DataBase Teaching tools

Keith Moss
Mathematics Computing and Technology
The Open University
Milton Keynes
k.e.moss@ieee.org

Abstract— **This paper is about how to improve the experience of students of the subject of database theory and practice, a subject many find somewhat daunting. It is by experiment that most scientists learn most about their discipline and in this paper the following ideas can be explored in simulations that in most cases use access to a database so that the exercises are personalised.**

The simulations are listed below.

- **Simple Entity-Relationship diagrams: Where simple Entity-Relationship diagrams can be composed from a set of components and their validity checked.**

- **Entity-Relationships and tables that describe the entities: Where the student has to decide from tables of two entities the degree of participation and the participation conditions. With prompting the correct solution can be found and as many attempts as necessary are possible.**

- **Structured Query Language (SQL): Given a database various SQL queries can input and the way that the queries are dealt with are shown in detail not just the final result.**

- **Functional dependency: Given a table from a database this can used to find out whether there is any functional dependency between any of the columns and also to show what the primary key/s might be.**

- **Normalisation: A topic that confuses may students. This allows table to be split into a number of sub-tables and the result can be tested as to whether the selected table can be joined together to form the original table.**

Each of these ideas has a simulation and these will be discussed in terms of how to run the simulation and explaining what the simulation is trying to teach. There are some third party software requirements and these will be explained together with the solution that is used for the demonstrations.

Keywords-Entity Relationship diagrams, participation conditions, multiplicity, Structured Query language, Tables, Normalisation, Functional dependency

I. INTRODUCTION

The tools are designed to be used with any database as they have a configuration files that are used to specify the ODBC-JDBC connection. There are also data files that can be used to store queries that have been used and have been shown to produce correct results. These data files are necessarily different for the various applications. For demonstration purposes a number of small databases files have been constructed that are typical database applications, one concerns a mythical Hospital and another, an online University.

To allow for connection a number of databases a configuration file is used that contains the details of the connection request required by the databases. The purpose of introducing a number of databases is to show that the software, written in java, is generic and can thus be customised by the user without any need to change the software. The configuration file format for the Open University course [7] which uses SyBase is shown below:

```
Hospital;jdbc:odbc:HospitalDSN;m359;m359
University;jdbc:odbc:UniversityDSN;m359;m359
```

Similar formats are used for the corresponding Excel and Access files.

```
University_Excel;jdbc:odbc:university;"";""
Hospital_Excel;jdbc:odbc:hospital;"";""

Hospital_Access;jdbc:Access:///C:/Netbeans_M359/Hospital.
mdb;"";""
Univesity_Access;jdbc:Access:///C:/Netbeans_M359/Universi
ty.mdb;"";""
```

The separator used in the Java simulation program is ";", the semi-colon and hence its use in constructing the entries in the configuration file

II. BACKGROUND

The idea of an Entity-Relationship model dates from 1976 at which time the paper by Peter Chen [1] proposed unifying the models that had been used prior to that, the network model, the relational model and entity set model to that of the Entity-Relationship model. Appendix 1 shows instances of each of these derived from Chen's paper. The idea of mandatory and optional participation does not appear to be supported at this time.

It seems that it took academia another six years before there was any publicised interest in applying these ideas to the teaching of database technology. Carol Chrisman [2]

introduced the idea of using entity relationship models as a tool in the teaching of database design using the entity relationship approach. The notation used is that of Chen.

Then in 1987 another author Judith D. Wilson [3] takes up the baton and analyses some of the problems associated with how students use entity-relationship model and how to circumvent them.

An interesting contribution as regards using computers to help in learning SQL came in 2004 from tutors at the University of Queensland, Australia [4]. It was called SQLator and enabled students to check the correctness of a query before submitting it to a database. It was however limited to this one application and did not concern itself with entity relationships.

In 2007 Edward Sciore [5] introduced a system that dealt again with SQL but this was concerned with how an SQL query was processed in the computer that hosted the application.

In 2008 there was another SQL query simulation [6] that dealt with a range of queries that would be used in the construction and testing of a database design, but again notice limited to this one area. This paper was as much interested in how the system dealt with the query as to how structure the query

III. ENTITY-RELATIONSHIP DIAGRAMS

A. The Graphical User Interface (GUI)

This is the only tool that is based entirely on simulation. At this stage of learning in a database course it is only concepts that need to be learnt. The tool can be personalised using component names that are part of the course and also Chen's notation or Crows foot notation can be used As can be seen from Figure 1, the screen shot of the simulation, there are four text areas and a number of graphical symbols. The two entities are shown together with the relationship that exists between them. The purpose of the exercise is place two of the participation conditions at either end of the relationship and for information about the choice to be displayed in the text areas.

In some cases the choice is a combination where a simple relationship is not possible but has to be represented by a Relation for Relationship instead. The screen is slightly redrawn with another entity in the centre and relationships connecting all three entities. An explanation to this effect is given in the top text area, it is hoped that this is sufficiently detailed for the student to continue with what can be a difficult choice.

B. Information files

There are two sets of files relating to the text areas they have been called Explanations and Relations. For some users the explanations may need to be expanded and this is obviously possible as they are only text files. There are however some twelve files that would have to be changed for it to be complete

a. Explanations

These describe how the two entities are related in terms of participation conditions that have been chosen they appear in the top text area. An example where a RELATION FOR RELATIONSHIP has needed to be used

"Entities A and B have discretionary participation as is shown. But now the 'many' relationship of entity B to entity A is replaced by the 'many' relationship of the entity RELATIONSHIP with entity A. Both participation conditions for the entity RELATIONSHIP are of necessity mandatory"

b. Relations

These set out the entity descriptions in a table format as would be used when starting the potential design of a new database. The student then gets used to this type of presentation.

