BSCCS2001: Practice Assignment with Solutions
Week 4

1. Consider the following relations : [MCQ: 2 points]
A = (P, Q, R)
B = (X, Y, Z)
Let relations $a(A)$ and $b(B)$ be given. Which of the following expressions in the tuple relational calculus is equivalent to $\Pi_{P,Z}(\sigma_{R=X}(a \times b))$?

   - √ $\{t \mid \exists p \in a, \exists q \in b(t[P] = p[P] \wedge t[Z] = q[Z] \wedge p[R] = q[X])\}$
   - ◯ $\{t \mid \exists p \in a, \exists q \in b(t[P] = p[P] \wedge t[Z] = q[Z] \vee p[R] = q[X])\}$
   - ◯ $\{t \mid \exists p \in a, \exists q \in b(t[P] = p[P] \wedge t[Z] = q[Z] \wedge p[X] = q[R])\}$
   - ◯ $\{t \mid \exists p \in a, \exists q \in b(t[Z] = p[Z] \wedge t[P] = q[P] \wedge p[R] = q[X])\}$

Solution:
$\Pi_{P,Z}(\sigma_{R=X}(a \times b))$ will return attributes P from relation $a(A)$ and Z from relation $b(B)$. So, options C and D are incorrect.
For SELECT operation, condition $R = X$ must be satisfied, So option B is incorrect.
Thus, option 1 is correct.

Consider the following relational schema and answer questions 2 and 3.

[MCQ:2 points]

- $Owner(aadhar\_number, o\_name)$
- $Vehicle(v\_number, v\_model)$
- $Registration(aadhar\_number, v\_number, purchase\_year)$

2. Which of the following relational algebra expressions is equivalent to the statement given below?

   - Find the Aadhaar numbers of owners who purchased the vehicle model $V20$ after year 2020.
     - ◯ $\sigma_{aadhar\_number}(\prod_{v\_model=\text{“}V20\text{”} \vee purchase\_year>\text{”}2020\text{”}}(Registration \bowtie Vehicle))$
     - ◯ $\sigma_{aadhar\_number}(\sigma_{v\_model=\text{“}V20\text{”} \wedge purchase\_year>\text{”}2020\text{”}}(Registration \bowtie Vehicle))$
     - √ $\prod_{aadhar\_number}(\sigma_{v\_model=\text{“}V20\text{”} \wedge purchase\_year>\text{”}2020\text{”}}(Registration \bowtie Vehicle))$
     - ◯ $\prod_{aadhar\_number}(\sigma_{v\_model=\text{“}V20\text{”} \vee purchase\_year>\text{”}2020\text{”}}(Registration \bowtie Vehicle))$

Solution: Selection Operator (σ), selects those rows or tuples from the relation that satisfies the selection condition.
Project operator is denoted by the symbol Π, and it is used to select desired columns (or attributes) from a table (or relation).
Option 1 and Option 2 are incorrect. Here, the SELECT operator is used, requiring a specific condition to select tuples from a relation.
Option 3: It will return the Aadhaar number of all the owners who purchased the vehicle model $V20$ **and** after year 2020.
So, Option 3 is correct.
Option 4: It will return the Aadhaar number of all the owners who purchased the vehicle model $V20$ **or** after year 2020.

3. Which of the following queries is equivalent to the statement given below?

   - Find the names of all owners who purchased vehicles with number 123 before the year 2019.

     ◯ $\{T \mid \exists O \in Owner, \exists R \in Registration(O.aadhar\_number = R.aadhar\_number \wedge R.v\_number = 123 \vee R.purchase\_year < 2019 \wedge T.o\_name = O.o\_name)\}$

     ◯ $\{T \mid \exists O \in Owner, \exists R \in Registration(O.aadhar\_number = R.aadhar\_number \wedge R.v\_number = 123 \vee R.purchase\_year < 2019 \vee T.o\_name = O.o\_name)\}$

     ◯ $\{T \mid \exists O \in Owner, \exists R \in Registration(O.aadhar\_number = R.aadhar\_number \vee R.v\_number = 123 \wedge R.purchase\_year < 2019 \vee T.o\_name = O.o\_name)\}$

