Proceedings World Geothermal Congress 2015
Melbourne, Australia, 19-25 April 2015

Design and Use of Relational Databases in the Geothermal Sector

Sigrún Gunnarsdóttir

Iceland GeoSurvey, Grensásvegur 9, 108 Reykjavík, Iceland.

Sigrun.Gunnarsdottir@isor.is

Keywords: Relational database management systems; data normalization; database user interfaces; geothermal information; geothermal data; geothermal database; Nicaraguan geothermal database; database for overview of geothermal work; African geothermal inventory database;

ABSTRACT

Geothermal information has been gathered worldwide for decades, through research and costly projects. This information is vital and requires safe and accessible storage. Doing this reliably is a challenge for supervisory authorities and all institutions involved. Currently such geothermal data, if stored in computerized form, is probably most often stored in so-called "flat file" databases of text files and/or Excel files, which can be useful when the amount of data are considered small and simple, but they have their limitations.

Using Relational Database Management Systems (RDBMS) for designing a database can be very beneficial to geothermal work. The benefits of a database as a relational is among other things to ensure data consistency, to avoid data duplication, minimize human errors, and maximize data security and accessibility. Data can be retrieved quickly with ease and compared on a large or a small scale, from time to time, from within and between areas and reservoirs, and within or between individual wells and other locations of measurements or sampling. Information can also be shared more easily, resulting in improved collaboration and thus advancing geothermal work around the world.

The main outlines of two kinds of relational databases regarding geothermal work will be described here. One for data gathered from geothermal exploration and production and the other for overview of geothermal development. Their design purpose and general usage will be presented, including the main structure, individual parts, tables and their internal relations, including a few significant table views and notable output examples.

1. INTRODUCTION

The Scottish geologist Charles Lyell introduced in his influential book *Principles of Geology* the doctrine of uniformitarianism where the central argument was based on the observation that "*the present is the key to the past*" (The University of Vermont, 2008). By using modern information technology handling data gathered from geothermal exploration and production it can be said that *the past and the present pave future pathways*. Thus storing invaluable geothermal data and metadata is vital, and it is crucial that it will be done by the best available means.

In many cases geothermal data are not yet stored in computerized file form and may only be preserved in word or text processed reports in computers. Geothermal data that has been computerized may then not be stored in central computers as yet, but different parts of the information only kept in computers of individual employees. It can be said that proper data storage is lacking far and wide, which puts it at great risk of time consuming data recovery and even permanent data loss.

Information technology nowadays offers some great software tools for storing and accessing data and by using them rightly one can greatly increase the potential for data utilization. Modern databases are structured storage facilities in computer systems, designed to offer an organized mechanism for storing, managing and retrieving information. Decades ago a computerized database used to be an extremely technical term used only by large institutions or companies, but however, with the rise of computer systems and information technology the database has, so to say, become a household term. The relational model (RDBMS: Relational Database Management System) is the most popular type and has been around for over three decades. It is an extremely powerful tool to store information, and to access it as well. Relational databases are currently the predominant choice in storing financial records, medical records, manufacturing and logistical information, personnel data and much more (Wikipedia, 2014).

The ability to retrieve data quickly from a database, that is well designed inside a relational database system, to look at the data from different points of view and to share it with collaborators, can considerably improve understanding, insight and overview of projects and thus result in better work procedures in all stages of the geothermal work.

At ÍSOR (Iceland GeoSurvey) we have developed two kinds of relational databases regarding geothermal work. The first one is ÍSOR‘s database and a similar database for the Geothermal Division of Ministry of Energy and Mines in Nicaragua. Both of these are designed for data gathered from geothermal exploration and production, such as the various measurements values and relevant location dependent information and metadata (Section 4). The other is a Database of Geothermal Development in Africa for ARGeo. This database is designed for an overview of the geothermal development, regarding projects, their progress and their funding, human capacity and equipment available or required at the various sites concerned (Section 5).

2. FROM FLAT FILE SYSTEMS OVER TO RELATIONAL DATABASES

"*Relational Databases – Not your Father's Flat Files*" says Linda Woodard, consultant in database design and usage in scientific applications at Cornell University, Center for Advanced Computing, who has also been working on spatial models in theoretical ecology. To store data in "flat files" is not problematic when the amount is considered small and simple, but it has its limitations (Woodard, 2008).

As already mentioned, information technology offers nowadays some great software tools for storing and accessing data, the relational database management system (RDBMS) being the most popular and an extremely powerful tool to store and access information. The reasons for the dominance of relational databases are significant; they have been offering the best mixture of simplicity and flexibility, performance and compatibility in managing data. The advanced data structuring capability of the relational database allows database builders and programmers to create complex relationships between the various data. To repeat data between tables and to keep unnecessary data in a relational database is against the design rules, takes extra space on hard disks and drives, and multiplies the possibilities of human error in handling the data. The benefits of a database as a relational is to make the most use of the data, to get benefits as to maximize data security, ensure data consistency, avoid duplication of data, minimize human error, and to increase the accessibility of the data. Although databases are built of columns and rows, like spreadsheets, they are far more powerful. Many actions can be performed on a database and would be difficult if not impossible to perform on a spreadsheet, these few can be mentioned: retrieve all records that match certain criteria; update records in bulk; cross-reference records in different tables; perform complex aggregate calculations. In order for a database to be truly functional, it must: store large amounts of records well; be accessed easily; new information and changes should also be fairly easy to input. Besides these features, all databases that are created should be built with high data integrity and the ability to recover data if hardware fails (Gunnarsdóttir, 2013a).

Before entering data into a relational database there are some stages of design. A good design of a relational database structure in the beginning can prevent levels of complexity at later stages. The design of a relational database is not dependent on what system (RDBMS) will consequently be chosen for its use, so the main design of a database can take place before that choice has been made. In the relational design so-called normalization is used, that is a data-structuring model which ensures data consistency and avoids data duplication, making sure not to repeat data in or between tables. Such a design should in fact force the database users to use rules for data entry. A relational database is designed with a group, or groups, of tables which are made up of columns and rows. So-called constraints and relationships are defined in and between the tables. A relational database can hold a special kind of information in one table while storing various other related data in separate tables, all of which are linked together by primary and foreign keys to look at a bigger picture. Derivatives of the various values kept in tables in modern databases should not themselves be stored there in table columns. There is no reason to do so, since they can easily be drawn out from the values, whenever needed, by using the data manipulation facilities and calculations possibilities of the SQL database language and the relational system‘s inbuilt functions, see Section 3 (Gunnarsdóttir, 2013a).