"Relation RELATIONSHIP
A1: A1s
B1: B1s
primary key A1, B1
foreign key B1 references A.
foreign key A1 references B."

An example of the text in the lower middle text area that is included when a RELATION FOR Relationship has needed to be used.

Foreign and primary keys are shown, as would be the case, in the other two lower text areas.

C. The exercise

The student is presented with the initial screen and has some instructions that explain the meaning of participation conditions and multiplicity and is encouraged to see how the entity table for each entity is set out in the text areas above each of the entities. Any combination of symbols is possible and the result for each should be noted. When those choices that give rise for the need of a Relation for Relationship are chosen, the student would be encouraged to note what these were but not pursue it but continue without completing that choice until all choices had been tested.

The pairs of symbols that do not require a Relation for Relationship are now known and these can be used to try and satisfy the situation where symbols have to be attached to all the three entities that are shown. The first text area gives some clue as to what might be an appropriate choice.

Figure 2 shows a completed exercise where the initial selection of participation conditions was optional and both ends had single occurrences. An explanation of why this representation is required is given in Appendix 2

D. The lessons learnt

The student should have now some understanding of how relations are expressed in the definition tables. How the domains are shown and how the primary and secondary keys are shown. There should also be some understanding of what is meant by a foreign key and hopefully why they are needed.

The relation for relationship idea will need further explanation but at least the idea will have been firmly entrenched.

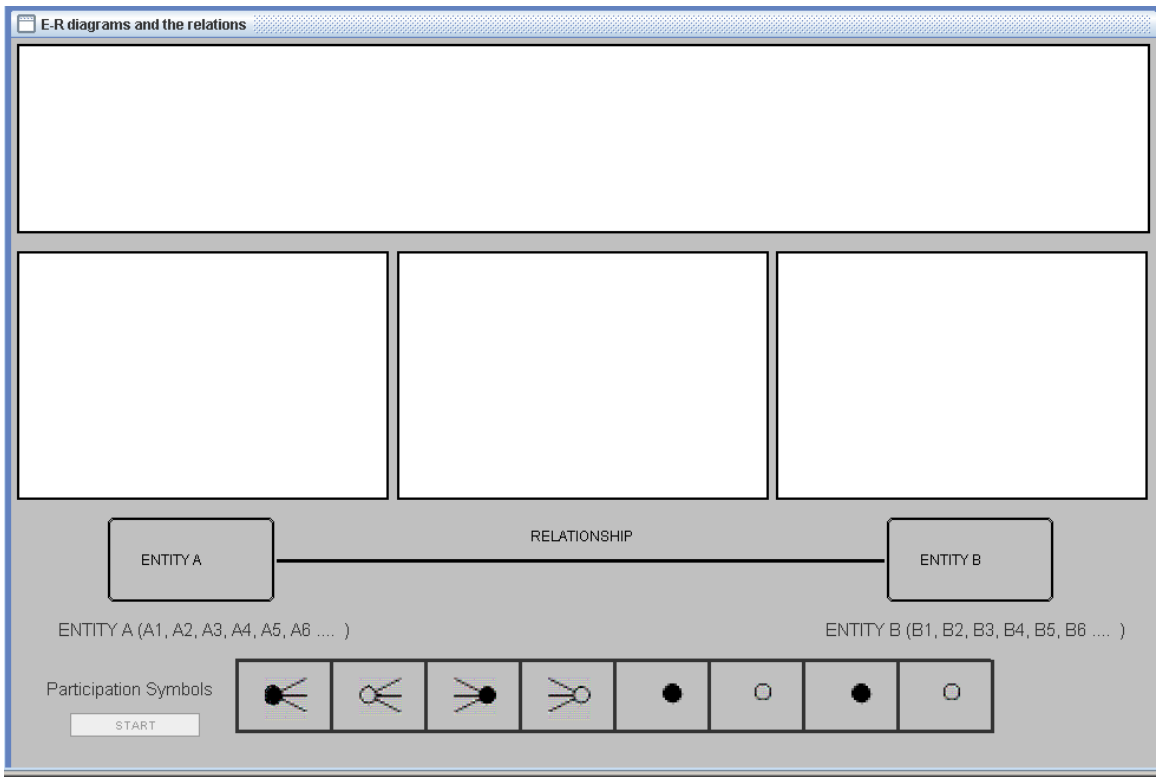


Figure 1. The initial screen for Entity Relationship diagrams.