     √ $\{T \mid \exists O \in Owner, \exists R \in Registration(O.aadhar\_number = R.aadhar\_number \wedge R.v\_number = 123 \wedge R.purchase\_year < 2019 \wedge T.o\_name = O.o\_name)\}$

Solution:
$\exists O \in Owner, \exists R \in Registration(O.aadhar\_number = R.aadhar\_number)$ will perform the NATURAL JOIN operation of Owner and Registration schema and $(R.v\_number = 123 \wedge R.purchase\_year < 2019 \wedge T.o\_name = O.o\_name)$ is the required condition for the names of all the owners who purchased the vehicle number 123 before year 2019.

4. Consider the E-R diagram for a cricket-training-camp database as given in Figure 1.

[MCQ: 2 points]

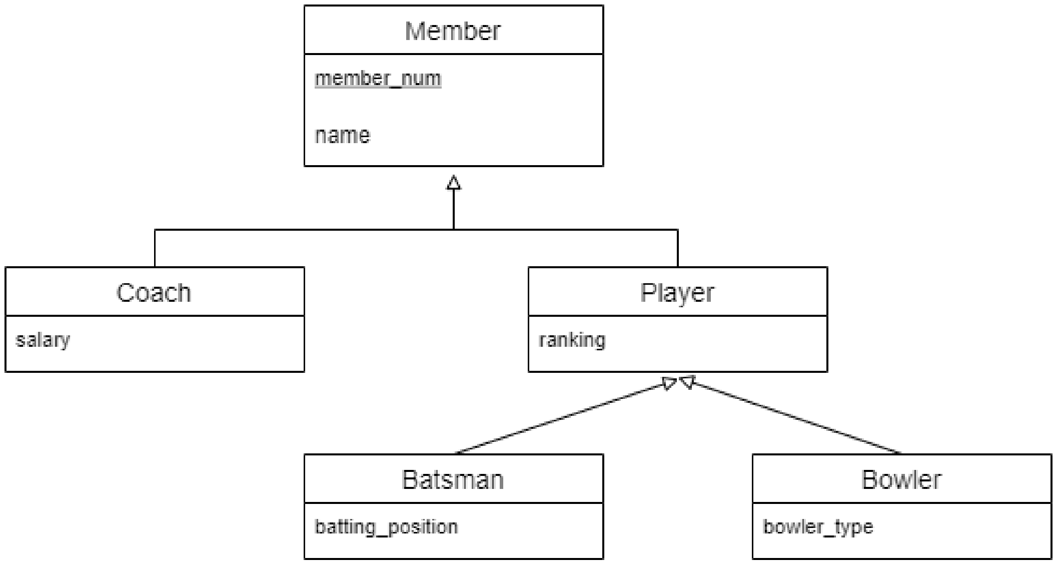

Figure 1: E-R diagram of cricket-training-camp database

Identify the option in which both the statements correctly describe the relations between the given entity sets.

- ◯ 1. Each member can be either a coach or a player or both in the cricket-training-camp.
  2. Each player can be a batsman or a bowler or both.
- ◯ 1. Each member can be a coach or a player or both.
  2. Each player can be either a batsman or a bowler. However, a player cannot be both, a batsman and a bowler at the same time.
- ◯ 1. Each member can be either a coach or a player. But, a member cannot be a coach and a player at the same time.
  2. Each player can be either a batsman or a bowler, but cannot be both.
- √ 1. Each member can be either a coach or a player or just a member of the cricket-training-camp. But, a member cannot be a coach and a player at the same time.
  2. Each player can be a batsman or a bowler or both.

Solution:

- **Coach** and **Player** are disjoint specializations of **Member**.

- **Batsman** and **Bowler** are overlapping specializations of **Member**.
- Both kind of specializations given in Figure 1 are partial specializations.

Hence,

- Each member must be a coach or a player or just a member. But, a member cannot be a coach and a player at the same time.
- Each player can be a batsman or a bowler or both.

