Course ISE 322: Database Systems
Recitation 2 Exercise

Michael J. May

October 27, 2009

The recitation begins with a series of review questions.

1 Exercise 1.1 in Ramakrishnan and Gehrke

Why would you choose a database system instead of simply storing data in operating system files? When would it make sense *not* to use a database system.

2 Exercise 1.2 in Ramakrishnan and Gehrke

What is logical data independence and why is it important?

3 Exercise 1.3 in Ramakrishnan and Gehrke

Explain the difference between logical and physical data independence.

4 Exercise 1.4 in Ramakrishnan and Gehrke

Explain the difference between external, internal, and conceptual schemas. How are these different schema layers related to the concepts of logical and physical data independence?

5 Exercise 1.5 in Ramakrishnan and Gehrke

What the responsibilities of a DBA? If we assume the DBA is never interested in running his or her own queries, dose the DBA still need to understand query optimization? Why?

6 Exercise 1.6 in Ramakrishnan and Gehrke

Scrooge McNugget wants to store information (names, addresses, descriptions of embarrassing moments, etc) about the many ducks on his payroll. Not surprisingly, the volume of data compels him to buy a database system. To save money, he wants to buy one with the fewest possible features, and he plans to run it as a stand-alone application on his PC clone. Scrooge does not plan to share his list with anyone. Indicate which of the following DBMS features Scrooge should pay for; in each case, also indicate why Scrooge should (or should not) pay for that feature in the system he buys.

1. A security facility.

2. Concurrency control.
3. Crash recovery.
4. A view mechanism.
5. A query language.

7 Exercise 1.7 in Ramakrishnan and Gehrke

Which of the following plays an important role in *representing* information about the real world in a database. Explain briefly.

1. The data definition language.
2. The data manipulation language.
3. The buffer manager.
4. The data model.

8 Exercise 1.8 in Ramakrishnan and Gehrke

Describe the structure of a DBMS, If your operating system is upgraded to support some new functions on OS files (e.g., the ability to force some sequence of bytes to disk), which layer(s) of the DBMS would you have to rewrite to take advantage of these new functions?

9 Exercise 1.9 in Ramakrishnan and Gehrke

Answer the following questions:

1. What is a transaction?
2. Why does a DBMS interleave the actions of different transactions instead of executing transactions one after the other?
3. What must a user guarantee with respect to a transaction and database consistency? What should a DBMS guarantee with respect to concurrent execution of several transactions and database consistency?
4. Explain the strict two-phase locking protocol.
5. What is the WAL property, and why is it important?

The second part of the recitation involves instantiating some of the concepts that we saw in lecture last time.

10 Students, Courses, Enrolled

Let us define the following relations and views as we have seen in lecture:

Students (*sid*:`int`, *name*:`string`, *login*:`string`, *age*:`integer`, *gpa*:`real`)

```
CREATE TABLE Students (sid integer,
    name varchar(40),
    login varchar(40),
    age integer,
    gpa real,
    primary key (sid))
```

Courses (*cid*:`int`, *cname*:`string`, *credits*:`integer`)

```
CREATE TABLE Courses (cid varchar(20),
    cname varchar(40),
    credits integer,
    primary key (cid))
```

Enrolled (*sid*:`int`, *cid*:`string`, *grade*:`real`)

```
CREATE TABLE Enrolled (sid int,
    cid varchar(20),
    grade real,
    primary key (sid, cid),
    foreign key (sid) references Students,
    foreign key (cid) references Courses)
```

Use the attached Excel work book provided to insert rows into the database and fill up the tables.

11 Views

Let's try creating the views mentioned in class to see how they work.

Enrollment (*cid*:`string`, *cname*:`string`, *enrollment*:`integer`)

```
CREATE VIEW Enrollment AS
    select C.cid, C.cname, count(E.sid) as enrollment
    from Courses C, Enrolled E
    where C.cid = E.cid
    group by C.cid, C.cname
```

- Let's not worry about the meaning of all of the parts of the query just yet. Run the command and then open up the view Enrollment that you will see added to the tree. Alternatively you can type the command `SELECT * FROM Enrollment` and get the results at the bottom of your screen.
- Notice that the course without any student enrollment doesn't appear. That is an artifact of how the view has been defined in SQL. We'll learn how to correct that problem later.
- Try adding a row to the Enrolled relation and reload the Enrollment view. Try deleting a row from Enrolled. What happens?
- Try deleting a row from Courses, then reloading the view Enrollment. What happens?

Let's create the second view as well.

GpaPerCourse (*cid*:`string`, *avggpa*:`real`)

```
CREATE VIEW GpaPerCourse AS
    select C.cid, avg(S.gpa) as avggpa
    from Courses C, Students S, Enrolled E
    where C.cid = E.cid
    and S.sid = E.sid
    group by C.cid
```

- Now try updating the Students, Course, and Enrolled relations by adding, modifying, and deleting rows. What happens?

12 ERD Models

After completing the database work, let us move on to the ERD exercises which we will do together.