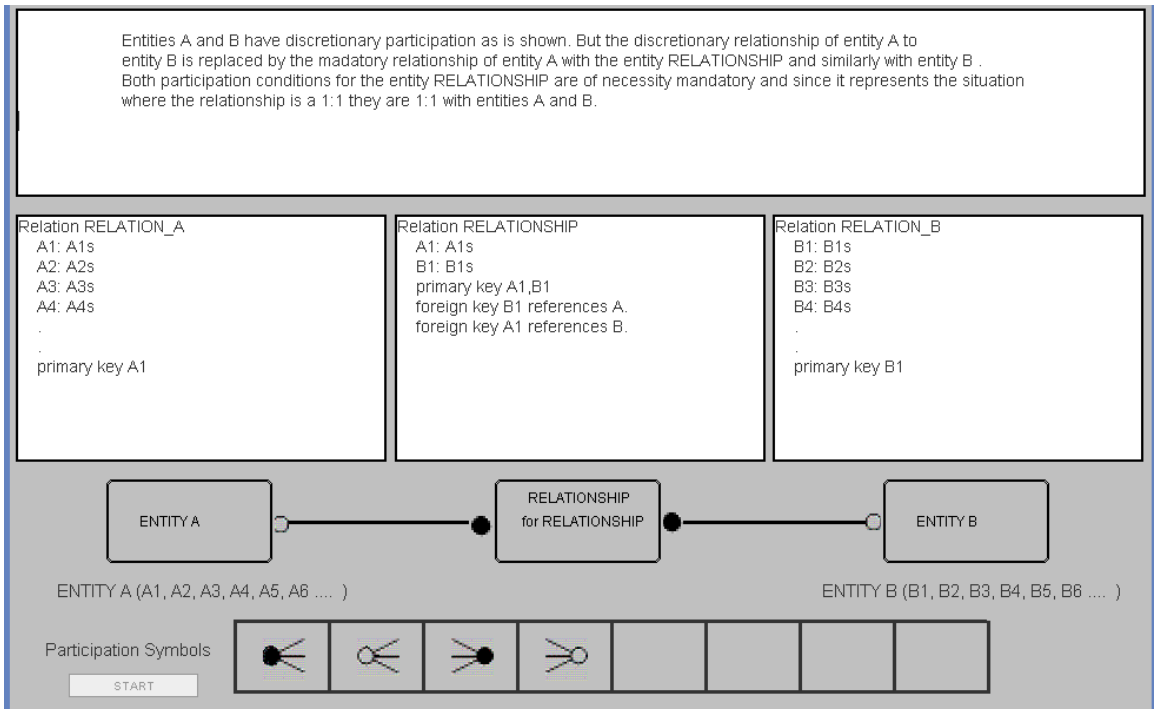


Figure 2Tables and ER diagrams that correspond to them

IV. ENTITY-RELATIONSHIPS AND TABLES

A. *The graphical user interface (GUI).*

There are three drop down menus, one to choose the database and two that choose individual tables from that database. The tables are displayed in their entirety once they have both been chosen. Two panes display the tables selected and the progress of the exercise.

For this tool it was decided to use a database for the data that was needed as this the simplest way of ensuring that the tool was general purpose and could be used on any database

Figure 3 shows the screen after a student has completed an exercise successfully. The same drawing and symbols are used as for the entity-relationship diagrams.

