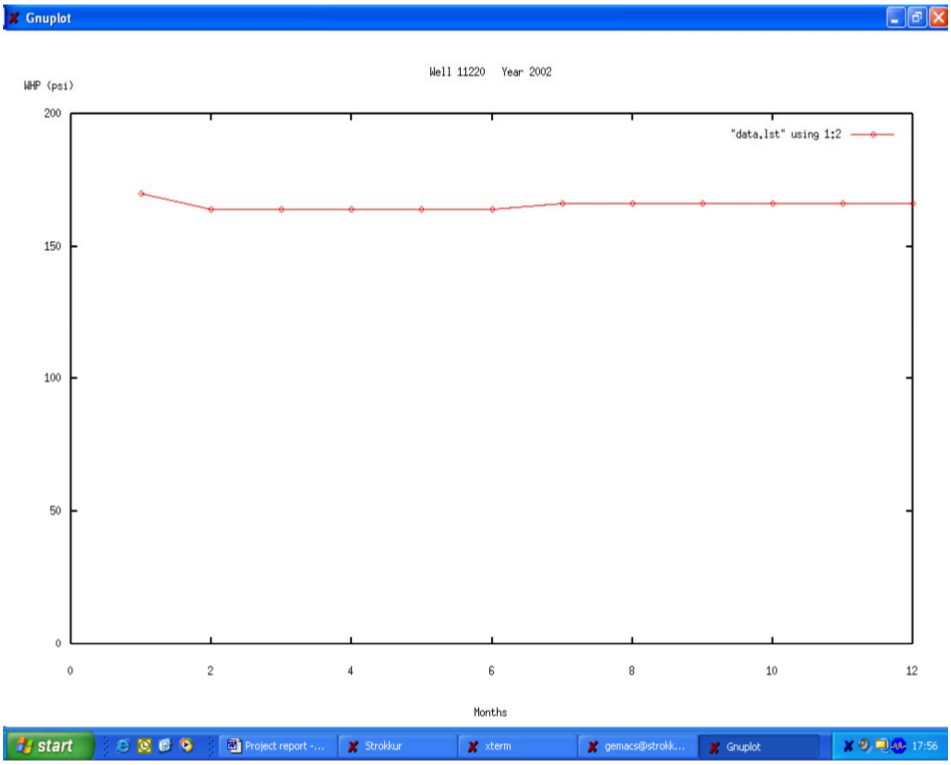


FIGURE 15: Graph of wellhead pressure in a well vs. time

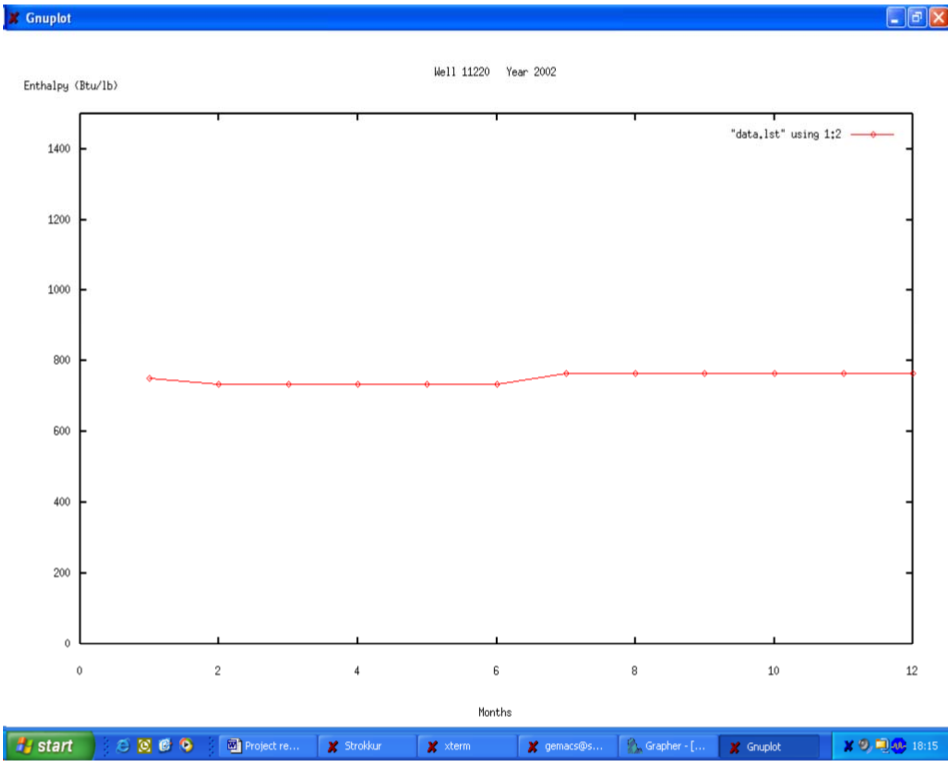


FIGURE 16: Graph of enthalpy in a well vs. time

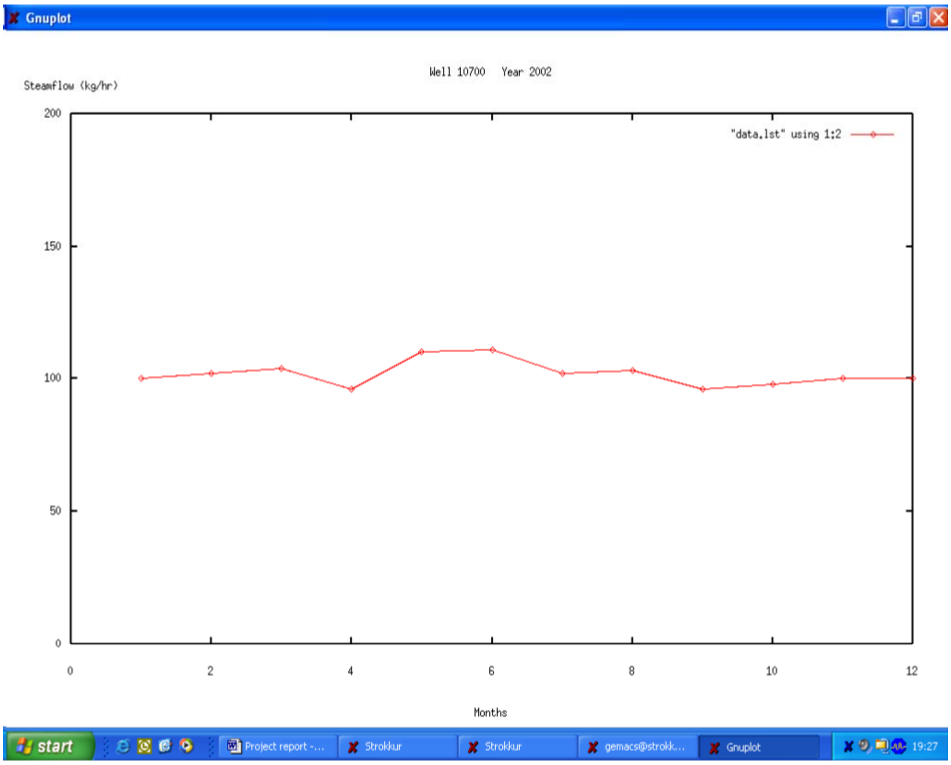


FIGURE 17: Graph of steam flow rate in a well vs. time

Temperature decline normally accompanies decline in enthalpy. Shifting of production or injection strategies is usually adopted to address the declines in temperature and enthalpy. Typically, increasing enthalpies would indicate drawdown resulting in expanding two-phase horizons and declining water levels in the well. This increase could also result in higher steam flow, which could meet or even surpass the steam requirement for the power plant (Malate, 2003).

4. CONCLUSIONS AND RECOMMENDATIONS

Geothermal energy development in the Philippines, for about 30 years now, has grown up utilizing its own experience and of course with the knowledge and skills shared by other countries such as Iceland. Experience was gained from the volume of data gathered and studied aiming for the sustainable management of the geothermal resource.

Accordingly, the Philippine Department of Energy (DOE) being the government agency mandated to supervise and regulate geothermal operations, recognizes that development and application of a computerized database is an important tool for monitoring the management strategies of geothermal steam fields and power plants in the country.

The database structure was developed and processed using the Oracle Relational Database Management System (RDBMS). Two data models designed are the steam field and the power plant data models. The steam field model composes tables that stores and provides data about the physical characteristics of the wells, drilling information and reservoir monitoring parameters. While, the power plant data model consists of tables that handle and provide information about the specifications and power generation of the power plants. Data from the paper reports were entered into the tables of the database using the SQL*Loader software. Test runs were conducted to check the design of the database according to the objectives. The capability of the database was integrated to create data forms, report layout and graphical outputs that can be used in monitoring the performance of the steam fields and power plants in the Philippines. Oracle Developer Forms, Reports Builder, and Gnuplot softwares were utilized in the said procedure.