A specially designed web-interface for interaction with a relational geothermal database is not necessarily required, although useful, as some standard tools for interaction could be quite sufficient (Section 3).

It is of high importance to stick to the main design of a relational database and the tables interrelations regarding entering data. To use an older system of flat text files or excel files has maybe been useful, nice and somehow easy, but people must not let the old methodology control their new relational database structure. The users should thus not at all depart from or “bend“ the relational design of their new database so they can kind of “copy/paste“ the structure and content of older files into the tables of the new database. To use a powerful database tool in that way, like it is some form of a super Excel, is not exploiting its possibilities. For users in charge it is advised that they have the relational diagrams of their database at disposal whilst they are internalizing the “relational way of thinking“. Finally, even though the relational keys and the constraints guide us, general caution must be shown by users when entering or changing data in a relational database, and retrieving data from it, as is true for any other database.

3. SYSTEMS AND SOFTWARE

3.1 Systems for Relational Database Management

The variety of RDBMS available is considerable, from expensive ones, used in the financial sector and elsewhere, over to open source software. The less expensive or free ones can even be enough for many geothermal companies or institutions. All these products provide for the creation of rights or privileges that can be associated with a specific database user. According to Curt A. Monash, a leading analyst of and strategic advisor to the software industry, it hardly matters what online transaction processing design is concerned, or transaction volume: "*... you can probably get the job done with any of Oracle, DB2, or Informix. Postgres Plus isn't far behind. If you're only using standard alphanumeric datatypes, the list expands further, to include products from Microsoft, Sybase, Progress, and perhaps MySQL as well.*" (Monash, 2008). Following the database design work of the databases described in Sections 4 and 5, a relational database management system (RDBMS) was chosen, which was Oracle for ÍSOR’s database and *MS SQL Server* for the Nicaraguan database, which both are databases designed for geothermal exploration and production data, and *MySQL* for AGID, ARGeo‘s database, which is intended for overview of geothermal in Africa.

3.2 SQL – Structured Query Language for Managing Data

As the Indiana University Knowledge Base says: "SQL is an ANSI and ISO standard, and is the de facto standard database query language. A variety of established database products support SQL, including products from Oracle and Microsoft SQL Server. It is widely used in both industry and academia, often for enormous, complex databases." (Indiana University IT Services Knowledge Base, 2013). Although SQL is not capable of as many computations as are computer languages like C, C++, Java, etc., it has data manipulation facilities beyond them. Thus, to get the benefits from both kinds of languages, SQL statements often are embedded into host languages, like the ones just mentioned, to handle so-called queries, calls, to a database. If not used embedded, SQL is specified by a so-called command-line tool. SQL has some minor syntax differences, but the majority of it is standard across MS Access, MS SQL Server, Oracle, Sybase, MySQL, Postgres, Informix, etc. (Holowczak, 2011). SQL can be used to select derivatives of the various values kept in database tables so derivatives of values should not themselves be stored in the tables. They can easily be drawn out from the values, whenever needed, by using the calculations possibilities of SQL and the relational system‘s inbuilt functions, like sum, average, max, min, sin, power, etc.

3.3 User Interfaces

A specially designed interface for interaction with a relational database is often not required, although useful. If a special user interface is to be designed for a particular database, web-based or not, it is done separately and usually following the database design. Many methods or tools can be used for interaction with a database, open source or not. Some standardized interface tools can be both inexpensive and quite good. They can list data in tabular forms as well as offer some in-built graphical possibilities. A variety of such established database products support SQL, like MS-Access (which can be used to connect with databases as well as storing a small one too), Navicat, dbVisualizer, etc. Many of them have in-built features, like import and export wizards, report and query builders. The various database queries results, so-called outputs, can in many such tools be exported to text files, XML files, HTML documents, and Excel files. So the convenience of using Excel to handle data, which is widely the case, is not at all lost. Map functions are not part of standardized interface tools. To show on maps the parts of the geothermal data that are location dependent, like areas and the various locations within them, directional paths of boreholes, etc., there must either be some spatial extension system (spatial DBMS) of the database system, like PostGIS for PostgreSQL, SpatiaLite for SQLite, etc., or a specially designed interface capable of map display. For the databases designed at ÍSOR, a special web-based interface has also been developed for interaction, see Sections 4 and 5. For the ARGeo database (AGID) such an interface was necessary, since the various users in charge of entering data into the database and updating it are working at it in many countries, and also because this database will in the future allow open access for the public.

4. RELATIONAL DATABASES FOR GEOTHERMAL EXPLORATION AND PRODUCTION

ÍSOR‘s geothermal database is designed for storing data gathered from geothermal exploration and production, like various measurement values and relevant location dependent information and metadata. Besides developing our own database, a similar database has been designed at ÍSOR for the geothermal division of Nicaraguan Ministry of Energy and Mines (MEM), from late 2009 to 2012, within the framework of the ICEIDA Geothermal Capacity Building Project in Nicaragua (Gunnarsdóttir et al., 2012a). Besides these database design projects there have been ongoing corresponding projects at ÍSOR of developing a special web-interface to interact with ÍSOR‘s database and the Nicaraguan database. Teaching material has also been prepared for the users of the database in Nicaragua, with some 150 useful database queries for managing and displaying data (Gunnarsdóttir, 2012b). Databases like these, with or without a special web-interface, could be taken into use elsewhere in geothermal companies or institutions, after some redesign work or changes and adaption to local conditions.

4.1 Design Purpose

As said above, ÍSOR‘s geothermal database and the Nicaraguan database, are meant for storing data gathered from geothermal exploration and production, like various measurement values and relevant location dependent information and metadata. The design of such a database is not dependent on what system (RDBMS) will consequently be chosen for its use. ÍSOR‘s database has partially been designed as a relational for two decades or so, but it is being revised in the latest years to make it relational as a whole. The recent design of the Nicaraguan database was based on existing parts of ÍSOR’s database and also on some planned and upcoming changes to its structure. Still there are various geothermal data kept in old fashioned "flat files" waiting to enter both the databases of ÍSOR and Nicaragua, and the appropriate relational design for these data is not yet finished. Design of, and changes in, a relational geothermal database cannot be said to be quite finalized at some particular time point since it is an ongoing process as long as the particular database is "alive" and in use.