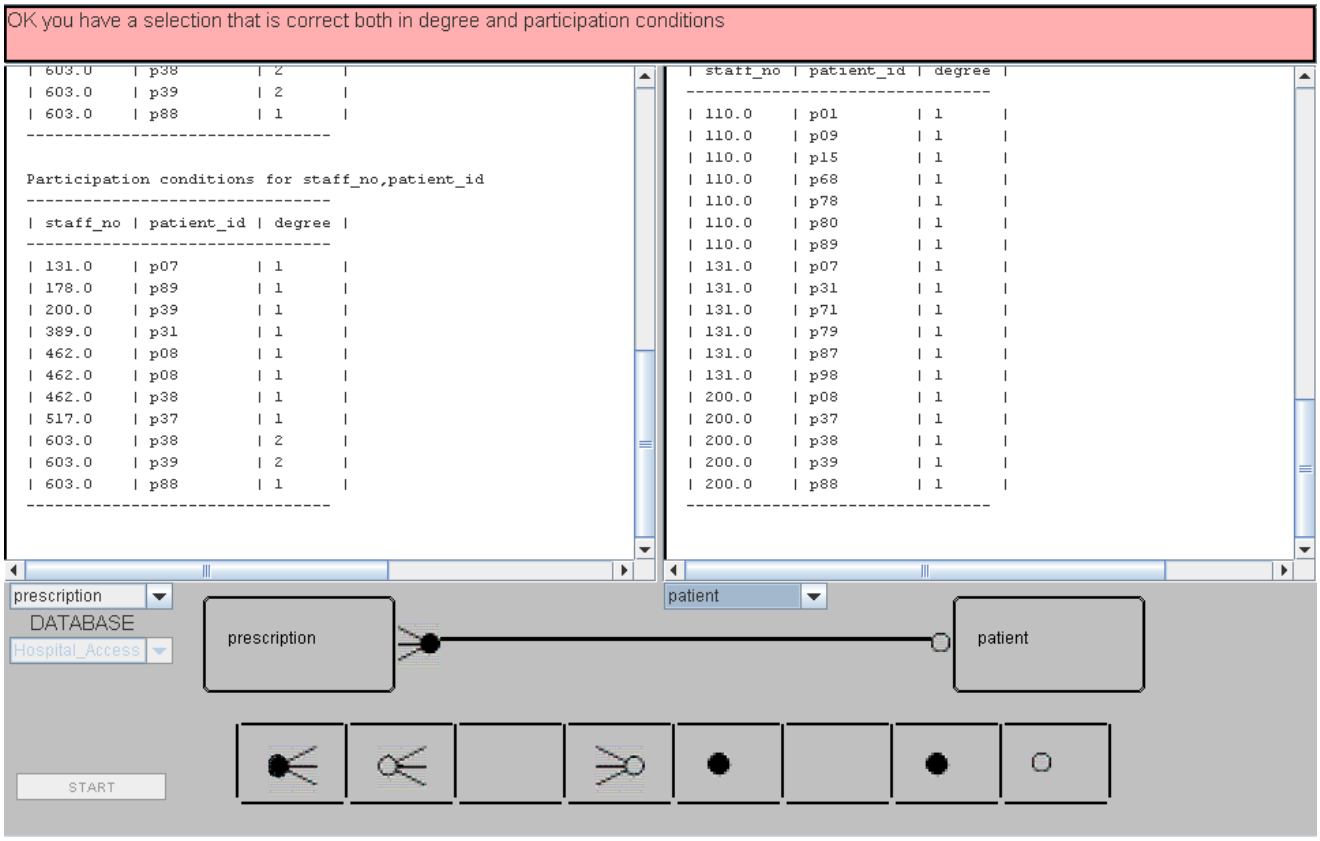


Figure 3Tables and ER diagrams that correspond to them

B. *The exercise*

Initially no database is selected and no tables. The top drop down menu on the left-hand side has a choice of database.

Some of the databases that are used relate to an Excel files some to Access files and some to Sybase files although in fact they contain the same data. When the database has been selected the other two drop down menus are available and tables form the chosen data base are listed. A choice has to be made for each menu and the table for each is displayed.

The student has to study the table and decide upon the participation conditions in degree and whether it is discretionary or mandatory. The symbols are chosen to reflect their decisions and placed on the ER-diagram. Whether this has been success or otherwise is shown in the upper text area with messages such as:

"OK you have a selection that is correct in degree but not participation conditions change the conditions" OR perhaps

"Wrong symbols chosen use the database output to make a better informed choice of symbols".

The progress of the exercise is thus apparent and only complete when the student gets the message *"OK you have a selection that is correct both in degree and participation conditions***".**

The first message means that the multiplicity of one or both the symbols needs correcting but that the participation conditions whether they should be shown as mandatory or

discretionary has been achieved. The second message indicates that neither of the choices made is appropriate. In either case the table content is condensed so that for each entry in each table the degree of participation is shown and this makes the choices more apparent.

Notice that after the tables have been selected and the student has made an initial choice of symbols that screen is updated to show the participation conditions in degree for the common key between the two tables. This is helpful especially in the situation where the tables a long.

C. The lessons learnt

The strongest lesson here is the fact that trying to appreciate a mass of data can be a daunting exercise, even for the relatively small tables that are available, and that reducing the size of the problem by looking only at the primary key and the multiplicity of the key it is then possible to analyse the data visually more easily. Also it should be possible for the ideas mandatory and discretionary participation conditions to become clearer.

V. STRUCTERED QUERY LANGUAGE (SQL)

A. The graphical user Interface (GUI)

There are two text areas, one to show the output from the database for a query that has been input and a second one used to write a query. This text area is editable but can also have text inserted from a multiple choice drop text box. The database source can be chosen from the databases that are available and which have been written into the configuration file.

There is also an area on the screen where the meta-data corresponding to the chosen database can be displayed if the metadata exists, these are the table names and the column names within the database. This is useful when framing a query as these details are needed to write a correct query.

B. Query examples.

And typical entries for the University database, for SQL queries, are also as below:

SELECT * ; FROM staff;
SELECT name, staff_number ; FROM staff;
SELECT DISTINCT staff_number ; FROM tutors;
SELECT staff_number ; FROM staff ; WHERE name = 'Jennings';

C. The scope and format of the simulation of SQL queries.

The simulator is capable of dealing with queries that contain:

- SELECT clauses
- FROM clauses
- WHERE clauses
- GROUP BY clauses
- HAVING clauses and
- ORDER BY clauses
- And a single level of nesting

The simulator not only carries out the processing of the requested query but shows all of the steps in the logical processing model in the order that the database would carry out the necessary calculations. The tables are shown separately and titled to show their relevance to the various stages.

B. The exercise

Figure 4 shows the result of displaying the result of the query "SELECT AVG (height) AS average_height FROM patient". The screen shows the complete SELECT clause and just the heading for the FROM clause. The order in which they are displayed is the order in which the database will process such a query**.**

The exercise proceeds by the student choosing the database from the upper drop down menu and then if a table of queries exists the query that they wish to run. Alternatively if the query that the student wishes to run is not available this can be typed into the upper right text area. Of course they have to get the syntax right for query to give any result.

The Sybase examples have metadata associated with them that allows the display of the columns of the relations. This is more of an aid to memory but it can be useful to have that information on the screen rather in a book that has to be searched through.

C. The lessons learnt

There are many lessons that the student will learn from experimenting with prepared queries and queries that they structure themselves.

The way that the various stages of the query are shown will help understand how the database deals with a query and also to construct their queries in a way that the database will understand.

They will also learn the need to be absolutely correct with the syntax of their own queries because otherwise the database will give an error message which usually is not that informative.

