University of Glasgow
Dip / MSc Information Technology
Information Systems and Databases

Practical Exercise

October 29th, 2009

The Task

On the second sheet is a set of requirements for a database application, which you are required to implement as a Java application on top of Oracle. This includes setting up a database schema, creating the database with some sample data, providing queries and views, and implementing a working application program to use the database. The steps for doing this include:

i) Create a conceptual schema in the form of an ER diagram.

ii) Derive a relational schema from the ER diagram.

iii) Implement this schema by using SQL commands through a suitable SQL interface in Oracle.

iv) Populate the database with a set of typical data using a suitable mechanism, i.e. SQL, SQL Loader or the Enterprise Console. The data should be significant but manageable, between one and two times the size of the bank database. The key aspect is that queries can tested since they select some and not other data.

v) Define indexes which accelerate certain queries; and specialised views which are appropriate to various sub-groups of users to focus their attention or to prevent them having unauthorised access to data. Decide on the storage structures you want for optimal use of your database.

vi) Define a range of SQL queries which could be used as the basis of "Canned Queries" for naïve users.

vii) Define some typical users and roles and assert the privileges for each for access to the tables and views.

ix) Create an application as a Java program connecting to the database with JDBC and having a user interface defined in AWT, Swing or HTML, with methods for summoning the queries from the interface. You will be given most of the application code and will only need to make a small number of application specific changes.

The Deliverable

You should hand a **report suitable for delivery to a client** including at least the following; (approximate marks out of 100 are shown). Ensure that you use suitable styles and font sizes to make the document attractive and readable.

a) An introductory section explaining clearly to a **non-technical** client, i.e. someone who is not a database specialist, what you are delivering to them. This will describe the structure of the data held in the database, the ways in which it is created and the uses to which it can be put. This must include an informal account of the requirements you have met (i.e. the assumptions you made about the data). (15 marks given to the clarity of the description from the point of view of a non-expert client).

b) A discussion of expected user roles including a table of user roles against table/view names indicating the privileges of each role on each table/view. NB you do not have to implement this, just describe what should have been done - this is best put in as a table as part of (a) (5 marks)

c) The conceptual schema - i.e. an ER diagram. We expect your schema to use all of the ER constructs – i.e. single, composite and multi-valued attributes, 1-1, 1-N and M-N relationships, attributes of relationships and weak entity types – you don't have to include non-binary relationships. (5 for milestone + 10 for final version)

d) The relational database schema - captured as create table commands in an Oracle SQL session. This should include not just the set of columns for the tables but also the integrity constraints (primary key, foreign key, range constraints, non-nullness, uniqueness, etc.) For each table, also indicate which storage structure you would ideally use - i.e. a sorted structure for tables which may become large and a heap for small ones (5 for milestone + 10 for final version).

e) The sample data (not too big please) - best printed in a small font to save paper. (5)

f) A set of index definitions - these should be designed to speed up the queries you have provided - you can create these in Oracle using the *create index* commands - note you do not see any effect of this, but you can check them using the Schema Manager. (5)

g) A set of view definitions listed as SQL *create view* commands + a description of their content and purpose. (5)

h) A set of up to 20 SQL queries that may be frequently used, including a description of what each does and sample output. A range of types of query should be included, so that you have demonstrated your mastery of a range of SQL querying techniques. Thus there will be marks for simple queries, the use of aggregates such as count or average, the use of nested queries (where the nesting is vital) and the use of negatives in queries. (30 – 10 for basic queries, 10 for using aggregate functions, updates and parameters, 10 for complex queries).

i) A short description of your application program. As you are all using a standard interface, all you need provide is a list of which queries you included - and make sure that you have described them as mentioned in (h) above. You should also provide a listing of the class *DBinfo*. (5).

Milestones and Deadlines

In order to make the project more feasible and to give you early feedback. We propose the following timetable:

Design Milestone. Let your tutor have an ER diagram and a provisional set of tables for your project by 5pm on Tuesday 10th November - hand back available during the next week.

Final Handin: This is 2pm on Wednesday, 16th December 2008. Deliver this by putting it in a drop box in House 14.

Assessment

The work counts for 30% of the final mark for the module – the 100 marks will be scaled down. Do not spend an inordinate amount of time on the project, but keep the database structure and the size of the data sufficiently large to exercise the facilities, but not to overburden you with typing. Note for instance that repeated use of the same technique (e.g. lots of multi-valued attributes) will not gain you more marks, but would use up a lot of your time. What will get you marks is demonstrating your mastery of different concepts (ER constructs, database constraints, SQL query types, etc.).

Creating a Database of Industrial Contacts

We want to be able to keep track of the university's contact with industrial organizations - including, information about companies, their locations, activities, staff and so on together with details of which members of the department have dealings with them. For each company, we will identify one member of staff as our primary contact. Company staff will only be considered to be uniquely described within their company.

Typical relationships with the companies include: providing a course that company employees can attend (NB courses for students are not part of the exercise); jointly working on a project; consulting with the company; having a vacation job for a student; having a studentship from the company; staff having previously worked for a company; visits by individuals or teams from the company or from the university to the company, etc. Relationships may be with individuals in the company or with departments.

Typical queries might be:
Who do we know at IBM, at which site are they and what are their interests?
What projects are we involved in with which companies?
Who in the department has worked for Apple?
Which courses are currently on offer to which companies?

Typical users might be:
Staff and students of the CS department
Staff and students of the Business Studies department
The publicity department of a local company
The university finance office

There are a range of possible sets of information here. You should specify a significant subset of the possibilities and start your database design from there. Remember again that it is the range of concepts that you exercise that is important not the completeness of the database to be able to handle everything that could possibly be useful.