In-Semester Test-2

IT 214 Database Management Systems

Time: 90 minutes

Max Points: 80

IMPORTANT NOTE:

1. Write answers neat and clean. Answers that are difficult to read may simply be discarded.
2. Answer all queries in order. If you want to defer answering a query, may move to next by leaving sufficient blank space. You may follow a strategy of not answering more than two queries on a page.
3. Each query is for 10 points, and awarding strategy will be discrete (i.e., 0, 5, and 10).
4. You may have to pay penalty for lengthier query expressions.
5. All queries are to be answered in **Relational Algebra**. No marks will be awarded if answered in SQL.

1. Given below is relational schema extended version of schema **DA-Acad** (discussed in lectures and used in labs). While extending it, names and meaning of original relations and attributes is retained.

Student(StudetID, StdName, ProgID, Batch, CPI)
Course(CourseNo, CourseName, Credit)
Faculty(FacultyID, FacultyName)
Offers(AcadYear, Semester, CourseNo, FacultyID, Grade_Submitted_Flag)
//Attribute "Grade_Submitted_Flag" is used to store grade submission status of a course offer
//FK: CourseNo refers to Course(CourseNo)
//FK: FacultyID refers to Faculty(FacultyID)
Registers(StudetID, AcadYear, Semester, CourseNo, grade)
//for better machine interpretability, let us store grades in numeric form, i.e. 10, 8, and so.
//FK: StudetID refers to Student (StudetID)
//FK: (AcadYear, Semester, CourseNo) refers to Offers
SemesterResult(StudetID, AcadYear, Semester, SPI, CPI)
//FK: StudetID refers to Student (StudetID)
HoR_Wing(wing, gender)
//wing attribute identifies a HoR wing, and draws value from alphabets A to J.
//wings are reserved for a specific gender 'M' or 'F', however assume that
// a wing can be assigned to other gender anytime.
HoR_Room(rno, wing, floor)
//FK: wing refers to HoR_Wing
//floor is labeled as 1 for ground, 2 for first, and 3 for second floor.
//to be informative to human users, attribute rno draws value from pattern
// "<wing><floor><number>", for example rno C110, indicates that it is in
// C wing, ground floor, and number is 10.
Allot(sid, rno)
//FK: **sid** refers to **student**
//All room allotments are recorded in this relation.
// the relation also has **NOT NULL** constraint for attribute **rno**.
// A students whose sid does not appear in this relation indicates that student does not reside in HoR.

Pass.

Student (s2) Empty

Using given relational schema, write relational algebra expressions for following queries-

1. Report (CourseNo, FacultyName) who have not submitted grades for semester 'Autumn', 2016.
2. Report (StudentID) who have got more than two F (i.e. zero) grade in 'Winter', 2016.
3. Report (Faculty-Name) who have taught more than three courses in academic year, 2015-16.
4. Report (StudentID, Student-Name, Grade) of students who are residing in C second floor and took course IT633 in 'Autumn', 2015.
5. Report (StudentID, Student-Name) of M Tech 2015 (progid=11) residing in HoR (Man).
6. Report (StudentID, CourseNo) backlogs* for B Tech ICT (progod = '01') batch 2012.

---

*Let us define backlog as a set of courses that a student has taken ever MINUS set of courses student has got pass grade (i.e. not zero) . Also account for that a student takes a course, let us say IT123, got zero grade in first attempt, and got pass grade in second (or even third) attempt.

7. Report (StudentID, Student-Name, CPI) that have scored more 7.0 grade in all of the courses given here {IT110, IT214, IT205, SC215, IT314, IT301}.

Acad Year

8. Report (CourseNo, FacultyName) for courses offered since 'Autumn', 2010 in which more than 50% students got grade got more than 7.0.