5. Consider the E-R diagram given in Figure 2.

[MCQ: 2 points:Solve with instructor]

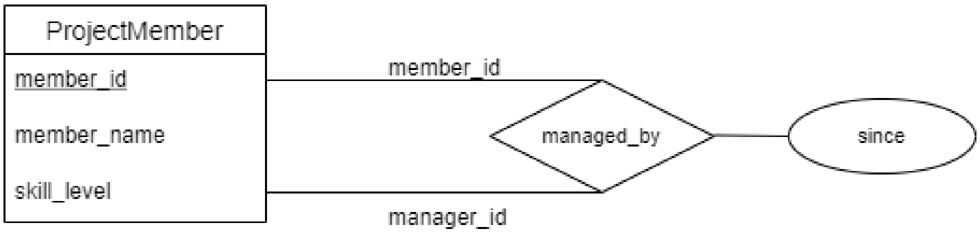

Figure 2: E-R diagram

The table for entity set **ProjectMember** is created using the command below:

```
CREATE TABLE ProjectMember(
member_id INT NOT NULL,
member_name VARCHAR(20) NOT NULL,
skill_level VARCHAR(20) NOT NULL,
PRIMARY KEY (member_id)
);
```

Select the appropriate command to create the table for relationship set **managed_by**.

○
```
CREATE TABLE managed_by(
member_id INT NOT NULL,
since INT NOT NULL,
PRIMARY KEY (member_id),
FOREIGN KEY (member_id) REFERENCES ProjectMember(member_id)
);
```

√
```
CREATE TABLE managed_by(
member_id INT,
manager_id INT,
since INT NOT NULL,
PRIMARY KEY (member_id, manager_id),
FOREIGN KEY (member_id) REFERENCES ProjectMember(member_id),
FOREIGN KEY (manager_id) REFERENCES ProjectMember(member_id)
);
```

○
```
CREATE TABLE managed_by(
manager_id INT NOT NULL,
since INT NOT NULL,
PRIMARY KEY (manager_id),
```

```
FOREIGN KEY (manager_id) REFERENCES ProjectMember(member_id)
);
```

- ◯
```
CREATE TABLE managed_by(
member_id INT NOT NULL,
manager_id INT NOT NULL,
since INT NOT NULL,
PRIMARY KEY (member_id, manager_id),
FOREIGN KEY (manager_id) REFERENCES ProjectMember(member_id)
);
```

Solution: The table **managed_by** must have {*member_id*, *manager_id*} as primary key, both reference to **ProjectMember**(*member_id*), and the descriptive attribute *since* also becomes an attribute in the table.
Thus, it must be created by the command:

```
CREATE TABLE managed_by(
member_id INT,
manager_id INT,
since INT NOT NULL,
PRIMARY KEY (member_id, manager_id),
FOREIGN KEY (member_id) REFERENCES ProjectMember(member_id),
FOREIGN KEY (manager_id) REFERENCES ProjectMember(member_id)
);
```

Please note that since {*member_id, manager_id*} is the primary key, the prime attributes *member_id* and *manager_id* by default not `NULL`.

6. Consider the entity set given in Figure 3. [MCQ: 2 points]

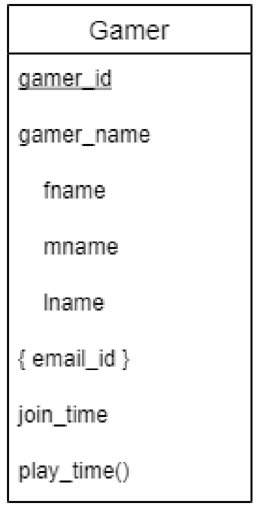

Figure 3: Entity set **Gamer**

Which of the following relational schemas appropriately represents the E-R diagram?

