



In-Semester Exam-I (Autumn'2017)
IT 214 Database Management Systems

Time: 90 minutes

Max Points: 70

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. In all questions marks awarding strategy will be discrete (i.e., 0, half, and full marks).
4. You may have to pay penalty for lengthier solutions.

Consider following relation schema created for recording some of placement activities. Assume that we have data for current year's placement records. For instance for year 2018 placement drive; student relation has data for BTech 2015, MscIT and M.Tech. 2017, and so. Explanation for each relation is given along with the relations below.

student(id, name, prog_id, batch, cpi)

- attribute **prog_id** is like '01' for BTech, '11' for M.Tech, '12' for MScIT, and so
- assume that the relation stores all students of UG 2015 and PG'2017

company(id, name)

profile(comp_id, profile, category, ctc, cpi_min)

- relation records profiles for which companies visit in 2018
- attribute **comp_id** is foreign key and refers into **company** relation.
- attribute **cpi_min** is minimum cpi requirement for the profile.
value being NULL for this attribute indicates No CPI criteria
- attribute **ctc** is annual salary that company offers for profile.
- attribute **category** is profile category, that is 'A', 'B', and so.

registers(stud_id, comp_id, profile)

- relation records the instances of students registering in various profiles
- for which companies visit in 2018
- {**comp_id, profile**} is foreign key and refers into **profile** relation.

offers(stud_id, comp_id, profile)

- the relation records all the offers that are made to students in 2018.
- {**comp_id, profile**} is foreign key and refers into **profile** relation.

Write **expressions in in Relational Algebra** to answer following queries
(No marks will be awarded if answered in SQL)

7x10

1. Find companies (List down: comp_name, profile_name, ctc) that are offering 15Lacs and more.

$$\pi_{name, profile, ctc}(\sigma_{ctc \geq 15} (profile \bowtie_{comp_id = id} company))$$

2. List down 'A' category profiles (company_name, profile_name, ctc) that are not having minimum CPI criteria

$$\pi_{name, profile, ctc}(\sigma_{category = 'A' \text{ AND } cpi \text{ IS NULL}} (profile \bowtie_{comp_id = id} company))$$

3. Find out M.Sc(IT) students (ID, Name) who have got 'A' category offers.

$r1 \leftarrow \sigma_{iprog_id='12'}(student)$
 \bowtie
 $r2 \leftarrow offers \boxed{stud_id = id} r1$
 $r3 \leftarrow \sigma_{category='A'}(r2 * profile)$
 $result \leftarrow \pi_{stud_id, name}(r3)$

4. Compute program wise average CTC (list prog_id, avg_ctc).

$r1 \leftarrow offers * profile$
 \bowtie
 $r2 \leftarrow r1 \boxed{stud_id = id} student$
 $result \leftarrow \pi_{prog_id} \mathcal{F}_{avg(ctc)}(r2)$

5. Compute company wise total number of offers made (list Company_Name, Total#Offers).

$r1 \leftarrow offers * profile$
 $r2(id, offers) \leftarrow \pi_{comp_id} \mathcal{F}_{count(stud_id)}(r1)$
 $result \leftarrow \pi_{name, offers}(r2 * company)$

6. List students (ID, Name) who have got 2 and more offers.

$r1(id, no) \leftarrow \pi_{stud_id} \mathcal{F}_{count(*)}(offers)$
 $r2 \leftarrow \sigma_{no \geq 2}(r1)$
 $result \leftarrow \pi_{id, name}(r2 * student)$

7. Compute placement percentage of MScIT (that is percentage of students who have got offers)

$r1 \leftarrow \sigma_{prog_id='12'}(student)$
 $r2(prog_id, ns) \leftarrow \pi_{prog_id} \mathcal{F}_{count(*)}(r1)$
 \bowtie
 $r3 \leftarrow offers \boxed{stud_id = id} r1$
 $r4(prog_id, no) \leftarrow \pi_{prog_id} \mathcal{F}_{count(*)}(r3)$
 $result(percentage) \leftarrow \pi_{no/ns*100}(r2 * r4)$