Both these databases are, as are relational databases in general, designed with groups of tables made of columns and rows and constraints and relationships defined in and between the tables. Special kind of information is held in one table while various other related data are held in separate tables, all of which are linked together by primary and foreign keys, often named ID columns (Gunnarsdóttir, 2013a). In principle the database designers and the users should do as little as possible of repeating data between tables, because doing so and keeping unnecessary data in a relational database is against the design rules. As an example of this, location information is held in a special table whilst information regarding sampling from the locations is held in another. These two tables are linked together by their primary and foreign keys.

The relational structure can be clarified by a simple example of two tables in the database. Tables for locations and collected liquid and gas samples at various locations in a relational database could look somehow like the following ones.

Table 1: Example of part of the contents of a table for locations in a relational database.

LocationID	LocationTypeID	AreaID	Name	X	Y	Z
2317	1	6	MT-**	550338.0	1370271.0	106.0
2319	1	6	MT-**	549096.0	1370355.0	67.0
2320	1	6	MT-**	549888.0	1370061.0	94.0

Table 2: Example of part of the contents of a table for samples collected from geothermal wells or spots in a relational database.

SampleID	LocationID	AgentID	DateTime	Initials	Depth	Temperature	Pressure	Wellhead pressure	Discharge
22	2317	1	16.4.2000 16:00	SG		93			11
23	2317	1	5.2.2000 14:00	FÓ		93			10
24	2317	1	5.2.2000 14:20	ThF		2			2

In the table for collected liquid and gas samples, which is linked to the table for locations, the records only need to keep the ID number of the location concerned. So when some location information has to be changed, e.g. name, only one correction has to be made (updating of one record) in the table for locations and no further updating is needed in other tables related to that location. But in an old fashioned flat file database, wherein a sample table would have included e.g. the location name in every single record representing that particular location, many corrections of sample records would have been the case (updating of many records), and probably also in many more tables storing data representing the location.

4.2 Database Structure, Tables and Relations

Tables or table groups in relational geothermal databases can vary, depending on the kinds of data that are to be kept within a particular database in the various organizations, companies or countries. A database for data gathered from geothermal exploration and production, when fully deployed, will include at least around 40 to 50 tables and probably many more. The table for geothermal areas, the *Area* table, and the related table for the locations within these areas, the *Location* table, are the ones that most of the other tables or table groups in the database are linked to, see the main tables diagram in Figure 1.

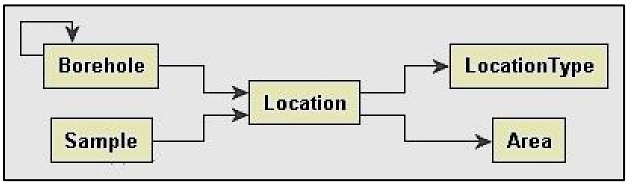

Figure 1: A relational diagram showing the main tables of the database along with the table for location types.

The tables *Area* and *Location* hold the main information about each and every area and location which have data related to them in the geothermal database. It is also possible to have tables for some kinds of subareas, like areas of high temperature, low temperature, power plant areas, preserved areas, etc. Then there are two additional so-called main tables depending on the *Location* table, the *Sample* table and the *Borehole* (well) table. It is also possible to have more specific tables attached to *Location*, like tables for hot springs, mud pools, etc. The two main groups of tables depending on each of the tables *Borehole* and *Sample*, are often spoken of as the "borehole tables group" and the "chemical tables group", although these groups are not quite separated regarding their data. Most of the tables in the so-called borehole tables group store information that is not contingent on the chemical data, and therefore often referred to as the borehole part of the database. Samples of geothermal fluids are of course often collected from boreholes (and then earmarked to the location of the borehole concerned) as they are also collected from many other types of locations, and thus boreholes have much data related to them in the so-called chemical part of the database (Gunnarsdóttir et al., 2012a). Following are two relational table diagrams in Figure 2 and Figure 3, which show parts of the database structure.

In the borehole part the main table is *Borehole*. Depending on it are many other tables storing information regarding boreholes, like *BoreholeHistory* for information of the "life span" of a borehole, *BoreholePath* tables with measured and calculated coordinates of the path down hole for directional boreholes, *Borehole Drilling* and *BoreholeCasing* tables, and *BoreholeTask* and *BoreholeData* for storing the various information on measurements performed in the boreholes and the values concerned (Gunnarsdóttir, 2013b).

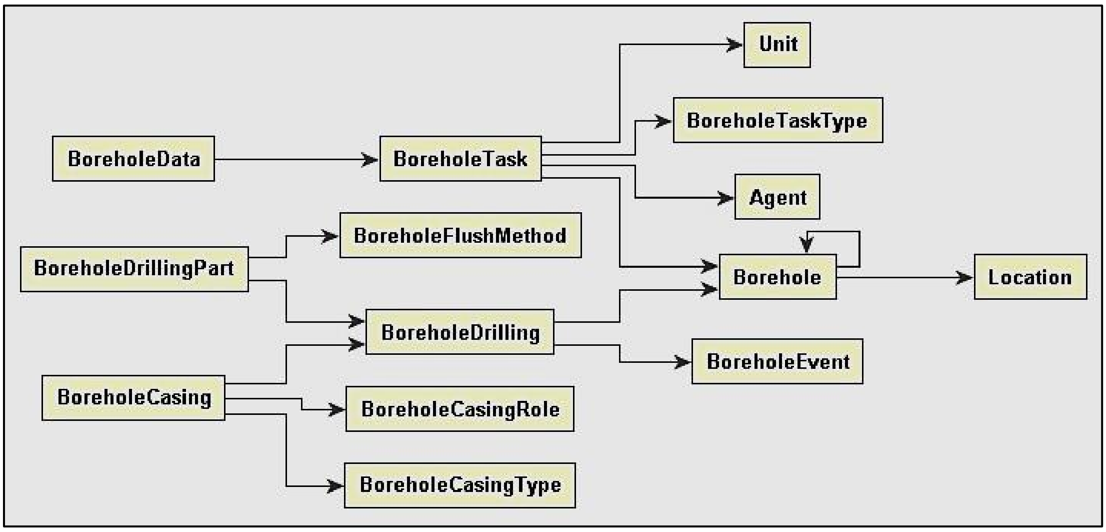

Figure 2: A relational diagram showing some of the table groups in the borehole part of the database.