- ◯ **Gamer**($\underline{gamer\_id}, gamer\_name, fname, mname, lname, email\_id, join\_time$)
- ◯ **Gamer**($\underline{gamer\_id}, fname, mname, lname, email\_id, join\_time, play\_time$)
  Gamer_email($\underline{gamer\_id}, email\_id$)
- √ **Gamer**($\underline{gamer\_id}, fname, mname, lname, join\_time$)
  Gamer_email($\underline{gamer\_id}, \underline{email\_id}$)
- ◯ **Gamer**($\underline{gamer\_id}, email\_id, join\_time$)
  Gamer_name($\underline{gamer\_id}, gamer\_name, fname, mname, lname$)

Solution:

- The identifying attribute *gamer_id* becomes primary key for the schema.
- The composite attribute *gamer_name* will be replaced by its parts *fname*, *mname* and *lname* in the schema.
- The simple attribute *join_time* becomes another attribute.
- The derived attribute *play_time* does not need to be added in the schema.
- For the multivalued attribute *email_id* a separate relation has to be created which will be:
  $Gamer\_email(\underline{gamer\_id}, \underline{email\_id})$.

7. Consider the relations given below: [MSQ: 2 points:Solvewithinstructor]

   - **doctor**($\underline{doc\_id}$, doc_name, $specialization$)
   - **patient**($\underline{patient\_num}$, $patient\_name$)
   - **operationRoster**($\underline{doc\_id}$, $\underline{patient\_num}$, $operation\_cost$)

   Identify the appropriate statement(s) to find the names of all doctors having specialization in orthopedics and who have charged more than $1000 as surgery charges.

   √ $\prod_{doc\_name}(\sigma_{specialization=\text{“}orthopedic\text{”} \ \wedge \ operation\_cost>1000}(doctor \bowtie operationRoster))$

   ◯ $\{s \mid \exists s \in doctor,\ \exists r \in operationRoster(s.doc\_id = r.doc\_id \wedge s.specialization = \text{“}orthopedic\text{”} \wedge r.operation\_cost > 1000)\}$

   √ $\{t \mid \exists s \in doctor,\ \exists r \in operationRoster(s.doc\_id = r.doc\_id \wedge s.specialization = \text{“}orthopedic\text{”} \ \wedge r.operation\_cost > 1000 \ \wedge \ t.doc\_name = s.doc\_name)\}$

   √ $\{< D_N > \ \mid \exists D_I \exists R_P \exists R_C (< D_I, D_N, \text{“}orthopedic\text{”} > \in doctor \wedge < D_I, R_P, R_C > \in operationRoster) \wedge R_C > 1000\}$

Solution: As per the specifications given in the question, a natural join needs to be applied between **doctor** and **operationRoster** as:
$doctor \bowtie operationRoster$.
Then, a select operation can be applied as:
$\sigma_{specialization=\text{“}orthopedic\text{”} \ \wedge \ operation\_cost>1000}(doctor \bowtie operationRoster)$.
Finally, apply project the doc_name as:
$\prod_{doc\_name}(\sigma_{specialization=\text{”}orthopedic\text{”} \ \wedge \ operation\_cost>1000}(doctor \bowtie operationRoster))$.
The equivalent **tuple relational calculus** is:
$\{t \mid \exists s \in doctor,\ \exists r \in operationRoster(s.specialization = \text{“}orthopedic\text{”} \wedge r.operation\_cost > 1000 \ \wedge \ t.doc\_name = s.doc\_name)\}$.
The equivalent **domain relational calculus** is:
$\{< D_N > \ \mid \exists D_I \exists R_P \exists R_C (< D_I, D_N, \text{“}orthopedic\text{”} > \in doctor \wedge < D_I, R_P, R_C > \in operationRoster) \wedge R_C > 1000\}$.
Please note the tuple relation calculus:
$\{s \mid \exists s \in doctor \exists \ r \in operationRoster(s.doc\_id = r.doc\_id \wedge s.specialization = \text{“}orthopedic\text{”} \wedge r.operation\_cost > 1000)\}$
projects all attributes rather than doc_name alone.

8. Consider the relations below: [MSQ: 2 points]

   - **customer**($\underline{customer\_id}$, $customer\_name$, $customer\_city$)
   - **invoice**($\underline{invoice\_number}$, $\underline{customer\_id}$, $amount\_payable$)

   Choose the correct relational algebra expressions that return the names of all customers having amount payable ($amount\_payable$) more than $1,000 and who are located in **Chennai**.