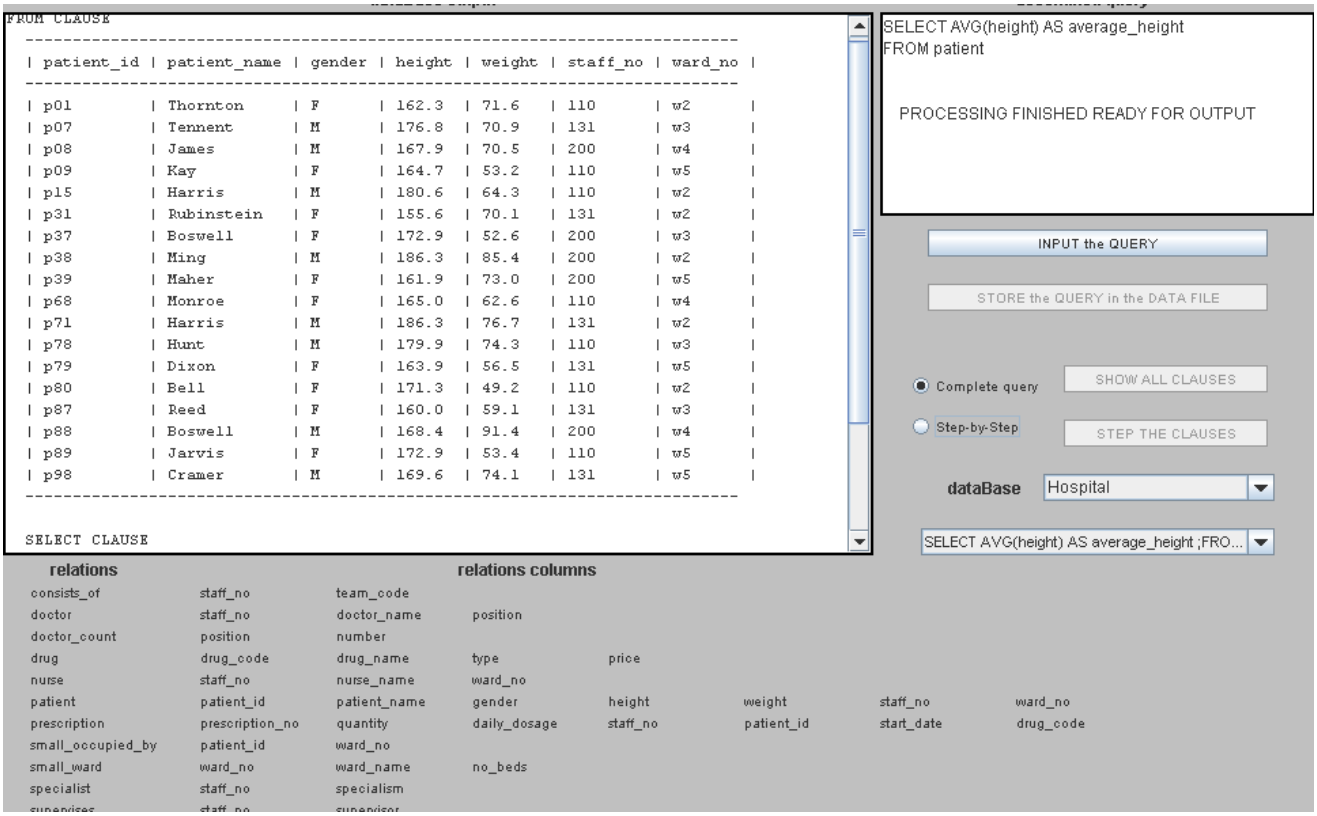


Figure 4 The screen showing the "FROM and SELECT" output

VI. FUNCTIONAL DEPENDENCY

A. The graphical user Interface (GUI)

As can be seen, from Figure 5, there are three areas shown on the right hand side of the screen, they have scroll bars that appear if necessary. The three areas are to display the results of single, double or triple determinant terms used for finding any functional dependencies within a table. There is screen area where the table is written out and three drop down menus and two buttons. The dropdown menus are used to choose a database and a table from that database or a data source being an individual page.

The far left hand button allows the exercise to be restarted at any time but preferably when one exercise has been completed. The reset selection button is used to begin a search through the table once a determinant has been chosen. The determinant is set via the check boxes above each column.

B. The exercise

As explained in the GUI description the START/RESTART button is pressed and that allows the choice of the database followed by the choice of the table.

The determinant to be tested is then chosen and the when satisfied the START SEARCH button is pressed. The student now has to work a little bit harder. The table has to be stepped through for each of the columns that are not part of the determinant. Once a column has been stepped through and a dependency is detected between the chosen determinant and the column then this is notified in one of the text areas at the bottom depending upon the length of determinant that has been selected.

This same procedure has to be carried out for all of the columns not within the determinant chosen. Clearly if all of the columns give rise to a notification of a dependency then a possible primary key has been found.

Various combinations can be tested but they should be combinations that the student thinks from observation would have some dependencies between the chosen determinant and the remaining set of columns.

C. The lessons learnt

A student with little idea of the meaning of functional dependency would eventually grasp the idea that it meant that for each particular determinant there could only be one instance of a parameter with a particular value in a column if that column was functionally dependent. If the column had two entries of the same value that there would be no functional dependency for that column. They would be able to understand the meaning of a primary and perhaps secondary

key if there were one in the table The significance of the results should be seen when using the normalisation simulation as normalisation depends upon being able to identify any dependencies within the table