The main tables in the chemical part are *Sample* and *Analysis*. *Analysis* is linked to *Sample* and then there are also seven so-called peripheral tables to the *Analysis* table, and data inserted into *Analysis* is linked to various information regarding the analysis from these tables: *Precipitation*, *Method*, *Filtration*, *Dilution*, *Component* and *Acidification*. Note that the following chemical table diagram is based on the fact that one sample could be divided up and thus be analysed more than once and even many times and with different working procedures and by different laboratories (Gunnarsdóttir, 2013b).

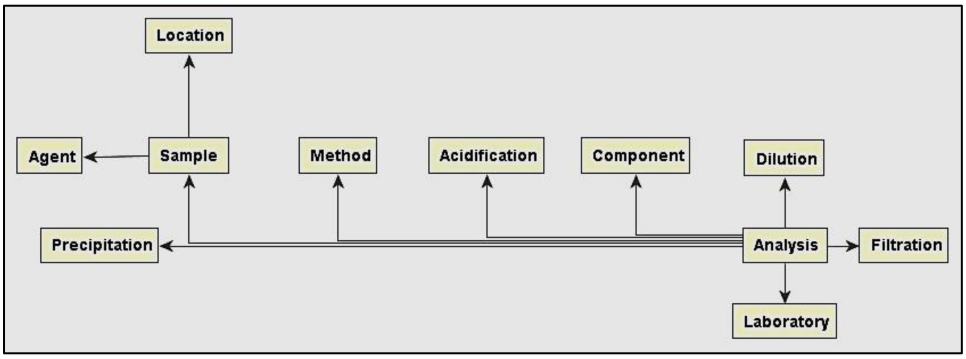

Figure 3: A relational diagram showing database tables regarding collected samples and their analysis.

As explained earlier, a database for data gathered from geothermal exploration and production, when fully deployed, will include at least around 40 to 50 tables and probably many more. The Nicaraguan geothermal database currently comprises around 35 tables (Gunnarsdóttir et al., 2012a). Table 3 shows a list of some of them.

Table 3: Part of the table list of the Nicaraguan geothermal database.

Table	Table comment
Borehole	Table for storing the main information on the various boreholes (wells), their locationID, current depth, if directionally drilled or not, and if they are combined in a fork or not.
BoreholeCondition	Table for storing the main information on the condition of boreholes, data that are regularly obtained with some estimates and quite simple measurements that are maybe not intended to be entered into the BoreholeTask and BoreholeData tables.
BoreholeData	Table for storing data from the various measurements in boreholes, both one shot measurements and series of down hole measurements.
BoreholeEvent	Table for storing a list of possible events that can be ongoing in borehole projects, like the various steps of drilling, utilization, changes (e.g. changing of direction, depth, function, agent, etc.), closure, and more.
BoreholeFunction	Table for storing a list of objectives or functions of boreholes, like production, injection, etc.
BoreholeHistory	Table for storing a list of main aspects of the lifespan of boreholes, like when drilling and the various drilling stages started, the start and end of production, when a borehole is on hold, etc., and when it is finally shut down or filled up.
BoreholePath	Table for storing information regarding paths of directionally drilled boreholes.
BoreholePathMeasured	Table for storing directional (gyro) data for directional boreholes.
BoreholePathCalculated	Table for storing data regarding calculated paths of directionally drilled boreholes.
BoreholeTask	Table for storing the main information on every measurement or measurement series that take place in a borehole.
Sample	Table for storing the main information on collected liquid and gas samples from geothermal sites or boreholes.
Analysis	Table for data resulting from analysis of chemical samples collected from various sites or boreholes.
Component	Table for storing a list of chemical components analyzed, their names and abbreviations or formulas and units of measure.
Laboratory	Table for storing a list of laboratories which are available for analysis, including field laboratory used for analyzing collected samples in the field or on site.
Method	Table for storing a list of analytical methods.
Acidification	Table for storing a simple list of acidification options in water sampling.
Dilution	Table for storing a simple list of dilution options in water sampling.
Filtration	Table for storing a simple list of filtration options in water sampling.
Precipitation	Table for storing a list of precipitations options in water sampling.
Agent	Table for storing a list of institutions, organizations or companies, persons or employees in charge of what is recorded in many of the tables in the database.
Unit	Table for storing and maintaining a list of all the units of the various measurements data in the database, their abbreviations and descriptions.
Reference	Table for storing information on useful literature, maps, and web sites of interest reg. the work.

Excluded from this list in Table 3 are some tables with descriptive names or which contents or part of contents are shown in a few examples here following in Section 4.3.

4.3 Some Descriptive Examples of Tables Contents

Note that the following examples are not to be looked at as complete contents of these tables.

Table 4: ***LocationType table: part of current contents.***

BoreholeTypeID	Name	Description
0	Unknown type of location	When location data are from old reports and no information regarding type of location and not possible to check
1	Borehole	Any kind of a drilled borehole
4	Hot spring or warm well	
5	Fumarole	
6	Mud pool	
11	Pipeline	

Table 5: ***BoreholeEvent table: part of current contents.***

BoreholeEventID	Name	Description
1	Assign - Function of borehole	Assign a function of the borehole
2	Assign - Agent of drilling	Assign agent of the drilling
3	Assign - Agent of borehole	Assign agent of the borehole
4	Drilling - Started	
5	Drilling - Rig up	Drill prepared for use
6	Drilling - Rig down	Drill is taken down
7	Drilling - Part 1	Surface Casing
8	Drilling - Part 2	Casing 2
9	Drilling - Part 3	Drilling Production Part
12	Change - Sidetrack	
19	Testing - Well Stimulation	
22	Maintenance - Acid Stimulation	

Table 6: ***BoreholeFunction table: part of current contents.***

BoreholeFunctionID	Name	Description
1	Production well	The borehole is used to obtain steam or water for use in a geothermal power plant
2	Injection well	The borehole is used to inject fluid back into the geothermal reservoir
3	Monitoring well	The primary function of the borehole is for observations reg. condition of reservoir
4	Gradient well	Shallow well used for geothermal gradient studies