○ $\prod_{customer\_name}(\sigma_{amount\_payable>1000 \ \wedge \ customer\_city=\text{“}Chennai\text{”}}(customer \times invoice))$

○ $\prod_{amount\_payable>1000 \ \vee \ customer\_city=\text{“}Chennai\text{”}}(customer \bowtie invoice))$

√ $\prod_{customer\_name}(\sigma_{amount\_payable>1000 \ \wedge \ customer\_city=\text{“}Chennai\text{”}}(customer \bowtie invoice))$

√ $\prod_{customer\_name}$
$(\sigma_{amount\_payable>1000 \ \wedge \ customer\_city=\text{“}Chennai\text{”} \ \wedge \ customer.customer\_id=invoice.customer\_id}$
$(customer \times invoice))$

Solution: First, natural join can be applied between the two relations **customer** and **invoice** such that tuples will be combined by equality in *customer_id*. Then, σ with predicate $amount\_payable > 1000 \ \wedge \ customer\_city = \text{“}Chennai\text{”}$ can be applied to find the tuples as per the specification given.
Alternatively, the same can also be achieved by a Cartesian product between **customer** and **invoice** along with σ with predicate equality of *customer_id* of both the relations and σ with predicate $amount\_payable > 1000 \ \wedge \ customer\_city = \text{“}Chennai\text{”}$.
Finally, we project ($\prod$) the *customer_name*.

9. Consider the table given in Figure 4. [MSQ: 2 points]

S1	
A	B
5	25
6	36
7	49
8	64
9	91

Figure 4: Relation **S1**

Choose the correct set of expressions that will return the tuple given below.

A
7

○ $\sigma_A(\Pi_{B=49}(S1))$

√ $\{t \mid \exists p \in S1(t[A] = p[A] \wedge p[B] = 49)\}$

○ $\{t \mid \exists p \in S1(t[A] = p[A] \wedge p[B] = 7)\}$

√ $\{< a > \mid \exists b(< a, b > \in S1 \wedge b = 49)\}$

Solution:
$\sigma_A(\Pi_{B=49}(S1))$ is logically incorrect, the correct TRC query is $\Pi_A(\sigma_{B=49}(S1))$, this will first perform the Select operation and return the row having B = 49 then it will project the corresponding value of attribute A.

$\{t \mid \exists p \in S1(t[A] = p[A] \wedge p[B] = 49)\}$ is equivalent to $\Pi_A(\sigma_{B=49}(S1))$

$\{< a > \mid \exists b(< a, b > \in S1 \wedge b = 49)\}$ is equivalent to $\Pi_A(\sigma_{B=49}(S1))$

10. Consider the E-R diagram in Figure 5. [NAT: 2 points]

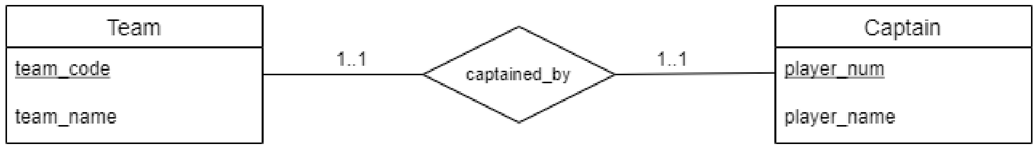

Figure 5: ERD

What is the minimum number of tables needed to represent this E-R diagram?

Solution: 1
The minimum and maximum cardinality is 1 (1..1).

- A minimum value of 1 indicates total participation.
- A maximum value of 1 indicates that the entity participates in at most one relationship.

Thus, it can be represented using a single table:
team_captain($\underline{team\_code}$, $team\_name$, $\underline{player\_num}$, $player\_name$).

Consider the E-R diagram given in Figure 6 and answer the questions 11 and 12.