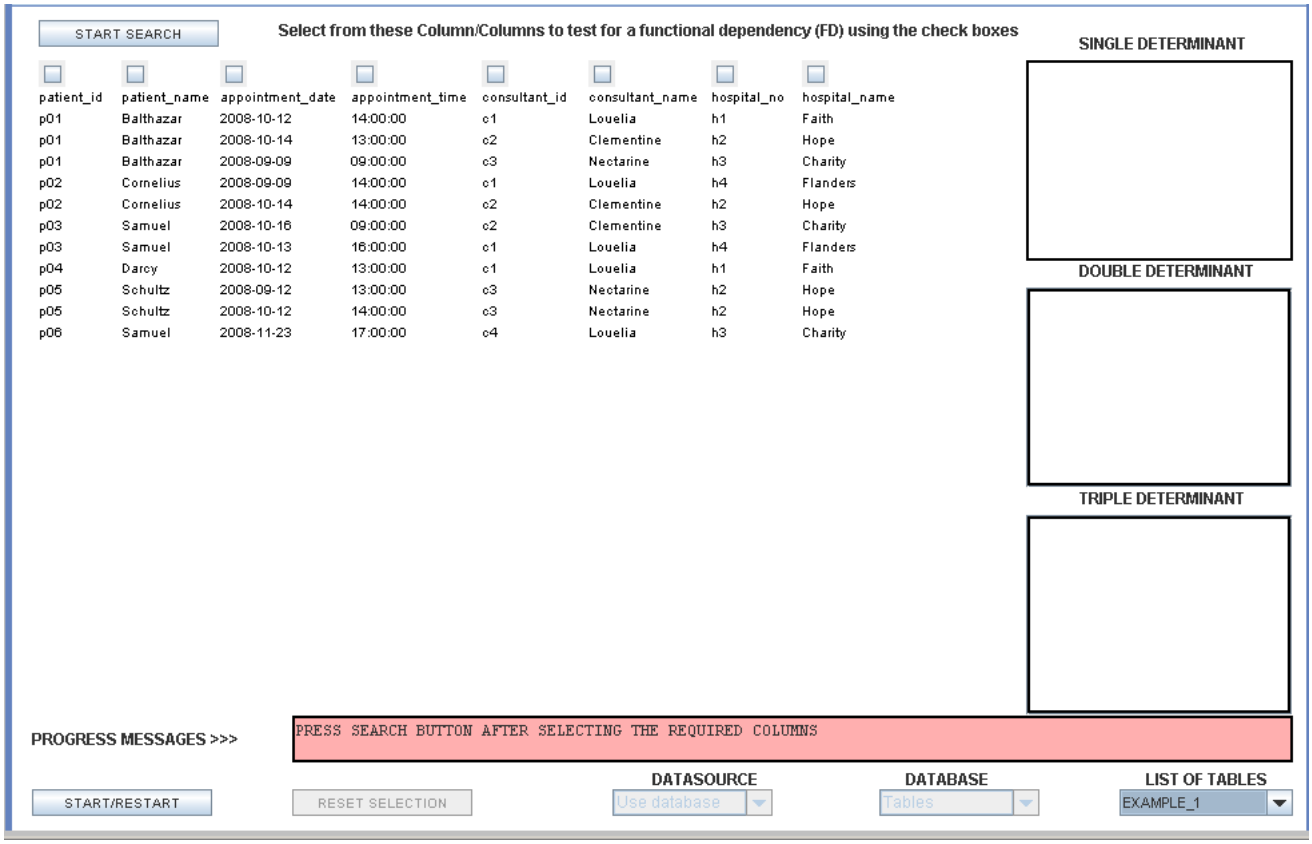


Figure5. The initial screen for functional dependency exercises where the database and table have been chosen

VII. NORMALISATION

A. The graphical user interface (GUI)

Figure 6 is a condensed view of this interface being in fact three pages to allow the display of the relatively large amounts of data that is possible using this particular tool. In Figure 6 the interface is shown part the way through an exercise. There are two prominent text areas; the top one is where the selected table is displayed and the lower one where de-selections of columns from this table are shown separately. The top text area also has a number of check boxes to allow the columns chosen in a selection to be indicated. There are drop down menus used to choose a database and a table within that database.

Initially the SHOW SELECTION button is enabled and when pressed will display the columns chose in the lower area. In order to make a fresh selection the RESET SELECTION button has to be pressed which resets the check boxes.

When the student is satisfied with all of their selections the RECOMBINE button is pressed and this joins the separate tables into one table. If the selection has been carried out correctly the original table will be drawn in the lower text area. If there has been a mistake and there is no dependency between the tables the table that results are dependant on which database is being used. For the Access databases the table that results is much longer and clearly wrong in this context. For the Sybase files the database indicates an error because the selections cannot be joined. The Excel databases cannot deal with joins and thus always give an error even if the table has been correctly partitioned.

B. The Exercise

This exercise should really be carried out after examining the chosen table for dependencies and the student should use the results to make an informed choice of selections. Selections are made and the recombination that results studied and explained even if there are errors. An example of a very small table is given in appendix 3 and it shows the effect of correct and incorrect normalisation.

C. Lessons learnt

Students find normalisation quite confusing and being able to experiment with different selections can help in lessoning the

confusion. They learn how to split tables into a number of tables where the columns are sensibly grouped to include information that is relevant to individual tables. They learn to discriminate between what is needed in a table and what can usefully be put into a different table. They also learn that to be able to recreate the original table after division that there has to be information in both tables that is common to both tables and has some dependency. They should also appreciate the benefit that using the functional dependency simulation has given to them in running this simulation.