As a result, the database developed here can be readily implemented and applied as a tool for monitoring the performance of steam fields and power plants in the Philippines, and is a good beginning for DOE to improve from its existing manual filing system. However, it is recommended that an additional and continuous study on database is performed. In addition, as a part of monitoring, more exposure in geothermal operations is required for DOE in verifying data integrity submitted by the steam field developers and power plant operators. DOE should also go along with the advancement of database and geothermal technology.

ACKNOWLEDGEMENTS

First of all, I would like to give glory to God for giving me the strength to successfully complete the UNU Geothermal Training Programme for six months in Iceland. I give my sincere gratitude to Dr. Ingvar B. Fridleifsson and Mr. Lúdvík S. Georgsson for the award of Fellowship as well as their continuing support to the Philippines; and Ms. Gudrún Bjarnadóttir and Maria-Victoria Gunnarsson for taking care of all our needs. All the lecturers, particularly the reservoir engineering teaching staff, are thanked for their dedication to impart their expertise. My patient supervisor, Mr. Hilmar Sigvaldason is thanked for sharing everything he knows about database and always being present in times of need from project preparation to project completion. All of my classmates, especially the reservoir engineering group, thanks for the friendly environment we felt with each other; also my Filipino “kababayans” in Iceland that made my stay an enjoyable one; and of course, to my colleague Jem for being a brother to me.

I also would like to thank Mr. Francisco A. Benito, Mr. Ranilo P. Abando, Mr. Ramon Allan V. Oca, Mr. Vicente M. Karunungan, Ms. Alicia N. Reyes and Mr. Alejandro F. Oanes for endorsing me in this training; and my officemates, mainly Mr. Edvin D. Butiu, Ms. Vanessa S. Lazaro and Ms. Elizabeth N. Tilos, for taking time to give me the necessary data that I needed. Lastly, I am grateful to my family, my wife Nhing and my daughter Arabella for being my inspiration in everything I do.

REFERENCES

Aho, A., Kerninghan, B., and Weinberger, P., 1988: *The AWK programming language*. Addison-Wensley Publ. Co., USA, 210 pp.

Chu, K., and Lim, G., 1994: *Forms developer's guide. Release 4.5.* Oracle Corp., Ireland, 478 pp.

Ellison, L., 1989: *ORACLE Introduction to SQL.* Oracle Corporation, U.S.A., 42 pp.

Ensor D., and Stevenson I., 1997: *Oracle design.* O'Reilly & Associates, Inc., U.S.A., 513 pp.

Geothermal Division-DOE, 2002: *2002 Year-end Report of the Geothermal Sector..* Department of Energy, Philippines, Internal report, 40 pp.

Malate, R.C.M., 2003: Management of geothermal resources: PNOC-EDC experience. In: *Lectures on the sustainable use and operating policy for geothermal resources*. Short course prior to the International Geothermal Conference, IGC 2003. UNU-GTP, Iceland, 153-183.

Moran, R., and Dimmick, S., 1987: *SQL*Loader users's guide.* Oracle Corporation, Belmont, Ca., U.S.A., 78 pp.

Requejo, R.A., 1996: Monitoring of production decline and pressure drawdown in geothermal reservoirs using decline curves analysis method. Report 14 in: *Geothermal Training in Iceland 1995.* UNU-GTP, Iceland, 311-340.

Salvania, N., 1995: Development of a geothermal database and resource assessment of Mt. Natib geothermal prospect, Philippines. Report 10 in: *Geothermal Training in Iceland 1995. UNU-GTP,* Iceland, 241-268.

Sarmiento, Z.F., 2000: Physical monitoring II: high-enthalpy geothermal systems. In: *Long-term monitoring of high- and low-enthalpy fields under exploitation*. World Geothermal Congress 2000, pre-congress short courses, Kokonoe, Japan, May, 23-56.

Van Der Lans, R.F., 1988: *Introduction to SQL.* Addison-Wesley Publ. Co., England, 348 pp.

Williams T., and Kelley, C., 1993: *Gnuplot reference manual.* 70 pp.

