Comp 5311 Database Management Systems

15. Review 1 Exercises
Consider the following tables, where keys are underlined and foreign keys are in italics. Primary keys are NOT NULL.

- Proposal (PID, SID, Title, Area)
  // The foreign key SID is NOT NULL and corresponds to the ID of the submitter who submitted the proposal.

- Submitter (SID, Name, Email) // A submitter may submit several proposals

- Reviewer (RID, Name, Email, Expertise)

- Review (PID, RID, Score)
  // PID and RID are foreign keys corresponding to the ID of the proposal that was reviewed by the reviewer of RID.
  // Score is in the range [1..5]. A reviewer may review several proposals.
Proposal (PID, SID, Title, Area), Submitter (SID, Name, Email), Reviewer (RID, Name, Email, Expertise), Review (PID, RID, Score)

1.1] Draw an ER diagram that results in the above tables.
1.2] Write an algebra query to return the names of all reviewers who reviewed a proposal in the “Database” area, submitted by Prof Dimitris (i.e., the submitter name is Dimitris).

\[ \pi_{\text{REVIEWER.NAME}} (\sigma_{\text{PROPOSAL.AREA}=\text{DATABASE} \text{ AND SUBMITTER.NAME}=\text{DIMITRIS}} (\text{REVIEWER JOIN}_{\text{RID REVIEW JOIN}_{\text{PID PROPOSAL JOIN}_{\text{PID SUBMITTER}}})) ) \]

1.3] Write an algebra query to return the IDs of reviewers who have only reviewed proposals in the area of their expertise (i.e., these reviewers have reviewed at least one proposal, and have not reviewed any proposal in an area different from their expertise).

\[ \pi_{\text{RID REVIEW}} - \pi_{\text{PROPOSAL.RID}} (\sigma_{\text{EXPERTISE}\neq\text{AREA}} (\text{REVIEWER JOIN}_{\text{RID REVIEW JOIN}_{\text{PID PROPOSAL}}})) \]
1.4] Write an algebra query that gives the same result as the following SQL statement:

```
SELECT SID
FROM PROPOSAL
GROUP BY SID
HAVING COUNT(*)>=2
```

- \( \pi_{P1.SID} (\sigma_{P1.SID = P2.SID \text{ AND } P1.PID \neq P2.PID} (P(P1, PROPOSAL) \times P(P2, PROPOSAL))) \)
1.5] Write an equivalent SQL query for the following algebra expression (\( / \) denotes division):

\[
\pi_{\text{name}} (\text{REVIEWER JOIN}_{\text{RID}} [\pi_{\text{PID, RID}} \text{REVIEW} / \pi_{\text{PID}} (\sigma_{\text{AREA=DATABASE}} \text{PROPOSAL})])
\]

SELECT NAME
FROM REVIEWER R
WHERE NOT EXISTS ( (SELECT PID
FROM PROPOSAL
WHERE AREA=DATABASE)
EXCEPT
(SELECT PID
FROM REVIEW
WHERE R.RID=REVIEW.RID))
1.6] Write an equivalent SQL query *without nested sub-queries* for the following SQL query.

SELECT NAME
FROM REVIEWER
WHERE RID IN (SELECT RID FROM REVIEW
              WHERE SCORE=5 AND PID IN (
                  SELECT PID FROM PROPOSAL
                  WHERE AREA="DATABASE")
              )

SELECT NAME
FROM REVIEWER R, REVIEW RV, PROPOSAL P
WHERE R.RID=RV.RID AND RV.PID=P.PID AND SCORE=5 AND AREA="DATABASE"
• **1.7**] Write a SQL query to return the title and average score of each proposal in the “Database” area.

```sql
SELECT TITLE, AVG(SCORE)
FROM PROPOSAL, REVIEW
WHERE PROPOSAL.PID=REVIEW.PID AND AREA="DATABASE"
GROUP BY PROPOSAL.PID, TITLE
```

• **1.8**] Write a SQL query to return the name, maximum and minimum score of each reviewer who reviewed exactly five proposals.

```sql
SELECT NAME, MAX(SCORE), MIN(SCORE)
FROM REVIEWER R, REVIEW
WHERE R.RID=REVIEW.RID
GROUP BY R.RID, NAME
HAVING COUNT(*)=5
```
1.9] Express in English the result of the following SQL query:

```
SELECT TEMP.TITLE
FROM (SELECT P.TITLE AS TITLE, AVG(SCORE) AS AV
     FROM PROPOSAL P, REVIEW R
     WHERE P.PID=R.PID
     GROUP BY P.PID, P.TITLE) AS TEMP
WHERE TEMP.AV > (SELECT AVG(AV) FROM TEMP)
```

Show the titles of the proposals each of which has an average score above the average of the average scores of all proposals.