Roll Number__________________

# CS3400 Database Systems

## Quiz 2 - 4th Dec 2006

**A. For the questions given below, write the corresponding relational algebra expression. The primary keys for each of the relations are underlined.**

Student (StudentID, Name, Gender, DOB, Curriculum, Major, Year_of_Joining)
Course (CourseID, CourseName, Credit)
PreRequisite (CourseID, PreReqID)
CourseOffering (CID, Semester, Year, Instructor)
GradeReport (SID, CID, Sem, Year, Grade)

**The foreign key, primary key relationships:**
PreRequisite (CourseID) references Course (CourseID)
PreRequisite (PreReqID) references Course (CourseID)
CourseOffering (CID) references Course (CourseID)
GradeReport (SID) references Student (StudentID)
GradeReport (CID) references Course (CourseID)
GradeReport (Sem) references CourseOffering (Semester)
GradeReport (Year) references CourseOffering (Year)

1. Retrieve all those tuples of Student who joined 'Btech' curriculum in 2005.

$$\sigma_{\substack{curriculum='Btech' \\ AND \\ Year\_of\_Joining = 2005}} (Student)$$

2. Find the names of all 2 credit courses.

$$\Pi_{CourseName} \left( \sigma_{Credit=2} (Course) \right)$$

3. Find the names of all those courses which have no pre-requisites.

$$All\_courses \leftarrow \Pi_{CourseID,\, CourseName} (Course)$$

$$Pre\_requisites \leftarrow \Pi_{CourseID,\, CourseName} (Course * PreRequisite)$$

$$\Pi_{CourseName} (All\_courses - Pre\_requisites)$$

#1

$$\Pi_{\substack{Name,\ Curriculum,\\ Year\_of\_Joining}} \left( Student \bowtie_{StudentID = SID} student\_ids \right)$$

4. Retrieve the name, curriculum and year of joining of all those students who got ‘F’ grade in DBMS in the year 2006.

$$Student\_ids \leftarrow \Pi_{SID} \left( \sigma_{\substack{Year = 2006 \text{ AND} \\ Grade = \text{‘F’ AND} \\ CourseName = \text{‘DBMS’}}} \left( GradeReport \bowtie_{CID = CourseID} Course \right) \right)$$

#1

5. Find the names of all the courses taken by ‘Tom’ in the year 2004.

$$course\_ids \leftarrow \Pi_{CID} \left( \sigma_{\substack{Name = \text{‘Tom’} \\ \text{AND } Year = 2004}} \left( Student \bowtie_{StudentID = SID} GradeReport \right) \right.$$

$$\Pi_{CourseName} \left( Course \bowtie_{CourseID = CID} course\_ids \right)$$

6. Find the course name and instructor of all those courses in which atleast 100 students had registered.

$$num\_students\_course \leftarrow \rho_{(CID,\ Sem,\ Year,\ no\_of\_stds)} \ {}_{(CID,\ Sem,\ Year)} \mathcal{F}_{COUNT\ SID} \left( GradeReport \right)$$

$$course\_instruct \leftarrow \Pi_{\substack{CourseID,\ CourseName, \\ Instructor}} \left( Course \bowtie_{CourseID = CID} CourseOffering \right)$$

#2

7. Retrieve all those course names which have always been offered by the same instructor.

$$R_1 \leftarrow \rho_{(CID,\ Instructor,\ num\_times\_offrd)} \ {}_{(CID,\ Instructor)} \mathcal{F}_{COUNT\ YEAR} \left( CourseOffering \right)$$

$$R_2 \leftarrow \rho_{(CID,\ num\_instructor)} \ {}_{(CID)} \mathcal{F}_{COUNT\ Instructor} \left( R_1 \right)$$

$$\Pi_{CourseName} \left( Course \bowtie_{CourseID = CID} \left( \Pi_{CID} \left( \sigma_{num\_instructor = 1} \left( R_2 \right) \right) \right) \right)$$

#2

$$\Pi_{\substack{CourseName, \\ Instructor}} \left( course\_instruct \overset{2}{\bowtie}_{CourseID = CID} \left( \sigma_{no\_of\_stds >= 100} \left( num\_students\_course \right) \right) \right)$$

**b. For each of the following updates, find if it violates any integrity constraint. If so, what would you do to handle it.**

**GradeReport**

| SID | CID | Sem | Year | Grade |
|---|---|---|---|---|
| 1001 | CS3400 | Spring | 2006 | A- |
| 1101 | EC3103 | Monsoon | 2005 | B |
| 1110 | MA7011 | Spring | 2005 | F |

**Student**

| StudentId | Name | Gender | DOB | Curriculum | Major | Year_of_Joining |
|---|---|---|---|---|---|---|
| 1001 | Ramesh | M | 24-08-84 | BTech | CSE | 2003 |
| 1002 | Maya | F | 12-11-84 | BTech | EC | 2002 |
| 1101 | Tom | M | 11-11-80 | MTech | VLSI | 2004 |
| 1110 | Jack | M | 02-10-84 | BSc | Bio | 2005 |

**Course**

| CourseId | CourseName | Credit |
|---|---|---|
| CS3400 | DBMS | 4 |
| MA7011 | LP | 4 |
| MA7012 | Adv. LP | 4 |
| SC1102 | Bio Informatics | 2 |
| EC3103 | Data Compression | 4 |
| CS4123 | DS & Algos | 5 |

**PreRequisite**

| CourseId | PreReqId |
|---|---|
| MA7012 | MA7011 |
| EC3103 | CS4123 |

**CourseOffering**

| CID | Semester | Year | Instructor |
|---|---|---|---|
| CS3400 | Spring | 2007 | Kamal |
| CS3400 | Spring | 2006 | Kamal |
| EC3103 | Monsoon | 2005 | Venkaiah |

**1.Insert**

a. Insert <‘CS3400’,null> into PreRequsite

Entity integrity constraint violation. Either reject this query or enter a valid ‘PreReqId’.

b. Insert <1011,’MA7011’, ‘Spring’, 2005, ‘F’> into GradeReport

Referential integrity constraint violation as SID = 1011 doesn’t exist in the Student table. Provide a valid SID other than 1110.

**2. Delete**

a. Delete the CourseOffering tuple with CID = ‘CS3400’ and Year = 2007

No Violation.

b. Delete the Course tuple with CourseID = ‘MA7012’

Violation of Referential integrity constraint as PreRequisite (CourseId) references Course (CourseId). Either reject the query or do ‘cascade deletion’ which will delete all the tuples referring to the tuples of Course with CourseId = ‘MA7012’

**3. Update**

a. Update the Sem of the GradeReport tuple with SID = ‘1001’ to ‘Fall’

Referential integrity constraint violation.
Update the value of ‘Sem’ to either ‘Spring’ or ‘Monsoon’.

b. Modify the CourseId of the Course tuple with CourseName = ‘Adv LP’ to ‘MA7011’

Violates key constraint as CourseId = ‘MA7011’ already exists.
Update it to some valid CourseId.