APPENDIX I: The scripts in the creation of the database tables in SQL*Plus

1) Well table

```
create table well (
Well_ID              number(5)            primary key,
Wellname             varchar2(10)         not null,
Welltype             varchar2(10),
Field_ID             number(5),
Locfield             varchar2(30),
x                    number(10,2),
y                    number(10,2),
z                    number(10,2),
KOPdepth             number(6,2),
KOPangle             number(6,2)
)
/
--
```

```
comment on table ops$arfr.well is
'table describing the wells';
comment on column ops$arfr.well.well_id is
'unique number for each well';
comment on column ops$arfr.well.wellname is
'name of the well';
comment on column ops$arfr.well.welltype is
'type of well such as production or injection';
comment on column ops$arfr.well.field_id is
'unique number for each steam field where the well is located';
comment on column ops$arfr.well.locfield is
'location of the well in the field like mahiao sector or separator station 01';
comment on column ops$arfr.well.x is
'x coordinate, easting based on map with a scale of 1:50,000 provided by the National
Mapping Resource Information Authority (NAMRIA) in the Philippines [m]';
comment on column ops$arfr.well.y is
'y coordinate, northing based on map with a scale of 1:50,000 provided by the National
Mapping Resource Information Authority (NAMRIA) in the Philippines [m]';
comment on column ops$arfr.well.z is
'z elevation above sea level [m]';
comment on column ops$arfr.well.KOPdepth is
'kick-off-point, depth of deviation of the well from surface level [m]'
comment on column ops$arfr.well.KOPangle is
'kick-off-point, angle of deviation of the well [˚]like 15˚';
```

2) Field table

```
create table field (
Field_ID            number(5)           primary key,
Fieldname           varchar2(15)        not null,
Address             varchar2(30)        not null,
Area                       number(10,2),
Contract            date,
Fieldcap            number(6,2),
Comp_ID             number(10)
)
/
--
comment on table ops$arfr.field is
'table containing information about the geothermal steam fields in the Philippines';
comment on column ops$arfr.field.field_id is
'unique number for each steam field';

comment on column ops$arfr.field.fieldname is
'name of steam field such as Bacman for Bacman Geothermal Production Field';
comment on column ops$arf.field.address is
'location of the steam field in the Philippines such as Tiwi,Albay for Tiwi Geothermal
Field';
comment on column ops$arfr.field.area is
'service contract area of the steam field [hectares]. service contract is the contract
between the government and a company as a steam field developer particularly the
contract between the Phil. Department of Energy and PNOC-EDC';
comment on column ops$arfr.field.contract is
'date that the service contract was awarded';
comment on column ops$arfr.field.fieldcap is
'the installed capacity of the power plant in the steam field in mega-watt electric
[MWe]';
comment on column ops$arfr.field.comp_id is
'unique number for each company in the Philippines involved in geothermal energy such
as steam field developer for PNOC-EDC';
```

3) Company table

```
create table company (
Comp_ID             number(5)           primary key,
Companyname         varchar2(50)        not null,
Address             varchar2(100),
Activity            varchar2(50),
Type_org            varchar2(50)
)
/
--
```

```
comment on table ops$arfr.company is
'table containing the information about the companies in the Philippines involved in
geothermal energy';
comment on column ops$arfr.company.comp_id is
'unique number for each company';
comment on column ops$arfr.company.companyname is
'name of the company such as PNOC-Energy Development Corporation';
comment on column ops$arf.company.address is
'the main office address of the company';
comment on column ops$arfr.company.activity is
'type of activity/work that the company is doing such as steam field developerfor
Philippine Geothermal,Inc. and power plant operator for National Power Corporation';
comment on column ops$arfr.company.type_org is
'type of organization of the company such as PNOC-EDC is a government owned and
controlled corporation, and Philippine Geothermal,Inc. is a private company';
```

4) Prod_Inj table

```
create table prod_inj (
Well_ID             number(5)           not null,
Year                       number(4)           not null,
Month                      number(2)           not null,
Time_oper           number(3),
WHP                        number(5,2),
Enthalpy            number(4),
Steamflow           number(10,1),
Waterflow           number(10,1),
Mass                       number(10,1),
Steam_ex            number(10,1),
Injrate             number(10,1),
Heat                       number(8,2),
primary key (well_id,year,month)
)
/
--
comment on table ops$arfr.prod_inj is
'table containing the mass and heat extraction and injection on production and
injection wells';
comment on column ops$arfr.prod_inj.well_id is
'unique number for each well';
comment on column ops$arfr.prod_inj.year is
'the year of production or injection of the well';
comment on column ops$arfr.prod_inj.month is
'the month of production or injection of the well';
comment on column ops$arfr.prod_inj.time_oper is
'the time that the well was operated for production or injection [hrs]';
comment on column ops$arfr.prod_inj.whp is
'wellhead pressure [MPa]';
comment on column ops$arfr.prod_inj.enthalpy is
'the measured enthlpy of the well [kJ/kg]';
comment on column ops$arfr.prod_inj.steamflow is
'steam flow rate [kg/s]';
comment on column ops$arfr.prod_inj.waterflow is
'separated water flow rate [kg/s]';
comment on column ops$arfr.prod_inj.mass is
'total mass extracted or injected [tonne]';
comment on column ops$arfr.prod_inj.steam_ex is
'steam extracted from the well [tonne]';
comment on column ops$arfr.prod_inj.injrate is
'injection rate to an injection well [kg/s]';
comment on column ops$arfr.prod_inj.heat is
'heat extracted or injected [TJ]';
```