Table 7: ***BoreholeTaskType table: part of current contents.***

BoreholeTaskTypeID	Name	Description
1	Wireline for Warming Up	
2	Wireline for Testing	
3	Production Testing	
4	Well Head Temperature	
5	Well Head Pressure	
6	Monitoring Wireline	With Kuster
8	Flow of Steam	Flow of steam from a borehole
9	Flow of Water	Flow of water from a borehole
11	Enthalpy	
12	Temperature	Temperature log
13	Pressure	Pressure log

4.4 Table Views for Convenience and Calculations

In a relational database table views can be built in for convenience of representing data in various output forms and for calculations of values or group of values. Views are usually virtual tables and are in fact queries to the database that are stored in the system, which retrieve data from a specified table or tables. Views make it possible to represent complicated tables in a more convenient way. They can also be used for security, by restricting user access to specific subsets of tables. The main difference between a view made by a database administrator (DBA) and a query made by a user is that views are stored in the database system, and underlying tables can be indexed to enhance performance. Views take up very little storage space since the database contains only their definition but not a copy of all the data the views represent. Various table views were built into the Nicaraguan database regarding locations, boreholes, history of boreholes, forked boreholes, collected liquid and gas samples and their analysis, calculated ion and mass balances based on particular analyzed components, and more (Gunnarsdóttir et al., 2012a).

4.4.1 Views for Convenient Data Display – Listing from One to Several Tables

Table views are often made for displaying data from more than one table at a time. Table 8 shows one example of such a view.

Table 8: ***BoreholeHistory view: an example of part of BoreholeHistory contents along with data from the tables Agent, Location, BoreholeEvent, and BoreholeFunction.***

Bore holeID	Name	Location ID	Bore hole EventID	Event name	Date	Borehole Function ID set	Function name	Agent ID set	Agent name
9	MT-**	2317	4	Drilling - Started	13.11.1974				
9	MT-**	2317	2	Assign - Agent of drilling	13.11.1974			4	EN** – Or**
9	MT-**	2317	10	Drilling - Finished	27.1.1975				
9	MT-**	2317	1	Assign - Function of borehole	27.1.1975	6	Abandoned well		

4.4.2 Views for Calculations – Exploiting the SQL Data Manipulation Facilities

The table views for mass and ion balance calculations based on values of certain analyzed components of liquid samples can be mentioned as a good example of what can be hardwired into a relational database, and should preferably be therein, since there it is observable, step by step, how these balances are calculated. To have these calculations as an internal part of the database has advantages compared to making use of mass and ion balance calculations programmed into some specially designed interface or in-built in other standard software tools for interacting with databases. The SQL query language has data manipulation facilities beyond many other computer languages, and makes use of the many in-built functions of the relational system, like sum, average, max, min, count, power, etc. By having these mass and ion balance views built into the database using SQL it is easy to show not only the steps and final outcome of the calculations, but also what the calculations are based on, like number of analyzed components, if certain components are yet analyzed or not, etc., see Table 9 and Table 10. The syntax of the views can be viewed any time by the database users. Table views for mass balance calculations are four, and seven for ion balance, whereof each is based on the previous one. However it only takes a fraction of a second to list ion or mass balance for a sample from these views, although their SQL-commands are selecting proper analysis records and also counting and doing complicated calculations on them in some steps. Finally these table views are "alive", that is, they get the result from the database based on every concerned analysis record, old and new, whenever a user is selecting from these views (Gunnarsdóttir et al., 2012a).

Table 9: ***MassbalanceViewFinal: here is an example of what results selected from it might look like.***

sampleID	count_analysed massb_comp	sum values	value TDS	mass balance
28	24	131.2235058	185	**-29.07**
39	25	12974.8579437	12958	**0.13**
36	24	28791.2823824	15400	**86.96**

Table 10: IonbalanceViewFinal: here is an example of what results selected from it might look like.

sampleID	count_ana lysed_ph	count_ana lysed_chloride	count_ana lysed_cations	count_ana lysed_anions	cation rounded	anion rounded	ion balance
35	1	1	11	8	0.00751	0.0091	**-19.21**
39	1	1	12	9	0.21725	0.21273	**2.1**
41	1	1	11	8	0.00305	0.0028	**8.42**

4.5 Main Points when Managing Data

Although relational databases are powerful tools, they of course do not take care of everything by themselves. Caution has to be shown by users of the database when entering or changing data and retrieving data from it, as is true for any other database. It has to be standardized from the start of using the database how various values are inserted into columns in tables, that is which units to use, like coordinates columns, the *x* and *y*, must always be filled in the database according to the standard that has been decided on using, values in columns representing depths regarding boreholes and collected samples, if they shall be in meters or another unit and with how many decimal digits, similar goes for columns for temperature, pressure, and more. Users have to be aware and cautious when retrieving information and data of forked boreholes, if such boreholes exist in the database, since most of these data are representing a whole fork but some represent only one branch of a fork. Otherwise, these data are not representative for the fork as a whole as they must be, since most of the measurements represent the whole fork and it cannot be known from which part of a fork the produced water, gas, steam or condensate comes from. The only occasions where the user would want to insert or retrieve information referring to a specific leg of a forked borehole are when the information is on borehole path, its cuttings or lithology (Gunnarsdóttir et al., 2012a).

When a database for data gathered from geothermal exploration and production is of a good relational design and used correctly the exploitation of these data can be greatly increased. Having various database reports run on a regular basis to make the data well observable from many points of view, as soon as they have been gathered, users will be more aware of the database content. They will be more capable of detecting which data are incorrect and have to be corrected or discarded and which data are qualified for experts to base decisions and theses on. Reviewed and reliable data can be retrieved quickly and with ease and compared on a large or a small scale, from time to time, from within and between areas and reservoirs and within or between individual wells and other locations of measurements or sampling. Information can also be shared more easily, resulting in improved collaboration and thus advancing geothermal work around the world.