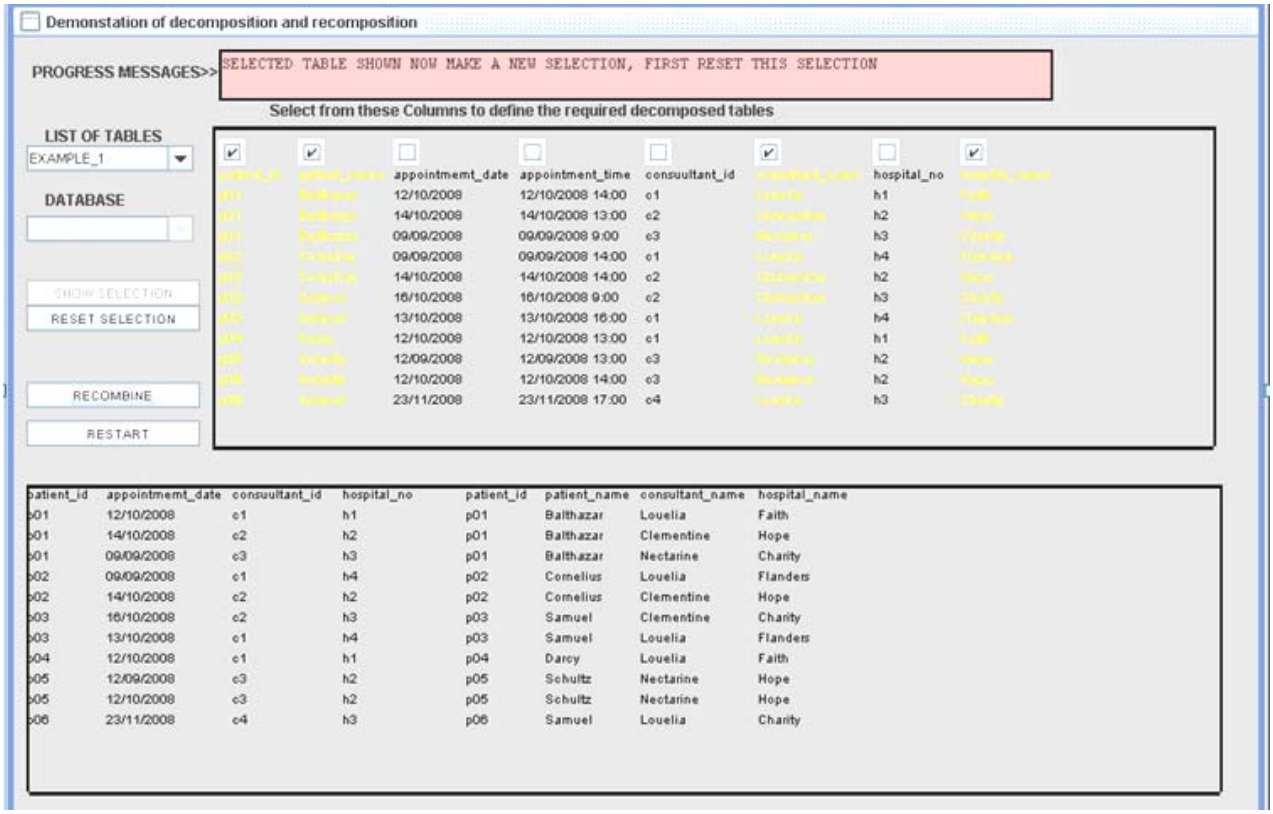


Figure 6. Selection of Tables prior to recombining them

VIII Third party software

To demonstrate that the simulations that involve access to a database a number of databases that are readily available, if not freely available have been used. MicroSoft Excel and MicroSoft Access are generally available to students as MicroSoft now offers fairly generous discounts to students so these were obvious choices. The Open University course 'Relational databases: theory and practice' uses SyBase for much of the SQL that the course uses and hence its use here.

Each of the databases used must have an ODBC data source registered with the Administrative tools of the host computer, such a registration is similar to the one in Figure 7. Figure 8 shows the need to specify the path to the database file in the case of an Excel file.

Excel has an ODBC-JDBC bridge as part of Excel so this would seem a great convenience to use Excel. It has its drawbacks as Excel lacks some of the facilities of a full database. Access does not have a bridge included and this did seem to be stumbling block at one point. However, a solution was found that had all of the capabilities required an ODBC-JDBC bridge from Hongxin Technology & Trade Ltd.

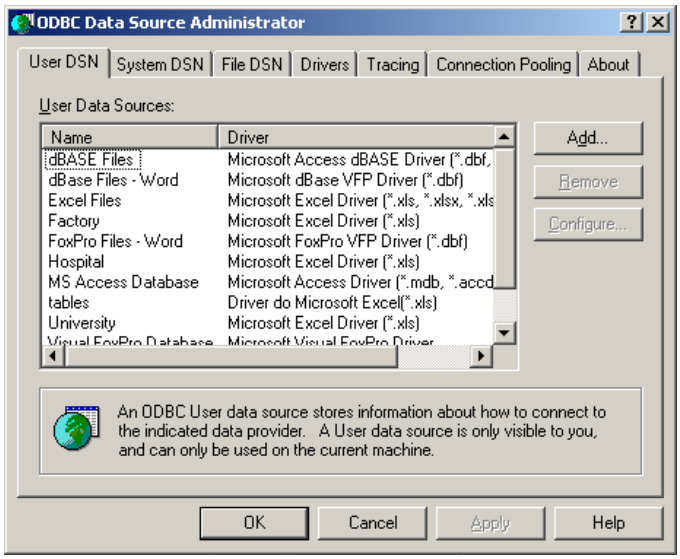


Figure 7 The user DSN examples.

The specification states "*HXTT Access packages include a Type 4 JDBC driver. Type 4 indicates that the driver is written in Pure Java, and communicates in the database system's own network protocol. It supports { UNION | INTERSECT | EXCEPT | MINUS } [ALL] query , INNER JOIN, FULL JOIN, LEFT JOIN, RIGHT JOIN, NATURAL JOIN, CROSS JOIN, and sub-query which includes single-row sub-query, multi-row sub-query, multiple-column sub-query, inline views, and correlated sub-query.*"

Sybase is a full database but does require a more complex setting procedure.