5) Log table

```
create table log (
Log_ID              number(5)                  primary key,
Well_ID             number(5)           not null,
Logtype             varchar2(20)        not null,
Dateconduct         date,
Tooltype            varchar2(20),
Comp_ID             number(5),
Remarks             varchar2(30)
)
/
--
comment on table ops$arfr.log is
'table containing log information';
comment on column ops$arfr.log.log_id is
'unique number for each log conducted';
comment on column ops$arfr.log.well_id is
'unique number for each well';
comment on column ops$arfr.log.logtype is
'type of logging parameter being measured such as temperature/pressure logs or diameter
of the well';
comment on column ops$arfr.log.dateconduct is
'date conducted the log';
comment on column ops$arfr.log.tooltype is
'type of measuring tool used for logging such as Kuster temperature/pressure or caliper
tool ';
comment on column ops$arfr.log.comp_id is
'unique number for each company in the Philippines involved in geothermal energy that
conducted the log';
comment on column ops$arfr.log.remarks is
'remarks/comments about the log';
```

6) Tdata table

```
create table tdata (
Log_ID                    number(5)           not null,
Datemeas            date,
Value                     number(4,2)
)
/
--
comment on table ops$arfr.tdata is
'table of log data with respect to time';
comment on column ops$arfr.tdata.log_id is
'unique number for each log conducted';
comment on column ops$arfr.tdata.datemeas is
'date of measurement';
comment on column ops$arfr.tdata.value is
'measured data such as temperature [˚C] or pressure [MPa]';
```

7) Ddata table

```
create table ddata (
Log_ID                    number(5)           not null,
Depth                     number(5,1),
Value                     number(4,2)
)
/
--
comment on table ops$arfr.ddata is
'table of log data with respect to depth';
comment on column ops$arfr.ddata.log_id is
'unique number for each log conducted';
comment on column ops$arfr.ddata.depth is
'depth of measurement [m]';
comment on column ops$arfr.ddata.value is
'measured data such as temperature [˚C] or pressure [MPa]';
```

8) Drilling table

```
create table drilling (
Actcode             number(5)           not null,
Well_ID             number(5)           not null,
Date_start          date,
Date_end            date,
Depthstart          number(6,2),
Depthend            number(6,2),
Rig_ID              number(5),
Remarks             varchar2(30)
)
/
--
comment on table ops$arfr.drilling is
'table about the drilling information on a well';
comment on column ops$arfr.drilling.actcode is
'a code number to distinguish the drilling activity conducted, (1) for new well
drilling and (2) for well work-over such as casing repair, acidizing and blockage
clearing';
comment on column ops$arfr.drilling.well_id is
'unique number for each well';
comment on column ops$arf.drilling.date_start is
'date of start of drilling';
comment on column ops$arfr.drilling.date_end is
'date of end of drilling';
comment on column ops$arfr.drilling.depthstart is
'starting depth of drilling from surface level [m]';
comment on column ops$arfr.drilling.depthend is
'final depth of drilling from surface level [m]';
comment on column ops$arfr.drilling.rig_id is
'unique number for each rig used in drilling';
comment on column ops$arfr.drilling.remarks is
'the description of the purpose, status in a certain period, type of drilling activity
such as new well drilling or workover';
```

9) Drillact table

```
create table drillact (
Actcode             number(5)           not null,
Well_ID             number(5)           not null,
Rig_ID                     number(5),
Statdate            date,
Depth                      number(6,2),
Status                     varchar2(30)
)
/
--
comment on table ops$arfr.drillact is
'table about the drilling activity on a well';
comment on column ops$arfr.drillact.actcode is
'a code number to distinguish the drilling activity conducted, (1) for new well
drilling and (2) for well work-over such as casing repair, acidizing and blockage
clearing';
comment on column ops$arfr.drillact.well_id is
'unique number for each well';
comment on column ops$arfr.drillact.rig_id is
'unique number each drilling rig performing the drilling activity';
comment on column ops$arf.drillact.statdate is
'date according to the status of the drilling activity';
comment on column ops$arfr.drillact.depth is
'depth of drilling as of the date of status of the drilling activity';
comment on column ops$arfr.drillact.status is
'the description of the status of the drilling activity according to date';
```