5. RELATIONAL DATABASES FOR OVERVIEW OF GEOTHERMAL DEVELOPMENT

ARGeo (the African Rift Geothermal Development Facility) is a programme being deployed by UNEP (the United Nations Environment Programme) for promoting development of geothermal resources in East Africa. ICEIDA (Icelandic International Development Agency) is one of the partners and co-financiers of the ARGeo programme. A special database for overview of African geothermal was developed at ÍSOR from 2010 to 2013. It was based on previous work performed by ÍSOR for ICEIDA on compilation of geothermal information from the east African region. The main purpose of this database is to provide geothermal related information about African countries to both private and public entities that are or will be involved in the geothermal development and investment. It can thus serve to promote the development of geothermal energy in Africa by storing and making available all scattered information. It is expected to contribute to more focused projects based on up to date knowledge about the present status. A database like this could also step up optimization of resource use by cooperation among the countries concerned and pooling existing resources (ÍSOR, 2012). The name of the database is *ARGeo Geothermal Inventory Database* (AGID). The users in charge of entering data into this database, for the African countries concerned, started to enter data into it early in the year 2014. A web-interface was developed, since the various users in charge of entering and updating the data are working in many countries, and also because this database will in the future allow open access for the public to view its contents. A database like ARGeo‘s, along with its special web-interface, could easily be taken into use in some other geothermal area of the world, with some changes and local adaption made to both.

5.1 Design Purpose

This database was designed to provide an overview of geothermal development in Africa, and is thus made for storing information on geothermal projects, their funding and progress, related information on human resources capacity and the various equipment available or required in the countries concerned. This database is not designed to store geothermal research or production data, in particular not data such as the results of physical or chemical measurements. The structure of this database is quite simple and it is easy to use (Gunnarsdóttir, 2011).

5.2 Database Structure, Tables and Relations

The main tables of this database are *Country*, *Project*, *Funding*, *Site*, *Stakeholder*, *Manpower*, *Equipment* and *Progress*. Up to half of the tables of this database are "independent" in the sense that each of them can function without the need to reference data from any other table. Two of these independent tables, *Country* and *Project*, are rather central to the database and its function and are the first to receive data. Some tables may be called peripheral, although each of these latter is connected to another table, and these keep lists like jobcategories, maptypes, labtypes, etc. About half of the database tables are "dependent" tables, i.e. tables that need to reference data in other tables, like *Progress*, *Manpower*, *Equipment*, *Laboratory*, etc.

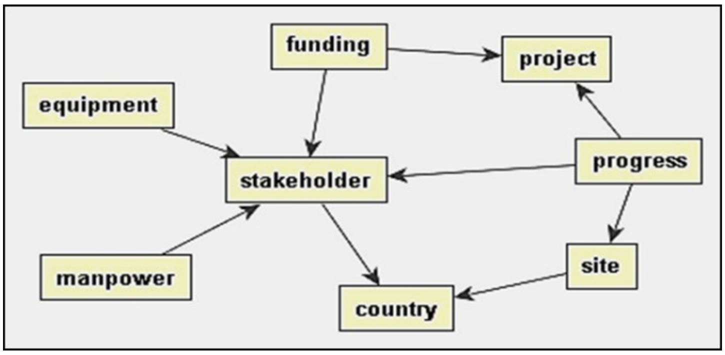

Figure 4. A relational diagram showing the main tables of the AGID database.

Tables that are named after two other tables, serve to associate items in the two, like the *Country_Project* table serves to maintain a list of projects in each country. The database currently comprises around 20 tables, listed in Table 11.

Table 11: The tables in the AGID database.

Table	Table comment
Borehole	Table for storing the main information on the various boreholes (wells), their locationID, current depth, if directionally drilled or not, and if they are combined in a fork or not.
BoreholeCondition	Table for storing the main information on the condition of boreholes, data that are regularly obtained with some estimates and quite simple measurements that are maybe not intended to be entered into the BoreholeTask and BoreholeData tables.

BoreholeData	Table for storing data from the various measurements in boreholes, both one shot measurements and series of down hole measurements.
BoreholeEvent	Table for storing a list of possible events that can be ongoing in borehole projects, like the various steps of drilling, utilization, changes (e.g. changing of direction, depth, function, agent, etc.), closure, and more.
BoreholeFunction	Table for storing a list of objectives or functions of boreholes, like production, injection, etc.
BoreholeHistory	Table for storing a list of main aspects of the lifespan of boreholes, like when drilling and the various drilling stages started, the start and end of production, when a borehole is on hold, etc., and when it is finally shut down or filled up.
BoreholePath	Table for storing information regarding paths of directionally drilled boreholes.
BoreholePathMeasured	Table for storing directional (gyro) data for directional boreholes.
BoreholePathCalculated	Table for storing data regarding calculated paths of directionally drilled boreholes.
BoreholeTask	Table for storing the main information on every measurement or measurement series that take place in a borehole.
Sample	Table for storing the main information on samples collected from geothermal sites or boreholes.
Analysis	Table for data resulting from analysis of chemical samples collected from sites or boreholes.
Component	Table for storing a list of chemical components analyzed, their names and abbreviations or formulas and units of measure.
Laboratory	Table for storing a list of laboratories which are available for analysis, including field laboratory used for analyzing samples in the field or on site.
Method	Table for storing a list of analytical methods.
Acidification	Table for storing a simple list of acidification options in water sampling.
Dilution	Table for storing a simple list of dilution options in water sampling.
Filtration	Table for storing a simple list of filtration options in water sampling.
Precipitation	Table for storing a list of precipitations options in water sampling.
Agent	Table for storing a list of institutions, organizations or companies, persons or employees in charge of what is recorded in many of the tables in the database.
Unit	Table for storing and maintaining a list of all the units of the various measurements data in the database, their abbreviations and descriptions.
Reference	Table for storing information on useful literature, maps, and web sites of interest reg. the work.

5.3 Main Rules when Managing Data

Regarding entering data into many of the tables, there are some very important rules, which must be followed so the database can be used properly and the users in charge must be fully aware of these rules. The rules dictate how records are to be entered into some tables, how some column contents have to be of standardized units, how a site must be restricted to one country in the database, and more (Gunnarsdóttir, 2014).