For students it might be a useful exercise for them to set up some Excel and Access files them selves. Full instructions can be found online from JavaWorld.

VIII ACKOWLEDGEMENT

I wish to thank Tony Valsamidis a senior lecturer at the University of Greenwich for his help in making it possible for some of these simulations to be tried out in the classroom in 2010.

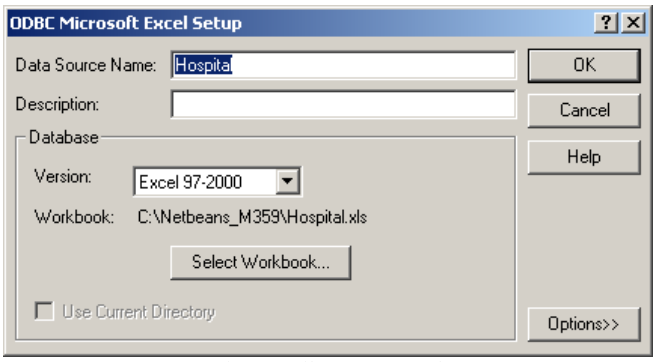


Figure 8 File setup as used

IX TESTING

Testing of the tools is to be carried out in 2010 and some results may be available later in 2010. The consensus of those that have seen the software is that it should help in students' understanding of relational database principles

X REFERENCES

[1] Pin-Shan Chen, "The entity-relationship model toward a unified view of data," ACM Trans. Database Syst., pp. 9–35, March 1976.

[2] Carol Chrisman, "Teaching Database Design Through an Entity-Realtionship Approach", SIGSE Bulletin, February 1982, pp.4 - 7.

[3] Judith D, Wilson, "Entity-Relantionship diagrams and English: An analysis of some problems encountered in a database design course", SIGSE '87: Proceedings of the eighteenth SIGSE technical symposium on Computer Science education

[4] Sadik, S., Orlowska M., Sadiq W., Lin J.,: "SQLator: an Online SQL Learning Workbench, Proceedings of the 9th Conference on Inovation and Technology in Computer Science Education, 36, 3 (june2004), pp 223 – 227.

[5] Sciore E., "SimplDB,: A simple Java_based Multi-User System for teaching Database Internals," Proceedings of the 28th SIGSE Technical Symposium on Computer Science Education, 39. 1 (March 2007). pp 561 – 565

[6] Allenstein B., Yost A., Wagner P., Morrison J., " A Query Simulation System Tp Illustrate Databse Query Execution" Proceedings of the 29th SIGSE SIGSE Technical Symposium on Computer Science Education, March 2008

[7] Teaching material used for the Open University course M359 "Relational databases: theory and practise" first published 2006

[A summary of the M359 Open University course material is availbale on 'OpenLearn]

XI Appendices

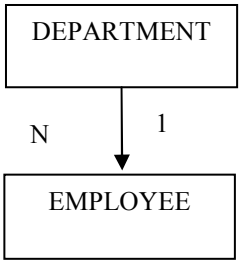
Appendix 1

A relational model of employee

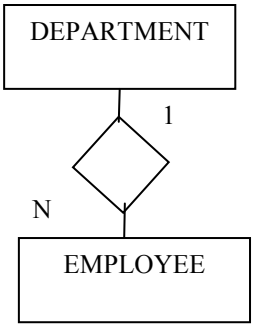
Role		Legal	Legal	Alternative	Alternative	
Domain	Employee-number	First-name	Last-Name	First-Name	Last-Name	No-Of-Years
Tuple	2566	Peter	Jones	Sam	Jones	25
	3378	Mary	Chen	Barb	Chen	23

Employee is the Relation and the components of the domain are the attributes.

A simple Network model



The equivalent entity-relationship diagram

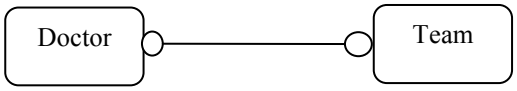


The above are all taken from Chen's paper.

Appendix 2

An example where the Relation for Relationship is needed

Consider a relationship had optional participation at both ends and use the example shown in figure 9.



Doctor (StaffNo, DoctorName, Position)
Team (TeamCode, TelephoneNumber)

Figure 9 an inadmissible Entity Relationship

Some teams are not headed by a doctor and some doctors do not head teams. That is none of the alternatives for the placement of the foreign key would not be possible, since the key indicates mandatory participation on the part of the owner of the key. Some doctors would not be associated with a team and some teams would not be associated with a doctor. In cases such as this, we have to introduce a new relation to represent the relationship.

Appendix 3

Sample table 'quota'

course_code	limit	date_reviewed
c2	3000	2007-10-01
c4	250	2007-10-01
c5	250	2008-04-23

The determinant course_code has a dependency between limit and between date-reviewed. Thus splitting the table into the two tables (course_code, limit) and (course_code, date_reviewed) would, when recombined give the initial table, but For the determinant 'limit' splitting the table into two tables of (limit, course_code) and (limit, date_reviewed) would produce the table

c2	3000	2007-10-01
c4	250	2007-10-01
c4	250	2008-04-23
c4	250	2007-10-01
c5	250	2007-10-01

Two rows that were not present in the original table have been introduced.