10) Comptest table

```
create table comptest (
Actcode             number(5)           not null,
Well_ID             number(5)           not null,
Date_start          date,
Date_end            date,
Injectindex         number(4,1),
MaxWHP                     number(5,2),
Permeability        number(8,5),
Lossstart           number(6,2),
Lossend             number(6,2)
)
/
--
comment on table ops$arfr.comptest is
'table containing the result of the completion test done to the well';
```

```
comment on column ops$arfr.comptest.actcode is
'a code number to distinguish the drilling activity conducted, (1) for new well
drilling and (2) for well work-over such as casing repair, acidizing and blockage
clearing';
comment on column ops$arfr.comptest.well_id is
'unique number for each well';
comment on column ops$arfr.comptest.date_start is
'date of start of the test';
comment on column ops$arfr.comptest.date_end is
'date of end of the test';
comment on column ops$arfr.comptest.injectindex is
'injectivity index measured during the test [li/s-MPa]';
comment on column ops$arfr.comptest.maxWHP is
'maximum wellhead pressure [MPa]';
comment on column ops$arfr.comptest.permeability is
'permeability-thickness [darcy-meter]';
comment on column ops$arfr.comptest.losstart is
'measured depth at start of circulation loss [m]';
comment on column ops$arfr.comptest.lossend is
'measured depth at end of circulation loss [m]';
```

11) Casing table

```
create table casing (
Well_ID             number(5)           primary key,
Casingtype          character(25),
Cement                     varchar2(20),
Pipetype            varchar2(20),
OD                         number(4,1),
ID                         number(4,1),
Connection          varchar2(20),
Top                        number(5,1),
Bottom                     number(5,1),
Vertdist            number(5,1),
Measdist            number(5,1)
)
/
--
comment on table ops$arfr.casing is
'table containing the casing information of the wells';
comment on column ops$arfr.casing.well_id is
'unique number for each well';
comment on column ops$arfr.casing.casingtype is
'type of casing like anchor casing';
comment on column ops$arfr.casing.cement is
'type of cement used for casing';
comment on column ops$arfr.casing.pipetype is
'type of pipe used';
comment on column ops$arfr.casing.od is
'outside diameter of pipe [mm]';
comment on column ops$arfr.casing.id is
'inside diameter of pipe [mm]';
comment on column ops$arfr.casing.connection is
'type of casing connection used';
comment on column ops$arfr.casing.top is
'top of casing from surface level [m]';
comment on column ops$arfr.casing.bottom is
'bottom of casing from surface level [m]';
comment on column ops$arfr.casing.vertdist is
'measured vertical length of casing from surface level [m]';
comment on column ops$arfr.casing.measdist is
'total measured length of casing from surface level [m]';
```

12) Rig table

```
create table rig (
Rig_ID                     number(5)           primary key,
Rigname             varchar2(20)        not null,
Rigspecs            varchar2(30),
Comp_ID             number(5)
)
/
--
comment on table ops$arfr.rig is
'table about the geothermal drilling rig';
comment on column ops$arfr.rig.rig_id is
'unique number for each drilling rig';
comment on column ops$arfr.rig.rigname is
'rig name';
comment on column ops$arfr.rig.rigspecs is
'technical specifications of the rig';
comment on column ops$arfr.rig.comp_id is
'unique number for each company in the Philippines involved in geothermal energy that
conducted the drilling';
```

13) Powerplant table

```
create table powerplant (
Plant_ID            number(5)           primary key,
Plantname           varchar2(30)        not null,
Field_ID            number(5),
Plantcap            number(6,2),
Datecomm            date,
x                   number(12,2),
y                   number(12,2),
Comp_ID             number(5)
)
/
--
comment on table ops$arfr.powerplant is
'table about the capacity and location of the power plant';
comment on column ops$arfr.powerplant.plant_id is
'unique number for each geothermal power plant';
comment on column ops$arfr.powerplant.plantname is
'power plant name';
comment on column ops$arfr.powerplant.field_id is
'unique number for each steam field';
comment on column ops$arfr.powerplant.plantcap is
'total megawatt-electric capacity of the power plant [MWe]';
comment on column ops$arfr.powerplant.datecomm is
'date of commissioning of the power plant';
comment on column ops$arfr.powerplant.x is
'x coordinate, easting based on map with a scale of 1:50,000 provided by the National
Mapping Resource Information Authority (NAMRIA) in the Philippines [m]';
comment on column ops$arfr.powerplant.y is
'y coordinate, easting based on map with a scale of 1:50,000 provided by the National
Mapping Resource Information Authority (NAMRIA) in the Philippines [m]';
comment on column ops$arfr.powerplant.comp_id is
'unique number for each company in the Philippines involved in geothermal energy such
as power plant operator for National Power Corporation';
```

