Comp 5311 Database Management Systems

15. Review 1 (Conceptual Level) 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).

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

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).

$$\pi_{\text{RID}} \text{ REVIEW} - \pi_{\text{PROPOSAL.RID}} (\sigma_{\text{EXPERTISE} \neq \text{AREA}} (\text{REVIEWER JOIN}_{\text{RID}} \text{ REVIEW JOIN}_{\text{PID}} \text{ PROPOSAL}))$$
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

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

\[ \pi_{\text{Name}} (\text{Reviewer JOIN}_{\text{RID}} [\pi_{\text{PID,RID}} \text{ Review} / \pi_{\text{PID}} (\sigma_{\text{Area}=\text{Database}} \text{ Proposal})]) \]

```
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.

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.

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.
Problem 2: Table: R(A,B,C,D,E)
F = \{ \{A\} \rightarrow \{B, C\}, \{B\} \rightarrow \{A, C\}, \{A, D\} \rightarrow \{E\},
\{E\} \rightarrow \{D\}\}\}

• **2. 1]** Which of the following sets of functional dependencies is a canonical cover of F?

a) \{\{A\} \rightarrow \{B, C\}, \{B\} \rightarrow \{A, C\}, \{A, D\} \rightarrow \{E\}, \{E\rightarrow D} \}\}
b) \{\{A\} \rightarrow \{B\}, \{A\} \rightarrow \{C\}, \{B\} \rightarrow \{C\}, \{A, D\} \rightarrow \{E\}\}
c) \{\{A\} \rightarrow \{B, C\}, \{B\} \rightarrow \{A\}, \{A, D\} \rightarrow \{E\}, \{E\} \rightarrow \{D\}\}
d) \{\{A\} \rightarrow \{B\}, \{B\} \rightarrow \{A, C\}, \{A, D\} \rightarrow \{E\}, \{E\} \rightarrow \{D\}\}
e) Both c) and d)

• **2. 2]** Which of the following functional dependencies is not in the closure F+?

a) \{A\} \rightarrow \{B\}
b) \{B\} \rightarrow \{B, C\}
c) \{C\} \rightarrow \{A\}
d) \{A, D\} \rightarrow \{C, E\}
e) \{A\} \rightarrow \{C\}
2.3] Which of the following set is a subset of \{A, D\}+?

a) \{A, B\}

b) \{B, C, D\}

c) \{E\}

d) All of the above

e) None of the above

2.4] Which of the following is a superkey for R?

a) \{A\}

b) \{AB\}

c) \{BC\}

d) \{ACD\}

e) \{ABC\}
Table: R(A,B,C,D,E)
F= {{A} → {B, C}, {B} → {A, C}, {A, D} → {E},
{E} → {D}}

• 2. 5] Which of the following is a candidate key for R?
a) AD
b) AE
c) BD
d) BE
e) All of the above

• 2. 6] Consider the following decomposition: {A, B, C}, {A, D, E}. Which of the following statements is true?
a) The decomposition is 3NF, lossless join and dependency preserving
b) The decomposition is 3NF, lossless join but not dependency preserving
c) The decomposition is 3NF, dependency preserving, but not lossless join
d) The decomposition is lossless join, dependency preserving but not 3NF
e) The decomposition is 3NF, but neither lossless join nor dependency preserving
Table: R(A,B,C,D,E)
F = \{ \{A\} \rightarrow \{B, C\}, \{B\} \rightarrow \{A, C\}, \{A, D\} \rightarrow \{E\}, \\
\{E\} \rightarrow \{D\}\}\}

2.7] Consider the following decomposition: \{A, B, C\}, \{A, E\}, \{D, E\}. Which of the following statements is true?

a) The decomposition is BCNF, lossless join and dependency preserving

b) The decomposition is BCNF, lossless join but not dependency preserving

c) The decomposition is BCNF, dependency preserving, but not lossless join

d) The decomposition is lossless join, dependency preserving but not BCNF

e) The decomposition is BCNF, but neither lossless join nor dependency preserving
Consider table R(A,B,C,D,E). Given the functional dependencies in the first column of the following form, complete the form accordingly.

- In the second column, list all candidate keys for R.
- In the third column, provide a maximal decomposition of R in 3NF (we only decompose when there is violation of 3NF) – if R is already in 3NF just write R(A,B,C,D,E) instead of a decomposition.
- In the fourth column, do the same for BCNF decomposition. If there are multiple options, choose any dependency preserving decomposition.
- Give the results without comments. The first row is given as an example. Each answer is 2%.

<table>
<thead>
<tr>
<th>Functional Dependencies</th>
<th>Candidate Keys for R</th>
<th>Decompose R in 3NF</th>
<th>Decompose R in BCNF</th>
</tr>
</thead>
<tbody>
<tr>
<td>{A \rightarrow B, C, D, E}</td>
<td>{A}</td>
<td>R(A,B,C,D,E)</td>
<td>R(A,B,C,D,E)</td>
</tr>
</tbody>
</table>
## Problem 3 – Normal Forms

<table>
<thead>
<tr>
<th>Functional Dependencies</th>
<th>Candidate Keys for R</th>
<th>Decompose R in 3NF</th>
<th>Decompose R in BCNF</th>
</tr>
</thead>
<tbody>
<tr>
<td>{ C \rightarrow D }</td>
<td>ABCE</td>
<td>R1(A,B,C,E)</td>
<td>R1(A,B,C,E)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>R2(C,D)</td>
<td>R2(C,D)</td>
</tr>
<tr>
<td>{ C \rightarrow D, D \rightarrow C }</td>
<td>ABCE, ABDE</td>
<td>R(A,B,C,D,E)</td>
<td>R1(C,D)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>R2(A,B,C,E) or</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>R1(C,D)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>R2(A,B,D,E)</td>
</tr>
<tr>
<td>{ A \rightarrow B, C \rightarrow D }</td>
<td>ACE</td>
<td>R1(A,B)</td>
<td>R1(A,B)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>R2(C,D)</td>
<td>R2(C,D)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>R3(A,C,E)</td>
<td>R3(A,C,E)</td>
</tr>
<tr>
<td>{ A \rightarrow BC, D \rightarrow AE }</td>
<td>D</td>
<td>R1(A,B,C)</td>
<td>R1(D,A,E)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>R2(D,A,E)</td>
<td>R2(A,B,C)</td>
</tr>
</tbody>
</table>