Computer Science 455

Final Exam

Name ____________________

The relational schema for the supplier-parts-projects (SPJ) database:

s(sno, sname, status, city)
p(pno, pname, color, weight, city)
j(jno, jname, city)
spj(sno, pno, jno, qty)

Monday, May 9, 2005
8:00 AM – 10:00 AM
200 pts

Outline: Over the course of the semester we have covered three areas in the introductory study of database systems: theory, design, and the database language SQL. After some introductory general questions, we will treat each topic in turn.

I. Introduction.

You are being interviewed for a new job job, and you find yourself suddenly called before a group of executives. They have, it appears, some questions for you. Answer the following questions, keeping in mind that it is possible (and sometimes necessary) to be precise and complete without being technical.

1. (10 pts.) The first question appears fairly innocuous. They want to know what a database management system is. But rather than a technical answer, they claim ignorance of the basics, and ask you to describe what a database is as if to an intelligent and interested person. They are particularly interested in what problems are solved (or at least mitigated) by this approach.

2. (10 pts.) It appears that in an overheard hallway conversation someone has mentioned the user, logical, and physical levels of a database, and they would like an explanation. Include in your explanation the role each level plays.

3. (10 pts.) The executives probe a bit deeper. They next want to know about data normalization. Not the formal definition (you get to discuss that later on with the technical people), but one which expresses the basic ideas. Include what it means for a relation to be normalized in the sense of first normal form, but include a brief description of the goal of higher normalizations.

4. (10 pts.) They finally ask you for a definition of the role of the database administrator and the data administrator.

Phew! Survived that one! As you left, they reminded you that there are plenty of administrative roles to play for which a solid knowledge of database management systems is important, and have asked if you would be interested in any of the administrative jobs (as in problem 4 above). Sadly, you don't get to answer that just yet, as you are quickly taken to a room with a bunch of people and a greaseboard filled with expressions in the formal relational algebra. You must be among the theorists. They also have some questions for you.

II. Theory

1. (10 pts.) What is a data model for a database system? As a part of your answer, list at least three of the four data models we have talked about this term.

2. (10 pts.) Describe the relational data model in more detail.

3. (10 pts.) Why is the relational model called relational?. Include some detail.

4. (10 pts.) Give a formal description of functional dependency: That is, what does it mean to say that in a relation **R**, attribute **A** functionally determines attribute **B** (don't worry at the moment that the idea extends to groups of attributes. But give a definition, and say what we say about A in this case.

5. (10 pts.) Give brief (but more formal than in the first part) definitions of 2NF, 3NF, and BCNF using the language of functional dependency.

6. (15 pts.) The relational algebra can be thought of as an assembly or machine language for the higher order SQL language. Consider the following SQL statement:

```
select sname from s, j, spj
        where j.city = 'LONDON'
        and s.sno = spj.sno
        and j.jno = spj.jno;
```

How would this appear in the relational algebra? Feel free to use any of the forms of the relational algebra we have studied (though this can be done with the most basic form of EASYALG), making connections between the algebraic statements and the elements of the SQL statement above.

7. (5 pts.) The full relational algebra is equivalent in expressive power to SQL. In SQL we have a (somewhat painful) way of handling **ForAll** queries (an example is coming up shortly). What command in the relational algebra does this?

Well, you got through that! On your way out to your next appointment, the group asked you if you had considered graduate studies in database systems. There are, they pointed out, lots of interesting questions to be answered, a portion of which include object oriented databases and the problem of adding semantics to database systems. But you don't have time to think about that just at the moment, since you now find yourself in a room filled with entity-relationship diagrams, data flow diagrams, and structured object design. You are clearly in the design group, and they also have some questions for you.

III. Design.

1. (10 pts.) Sketch an entity-relationship diagram for the following problem:

   The InterDepartmentalProjects group consists of several teams. working on projects. Each project is headed up by one of the team members in the project team. Each employee is a member of just one team. Each project is tied to just one contract each. Each contract is ordered by one customer, though a customer can have several projects. Employees have an employee number, employee name, and annual salary. Contracts have a contract number and total contract cost. Projects have a project title and a total budget.

2. (10 pts.) From the entity-relationship diagram in the preceding problem (1) above, sketch a relational schema, carefully labeling primary and foreign keys, and indicating which foreign keys support the relationships in problem (1).

3. (10 pts.) Top-down and bottom-up design techniques have complementary advantages and disadvantages. Describe briefly what top-down and bottom-up design are, and how they complement each other.

Design is fun! One of the best things about it is that there is rarely (except on exam questions) just one right answer – only answers that have differing strengths and weaknesses. You hated to leave that room, but you are finally ushered into a room with nobody in it. You are presented only with a workstation running SQL*Plus and a series of questions. Here they are.

IV. SQL problems.

SQL consists of both a data definition language (DDL) and a data manipulation language (DML). We begin first with a few DDL statements.

1. (10 pts.) A student table consists of a student number (primary key), a student name, major, and a faculty advisor (a foreign key referencing the FNO field of the FACULTY table). Using reasonable data types and enforcing entity and referential integrity, write the CREATE TABLE statement for this table.

2. (5 pts.) In the SPJ table, write the SQL code to create a non-unique index over the SNAME field of the S table.

3. (10 pts.) Create a view for the SP table. That is, create a view called SP(sno, pno, qty) which contains the **sno** and **pno** information from the **spj** table together with the sum (over projects) of each part supplied by a supplier.

We now (and finally) turn to questions in the SQL DML.

4. (5 pts.) Find the names (only of RED parts.

5. (10 pts.) Find the names (only) of parts used on LONDON projects.

6. (10 pts.) Find the names (only) of parts used on **no** LONDON projects.

7. (10 pts.) Find the names (only) of parts used on all LONDON projects (a standard FORALL question).

The moral of the story is that database management systems presents students with a great many career opportunities in administration, theory, software engineering (either database design or the integration of databases into software systems), or the nuts and bolts of making a database do what the client wants it to do. There is, of course, no reason to restrict yourself to just one of these areas. Go for it! And have a great Summer!