EFREI – M1 – RDBMS – 2008-2009
Written Exam (2 hours)

Yann Thierry-Mieg – Yann.Thierry-Mieg@lip6.fr

December 16, 2008

All documents authorized.

1   General Questions (5/20)

Answers should be about one paragraph long.

1. Which part(s) of SQL are not portable across RDBMS implementations (e.g. Oracle to MySQL)?

2. What kind of business constraint can you enforce by creating Foreign Key constraints ?

3. Is SQL a programming language ? Explain your answer.

4. In which case(s) can it be indicated to use recursive SQL queries ?

5. How do you write a Java program that interacts with a DBMS ?

2   Design (7/20)

We want to design a database to store emails within an engineering school intranet. There are two types of email accounts: personal accounts and group accounts. The back-end database should keep the following information:

- Each account is identified by a unique alias (e.g. "Y.ThierryMieg", "group-SIA").

- For every personal account, we want to record the name of the person and his status (e.g. student, faculty, administrator, director...).

- For every group account, we want to record its one and only owner and all its members. The group owner must be a personal account; group members may be group accounts.

- A group must have at least one member.

- Each message has one sender and at least one receiver. The sender must be a personal account. We also want to record the subject of the message (e.g., "Re: Free pizza!") as well as the time stamp (e.g., 1:45:24AM, Aug 22, 2008). Assume a unique identifier is generated automatically for each message.

1. Design an E/R schema to represent this system.

2. Give the relational model obtained from the E/R schema of question 1 by application of translation rules.

3 SQL (8/20)

We consider the database COURSES of schema :
STUDENT (<u>SID</u>, SNAME, SBIRTHDATE, SSEX)
GRADE (*SID, CID*, GRADE)
COURSE (<u>CID</u>, CNAME, CTHEME, CCOEFF, *FID*)
FACULTY (<u>FID</u>, FNAME, FTITLE).

SID, CID, FID are unique identifiers. Primary keys are underlined. GRADE.SID is a foreign key to STUDENTS. GRADE.CID is a foreign key to COURSES. COURSE.FID is a foreign key to FACULTY.

1. Give the average age of Male and Female students in the database today.

2. Give the name and title of faculty that teach "History" (Course.CTHEME).

3. Give the name, and SID of students that do not have a grade in "Sociology"(Course.CTHEME).

4. Find the name and coefficient of courses taught by "Assistant" or "Professor" (Faculty.FTITLE) faculty members.

5. Give for each student the average grade obtained for each course theme. Present the result as a table (SID, SNAME, CTHEME, SAVG) sorted by SID then by CTHEME.

6. Give the name, sex, and birth date of students that had a better average grade in "Arts" than the class average for "Arts".

7. Give the list of students (SID,SNAME) that have a grade for every course.

8. Give for each course the name and grade of the best student in the course.

9. Give a table creation script for table GRADE, assume that ID are all fixed length 5 character strings and that grade is expressed as a percentage (e.g. 71%).