The ***Country*** table holds a list of all countries involved in geothermal work in Africa, one record for each country and the African countries in the ARGeo developmental work can be specially marked herein, if needed to do so. The ***Project*** table stores a list of all projects, one record for each with a unique country name, and main information regarding a project. Both older and ongoing projects can be inserted into this table, and also upcoming or expected projects. The ***Funding*** table stores information about the funding of geothermal projects. The actual status of the funds is immaterial, however. Thus the table may hold information on funds already appropriated, granted, or agreed, or simply funds that are expected. The ***Site*** table is for storing a list of all sites in geothermal areas within the countries under consideration. A site may be as large or as small as desired; it can range from a small plot up to an entire country. Because of the structure of the database, however, a single site record cannot refer to an area spanning more than one country. The ***Stakeholder*** table serves to maintain a list, one record for each stakeholder, of all companies, institutions, authorities, governments and funding agencies that take part in the geothermal development work in the relevant world area. A stakeholder may even be an entire country, but such a "country stakeholder" is only defined to permit the attaching of data in other tables, such as *Manpower* or *Equipment* when needed to keep track of what is owned or required on a country basis. The ***Manpower*** table is for storing a specification of all manpower available to the geothermal development projects. The ***Equipment*** table is for storing information about various types of equipment. The ***Progress*** table is of particular importance. It is for keeping track of completed, ongoing, and planned geothermal development work on each particular project of a specific stakeholder at some given site (Gunnarsdóttir, 2011).

5.4 AGID Data Collected

In the design and testing phase data was entered into the database from information gathered by Gestur Gíslason and Knútur Árnason in Africa 2005 (pending publication), and also more recent information “... *gathered under supervision of the African Union Commission in their work to assist the geothermal initiatives taking place in Africa, including ARGeo and the Geothermal Risk Mitigation Facility“* (Árnason et al., 2011, p. 9–10). Thus a moderate amount of data was inserted into the database during the design and testing phase, especially regarding information of equipment and manpower. These data were updated and new data inserted by the ARGeo staff in cooperation with their focal points in the concerned African countries early 2014.

5.5 Some Output Examples

Besides inserting the previously mentioned data into the database during the design and testing phase, some programming was also required to take useful outputs from the database content. These outputs and many more were then developed in the AGID web-interface by its designer Mr. Jón Ragnarsson and the project manager Dr. Árni Ragnarsson. These include both queries from single tables and from various combinations of tables. Following are some parts of these output lists from the web-interface, in Table 12 to Table 16. One output of a special database report, in Table 17, is showing a part of a list of equipment that is required, owned, or both, compared between the countries concerned (this report is not yet part of the web-interface).

Figure 5. A map of Kenya from the web-interface of the AGID database.

Table 12*: Site: part of the site list for Kenya – to be looked at here as test data.*

Name	Estimated Power	Existing Power	Reconnaissance	Surface exploration	Drilling
Eburru	60		Yes	Yes	Yes
Longonot	700		Yes	Yes	No
Menengai	1000		Yes	Yes	Yes
Silali	1000		Yes	Yes	No

Table 13: Stakeholders Funding Projects: part of the list for Kenya – *to be looked at here as test data*.

Name	Workbase	Status	Projects
The International Development Association	USA and many places worldwide	Active	Geothermal Exploration Project Emergency Power Supply Project
German Federal Ministry for Economic Cooperation and Development	Germany	Planned	GEOTHERM (BGR) Olkaria II Olkaria IV
United Nations Development Programme	USA and many places worldwide	Active	GEOTHERM (BGR)

Table 14: Projects: part of the list for Kenya – *to be looked at here as test data*.

Name	Starts	Ends	Remark	Estimated Cost	Final Cost	Status
GEOTHERM (BGR)			Phase I 2003-2009Phase II 2010-2012			Active
Olkaria IV			Financial closure expected in first quarter of 2011.			Active
AUC-KfW Geothermal Risk Mitigation Facility	2013-10-01	2017-01-01				Active
ICEIDA-NDF Regional Geothermal Project	2013-12-20	2015-12-31				Active

Table 15: Equipment: part of the list for Kenya – *to be looked at here as test data*.

Equipment	Topic	Condition	Availability
TEM transmitter	Geophysical exploration	Working	Owned
Well test silencer	Geophysical research	Working	Owned
Topographic maps, scale 1:50.000	Geological mapping	Working	Owned
Polarizing microscope	Petrological lab-research	Working	Owned
Large format scanner	Computer hardware	Working	Owned

Equipment	Topic	Condition	Availability
Digitizer	Computer hardware	Working	Owned
Magnetic coils	Geophysical exploration	Working	Owned
pH meter	Geochemical analysis	Working	Owned

Table 16: Laboratories: part of the list for Kenya – ***to be looked at here as test data*****.**

Name	Type	Agent
KenGen Geochemical laboratory	Geochemical	KenGen
KenGen Geological laboratory	Geological	KenGen
KenGen Geophtsical laboratory	Geophysical	KenGen
KenGen Environmental laboratory	Environmental	KenGen
KenGen GIS laboratory	GIS	KenGen

Table 17: ***A database report output: part of a list of Equipment, required, owned, or both, compared between the countries concerned – to be looked at here as test data.***

Topic	Equipment	Eritrea own	Eritrea req	Ethiopia own	Ethiopia req	Kenya own	Kenya req	Malavi own	Malavi req	Rwanda own	Rwanda req	Tanzania own	Tanzania req
Aerial photographs	Aerial photographs			Own									
Borehole drilling	Drilling rig												Req
Borehole logging	Down hole instruments (wireline or slick-line)		Req	Own	Req						Req		Req
Borehole logging	Logging Truck		Req		Req						Req		Req
Computer software	ArcGIS					Own						Own	
Computer software	Central database system												
Geochemical analysis	Ion chromatograph (IC)		Req	Own							Req	Own	Req
Geochemical research	Webre steam-wate r separator		Req		Req		Req		Req		Req		Req
Geological mapping	Topographic maps , scale 1:50.000					Own						Own	
Geophysical exploration	Gravimeter	Own		Own	Req	Own					Req	Own	Req
Geophysical research	Differential GPS		Req		Req	Own					Req		Req
GPS	GPS Handheld			Own		Own						Own	Req
Meteorological datasampling	Meteorological station		Req		Req				Req		Req		Req

6. CONCLUSIONS

For institutions and companies in the geothermal sector it is highly recommended to take into use some relational database management system (RDBMS) and maintain a database designed to preserve and access their geothermal information. This is especially true for the data gathered from geothermal exploration and production, such as various measurement values and relevant location dependent information and metadata. Those who do not yet have a centralized relational database, but are either using a flat file database, a decentralized database, or even no database at all, should set up such system as soon as possible. The cost of doing so does not have to be high, and it is definitely low compared to what is gained from it.