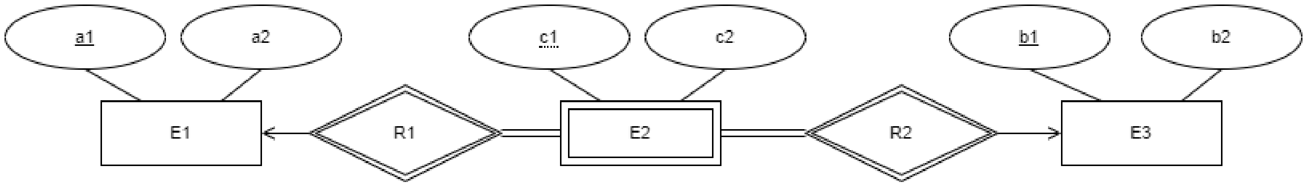

Figure 6: E-R diagram

11. The minimum number of tables required to represent the entity sets and relationship sets is .... [NAT: 2 points]

Answer: 3

Solution: 3

- **E1** is associated with **E2** via **R1** in a one-to-many relation.
- **E3** is associated with **E2** via **R2** in a one-to-many relation.

Many-to-one and one-to-many relationship sets that are total on the many-side can be represented by adding an extra attribute to the "many" side, containing the primary key of the "one" side. Thus, we can represent the entire ERD using 3 tables as follows:

- **E1**$(\underline{a1}, a2)$
- **E2**$(\underline{c1}, c2, \underline{a1}, \underline{b1})$
- **E3**$(\underline{b1}, b2)$

12. What will be the correct attribute set for the table corresponding to the entity set **E2**? [MCQ: 2 points:Solvewithinstructor]

◯ **E2**$(\underline{c1}, c2)$

◯ **E2**$(\underline{c1}, \underline{a1}, c2)$

◯ **E2**$(\underline{c1}, a1, b1, c2)$

√ **E2**$(\underline{c1}, c2, \underline{a1}, \underline{b1})$

Solution:

- **R1** is a one-to-many relationship set from **E2** to **E1**.
- **R2** is a one-to-many relationship set from **E2** to **E3**.

Many-to-one and one-to-many relationship sets that are total on the many-side can be represented by adding an extra attribute to the "many" side, containing the primary key of the "one" side. Thus, we can represent the entire ERD using 3 tables as follows:

- **E1**($\underline{a1}, a2$)
- **E2**($\underline{c1}, c2, \underline{a1}, \underline{b1}$)
- **E3**($\underline{b1}, b2$)

13. Consider the E-R diagram with aggregation given in Figure 7. [MCQ: 2 points]

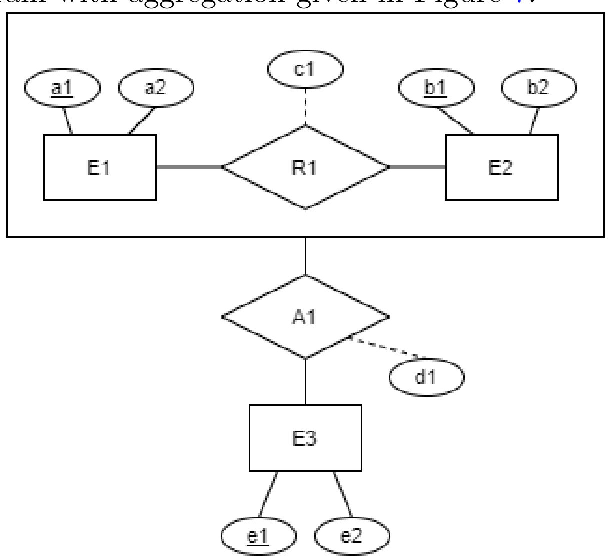

Figure 7: ERD

What will be the correct attribute set for the table corresponding to relationship-set **A1**?

○ $c1, e1, d1$

○ $a1, b1, d1, c1, e1, e2$

○ $a1, b1, d1, e1, e2$

√ $a1, b1, e1, d1$

Solution: The ER-diagram presents a scenario of aggregation. Thus, the relationship set **A1** must be mapped to a table having the following:

- Primary keys of **E1**, **E2** and **E3**.
- Any descriptive attributes of **A1**.

So the attribute set for **A1** is: $\{a1, b1, e1, d1\}$