14) Plantspecs table

```
create table plantspecs (
Plant_ID            number(5)           primary key,
Unitname            varchar2(30)        not null,
Turbine             varchar2(30),
Generator           varchar2(30),
Condenser           varchar2(30),
Coolingtower        varchar2(30),
Pump                        varchar2(30)
)
/
--
comment on table ops$arfr.plantspecs is
'table about the technical specifications of the facilities of the power plant';
comment on column ops$arfr.plantspecs.plant_id is
'unique number for each power plant';
comment on column ops$arfr.plantspecs.unitname is
'name given to each unit of a power plant such as Mak-Ban Unit 1';
```

```
comment on column ops$arfr.plantspecs.turbine is
'turbine specifications';
comment on column ops$arfr.plantspecs.generator is
'generator specifications';
comment on column ops$arfr.plantspecs.condenser is
'condenser specifications';
comment on column ops$arfr.plantspecs. coolingtower is
'cooling tower specifications';
comment on column ops$arfr.plantspecs.pump is
'pump specifications';
```

15) Generation table

```
create table generation (
Plant_ID            number(5),
Year                      number(4)           not null,
Month                     number(2)           not null,
Netgen                    number(10,2),
Grossgen            number(10,2),
BFOE                      number(5,2),
Forexsavings        number(5,2),
Utilizationfac      number(5,2),
Remarks             varchar2(50)
)
/
--
comment on table ops$arfr.generation is
'table about the electricity generated by the power plant';
comment on column ops$arfr.generation.plant_id is
'unique number for each power plant';
comment on column ops$arfr.rig.generation.year is
'year of generation';
comment on column ops$arfr.rig.generation.month is
'month of generation';
comment on column ops$arfr.generation.netgen is
'net generation of the power plant [MWh] ';
comment on column ops$arfr.generation.grossgen is
'gross generation of the power plant [MWh]';
comment on column ops$arfr.generation.bfoe is
'barrels of fuel oil equivalent [MMBFOE]';
comment on column ops$arfr.generation.forexsavings is
'foreign exchange savings [MM$]';
comment on column ops$arfr.generation.utilizationfac is
'utilization factor of the power plant [%]';
comment on column ops$arfr.generation.remarks is
'remarks/comments about the generation';
```

16) Oilprice table

```
create table oilprice (
Year                      number(4)           not null,
Month                     number(2)           not null,
Price                     number(3,2)
)
/

comment on table ops$arfr.oilprice is
'table about the average fuel oil price to compute the savings using geothermal
energy';
comment on column ops$arfr.oilprice.year is
'the year of the oil price';
comment on column ops$arfr.oilprice.month is
'the month of the oil price';
comment on column ops$arfr.oilprice.price is
'average price of fuel oil in a month [$/barrel] ';
```

Appendix II: The scripts in the creation of views in SQL*Plus

1) All_wells view

```
create  view  all_wells  (well_id,  wellname,  welltype,x,y,z,  locfield,  field_id,
fieldname, address, area, contract, fieldcap, comp_id, companyname)
as
select a.well_id, a.wellname, a.welltype, a.x, a.y, a.z, a.locfield, a.field_id,
b.fieldname, b.address, b.area, b.contract, b.fieldcap, b.comp_id, c.companyname
from ops$arfr.well a, ops$arfr.field b, ops$arfr.company c
where a.field_id = b.field_id
and b.comp_id = c.comp_id
/
```

2) All_generation view

```
create view all_generation (plant_id, plantname, plantcap, unitcap,datecomm,
x, y, field_id, fieldname, comp_id, companyname,
year, month, grossgen, netgen, bfoe, forexsavings, utilizationfac, remarks
)
as
select a.plant_id, a.plantname, a.plantcap, a.unitcap, a.datecomm,
a.x, a.y, a.field_id, b.fieldname, a.comp_id, c.companyname,
d.year,  d.month,  d.grossgen,  d.netgen,  d.bfoe,  d.forexsavings,  d.utilizationfac,
d.remarks
from ops$arfr.powerplant a, ops$arfr.field b, ops$arfr.company c, ops$arfr.generation
d
where a.field_id = b.field_id (+)
and a.comp_id = c.comp_id (+)
and a.plant_id = d.plant_id (+)
/
```