A database like the one ÍSOR developed for Nicaragua, with or without a special web-interface, could be taken into use in geothermal companies or institutions elsewhere, after some redesign or changes and adaption to local circumstances. The design of such a relational database is quite flexible and it can easily be expanded to store some related information, like financial and operational data from areas, boreholes (wells) and more. A database like AGID, the one ÍSOR developed for ARGeo/ICEIDA, for the overview of geothermal development in Africa, along with its special web-interface could easily be taken into use in some other geothermal areas, again with some changes and local adaption made to both database and web-interface.

The benefits of designing a relational geothermal database for data gathered in geothermal work, among other things, are to ensure data consistency, avoid data duplication, minimize human errors, maximise data security and to increase the accessibility of the data. Such a relational database, properly designed and correctly used, will thus play an essential role in the research work and utilization management of the geothermal resources. It will ensure safe, systematic and centralized storage of the data gathered in the geothermal work, as well as easy access for all.

Having various database reports run on a regular basis to make the data well observable from many points of view, as soon as they have been gathered, users will be more aware of its content. They will be more capable of detecting which data are faulted and have to be corrected or discarded and which data are qualified for experts to base decisions and theses on. Reviewed and reliable data can be retrieved quickly and with ease and compared on a large or a small scale, from time to time, from within and between areas and reservoirs and within or between individual boreholes (wells) and other locations of measurements or sampling. Information can also be shared more easily, resulting in improved collaboration and thus advancing geothermal work around the world.

ACKNOWLEDGEMENTS

The database projects and writing of this paper were supported by Iceland GeoSurvey and The Icelandic International Development Agency (ICEIDA). Samples data are from the geothermal division of Nicaraguan Ministry of Energy and Mines (MEM) and ARGeo's AGID database. The author would like to thank the following colleagues at Iceland GeoSurvey: Dr. Jón Örn Bjarnason for commentary on his original relational design of the chemical part of ÍSOR's geothermal database; Dr. Þráinn Friðriksson and Mr. Finnbogi Óskarsson for advice on the design of the recent additions to the database chemical part; Dr. Sigvaldi Thordarson for the teamwork in redesigning the database borehole part; Dr. Árni Ragnarsson and Mr. Jón Ragnarsson for their part in refining the design of the AGID database; Dr. Guðni Axelsson for reading through this paper.

REFERENCES

Árnason, K., Gíslason, G., Ásmundsson, R. and Gunnarsdóttir.S.: *Inventory of Equipment and Manpower for Geothermal Work along the Great African Rift Zone. Potential for Cross-Border Collaboration and Utilization of Existing Facilities and Knowledge in the ARGeo Countries of Djibouti, Eritrea, Ethiopia, Kenya, Tanzania and Uganda with the Addition of Rwanda and Malawi*. Iceland GeoSurvey, ÍSOR-2011/030. 71 p . (2011).

Ásmundsson, R., Ragnarsson, Á. and Gunnarsdóttir, S.: *Geothermal Development in East Africa – Site List and Database. Development Aid and Activity in the East African Nations of Burundi, Djibouti, Eritrea, Ethiopia, Kenya, Malawi, Mozambique, Rwanda, Tanzania, Uganda and Zambia with a List of Institutions and Companies Involved in Geothermal Work*. Iceland GeoSurvey, ÍSOR-2011/019. 63 p. (2011).

Gunnarsdóttir, S.: *IDAG – Inventory Database of African Geothermal. Design Outline and Directions for Use*. Iceland GeoSurvey, ÍSOR-2011/070. 169 p. (2011).

Gunnarsdóttir, S., Ragnarsson, J. and Pentzke, J. F.: *A Relational Geothermal Database and Web Interface for the Ministry of Energy and Mines in Nicaragua. Design Outline and Directions for Use*. Iceland GeoSurvey, ÍSOR-2012/063. 96 p. (2012a).

Gunnarsdóttir, S.: *Database Teaching Material – Geothermal Database of MEM – Ministry of Energy and Mines Nicaragua.* Iceland GeoSurvey, short report, ÍSOR-12079. 88 p. (2012b).

Gunnarsdóttir, S.: *Basics of Modern Databases – Mainly the Relational One*. Iceland GeoSurvey, unpublished short report. 15 p. (2013a).

Gunnarsdóttir, S.: *A Relational Geothermal Database – Overview for UNU Students at Geothermal Training Programme with some SQL exercises*. Iceland GeoSurvey, unpublished short report. 31 p. (2013b).

Gunnarsdóttir, S.: *AGID Database – Guidelines for Use and Principles of Managing Data*. Iceland GeoSurvey, unpublished short report. 4 p. (2014).

Holowczak, R.: *What is SQL, and what are some example statements for retrieving data from a table?* Zicklin School of Business – Barugh College City University of New York. See http://cisnet.baruch.cuny.edu/holowczak/classes/3400/sql/ (last modified on October 13, 2011; last viewed 30th of May 2014) (2011).

Indiana University Information Technology Services Knowledge Base: *What is SQL, and what are some example statements for retrieving data from a table?* See https://kb.iu.edu/data/ahux.html (last modified on January 10, 2013; last viewed 30th of May 2014) (2013).

Iceland GeoSurvey: *Annual report 2012.* 43 p. (2012).

Monash, C. A.: *The Explosion in DBMS Choice. Database options in a cost conscious world*. See http://www.monash.com/uploads/explosion-database-choice.pdf (last viewed 30th of May 2014) (2008).

The University of Vermont: *Charles Lyell*. See http://www.uvm.edu/perkins/evolution/darwin/?Page=players/lyell.html&SM=players/playersmenu.html (last modified on October 06, 2008; last viewed 30th of May 2014) (2008).

Woodard, Linda *Relational Databases Not your Father's Flat Files*. See http://www.cac.cornell.edu/education/Training/DataAnalysis/RelationalDatabases.pdf (last viewed 30th of May 2014) (2008).

Wikipedia: *Relational database*. See http://en.wikipedia.org/wiki/Relational_database (last modified on 17 April 2014; last viewed 30th of May 2014;