Appendix III: The scripts in the creation of graphs with SQL*Plus commands and Gnuplot commands run in Unix system

```
# We create 2 commandfiles - 1. sql-commands and 2. gnuplot-commands
##########
# Create sql-command file (run.sql) to select data from database
# Data is spooled into file data.lst
 if [[ $# -lt 6 || $1 = "-h" ]]
 then
  echo "
             display WHP of a well over one year
             Usage: c.sh -T TypeOfLog -y year -w well_id
             for example: c.sh -T WHP -y 2002 -w 11800
             TypeOfLog:
             WHP : Well Head Pressure
             E   : Enthalpy
             M   : Mass
             S   : Steamflow
             Options:
             -y year       :      Year
             -w well_id    :      ID of well
             " 1>&2
             exit 1
             fi
# Options
set -- $(getopt "y:w:T:" "$@")
for arg in $@
do
case $arg in
 -y) year=$2 shift 2;;
 -w) well=$2 shift 2;;
 -T) type=$2 shift 2;;
 --) shift;  break;;
esac
done
```

```
############ Well Head Pressure
if test $type = "WHP"
then
echo "
set echo off
set feedback off
set flush off
set term off
set verify off
set newpage 0
set lines 80
set pages 0
spool data
select month,whp
from ops\$arfr.prod_inj
where year = $year
and well_id = $well
order by month;
spool off
quit" > run.sql
####
# Run the sql-commands - and data goes to file data.lst
sqlplus / @run.sql > /dev/null
##########
# Create gnuplot-commands file (run.gnu)
echo "set title 'Well $well   Year $year'
set xlabel 'Months'
set ylabel 'WHP (psi)'
set yrange [0:200]
plot \"data.lst\" using 1:2 with linespoints
pause -1" > run.gnu
####
# Run the gnuplot-file
gnuplot run.gnu
fi
############### Enthalpy
if test $type = "E"
then
echo "
set echo off
set feedback off
set flush off
set term off
set verify off
set newpage 0
set lines 80
set pages 0
spool data
select month,enthalpy
from ops\$arfr.prod_inj
where year = $year
and well_id = $well
order by month;
spool off
quit" > run.sql
####
# Run the sql-commands - and data goes to file data.lst
sqlplus / @run.sql > /dev/null
##########
# Create gnuplot-commands file (run.gnu)
echo "set title 'Well $well   Year $year'
set xlabel 'Months'
set ylabel 'Enthalpy (Btu/lb)'
set yrange [0:1500]
plot \"data.lst\" using 1:2 with linespoints
pause -1" > run.gnu
####
# Run the gnuplot-file
gnuplot run.gnu
fi
```

```
############### Mass
if test $type = "M"
then
echo "
set echo off
set feedback off
set flush off
set term off
set verify off
set newpage 0
set lines 80
set pages 0
spool data
select month,mass
from ops\$arfr.prod_inj
where year = $year
and well_id = $well
order by month;
spool off
quit" > run.sql
####
# Run the sql-commands - and data goes to file data.lst
sqlplus / @run.sql > /dev/null
##########
# Create gnuplot-commands file (run.gnu)
echo "set title 'Well $well   Year $year'
set xlabel 'Months'
set ylabel 'Mass (kg/hr)'
set yrange [0:1500]
plot \"data.lst\" using 1:2 with linespoints
pause -1" > run.gnu
####
# Run the gnuplot-file
gnuplot run.gnu
fi
############ Steamflow
if test $type = "S"
then
echo "
set echo off
set feedback off
set flush off
set term off
set verify off
set newpage 0
set lines 80
set pages 0
spool data
select month,steamflow
from ops\$arfr.prod_inj
where year = $year
and well_id = $well
order by month;
spool off
quit" > run.sql
####
# Run the sql-commands - and data goes to file data.lst
sqlplus / @run.sql > /dev/null
##########
# Create gnuplot-commands file (run.gnu)
echo "set title 'Well $well   Year $year'
set xlabel 'Months'
set ylabel 'Steamflow (kg/hr)'
set yrange [0:200]
plot \"data.lst\" using 1:2 with linespoints
pause -1" > run.gnu
####
# Run the gnuplot-file
gnuplot run.gnu
fi
```