CASE STUDIES

of

Database Management Systems Laboratory

(CS238)

Bachelor of Technology (CSE)

By

Ramoliya Kaushal (22000409)

Second Year, Semester 4

Course In-charge: Prof. Ujwala Petigara

NAVRACHANA UNIVERSITY

a UGC recognized University

Department of Computer Science and Engineering

School Engineering and Technology

Navrachana University, Vadodara

Spring Semester

(2024)

CASE STUDIES

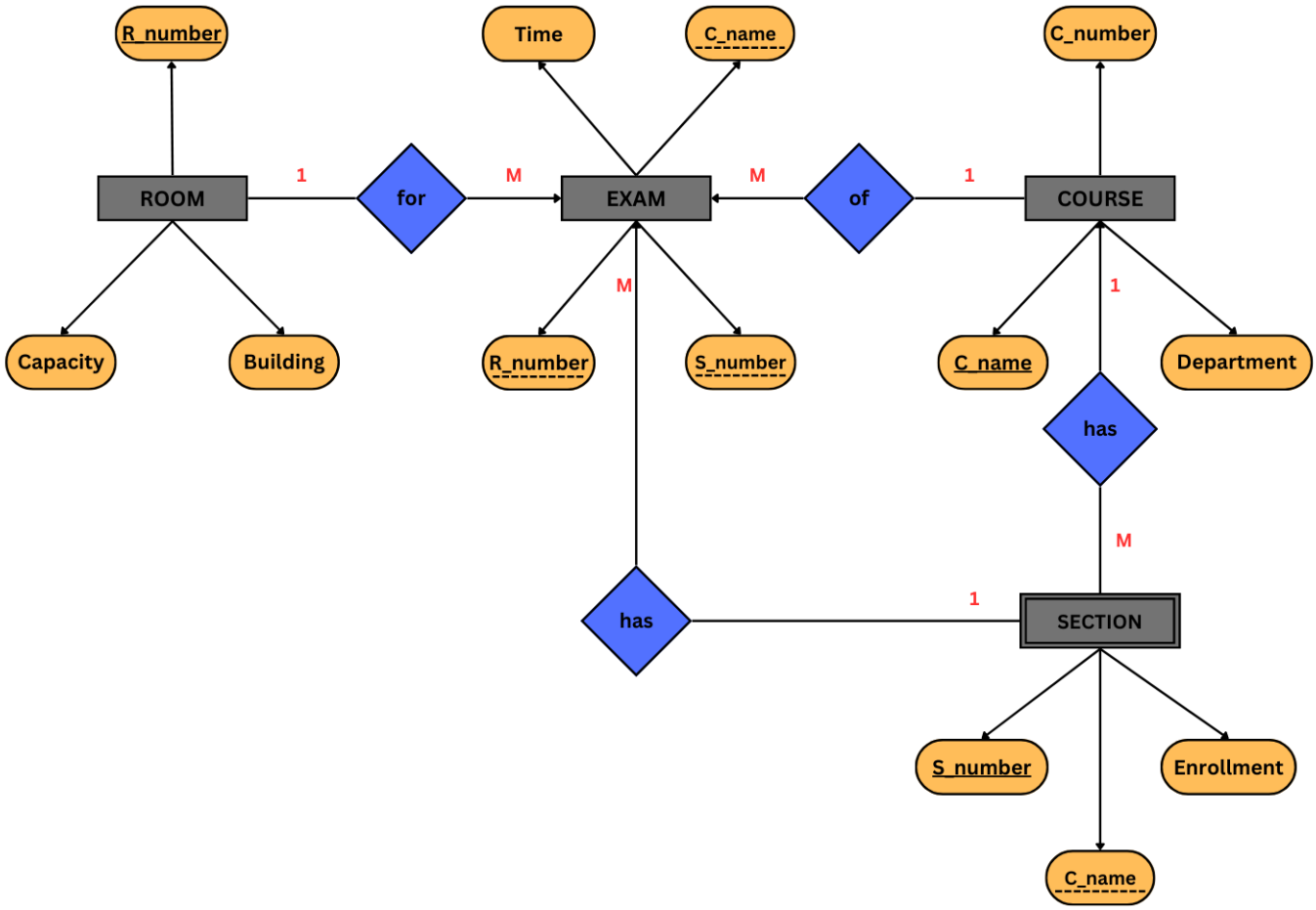
Design the E-R diagrams as per the instructions. Also try to incorporate mapping cardinalities and later on convert the ER diagram into actual tables and write the SQL queries for the creation of those tables.

Question 9

Consider a university database for the scheduling of classrooms for -final exams. This database could be modeled as the single entity set exam, with attributes course-name, section number, room number, and time. Alternatively, one or more additional entity sets could be defined, along with relationship sets to replace some of the attributes of the exam entity set, as course with attributes name, department, and c-number, section with attributes s-number and enrollment, and dependent as a weak entity set on course , room with attributes r-number, capacity, and building Show an E-R diagram illustrating the use of all three additional entity sets listed.

INDEX

E-R Diagram



Description of E-R Diagram

Entity:-

1. Exam
2. Room
3. Course
4. Section

Strong Entities:-

1. Course

2. Room

3. Exam

Weak Entities:-

1. Section

Attributes:-

1. **Exam**:- C_name,S_number,R_number,Time.

4. **Room**:- R_number,Capacity,Building.

2. **Course**:- C_name,C_number,Department.

3. **Section**:- S_number,Enrollment.

Primary Key:-

1. **Course:-** C_name (Primary Key)

2. **Section:-** S_number (Primary Key)

3. **Room:-** R_number (Primary Key)

Foreign Key:-

1. **Exam:-** C_name (Foreign Key) - References Course.C_name

2. **Exam:-** S_number (Foreign Key) - References Section.S_number

3. **Exam:-** R_number (Foreign Key) - References Room.R_number

4. **Section:-** C_name (Foreign Key) - References Course.C_name

Cardinality

1. **Exam to Room:-** Many-to-One
    - One exam can be held in one room, but a room can host many exams.
2. **Exam to Course:-** Many-to-One
    - One exam can belongs to one course, but a course can be part of many exams throughout the semester.
3. **Course to Section:-** One-to-Many
    - one course can have multiple sections associated with it, but each section belongs to only one course.
4. **Exam to Section:-** Many-to-One
    - many exams can belong to one section, but each exam is associated with only one section.

Creating Tables and Insert Data

Creating Tables:-

Step-1: create database university;

```
mysql> create database university;
Query OK, 1 row affected (0.08 sec)

mysql> use university;
Database changed
mysql>
```

Step-2: Create table Room:

Query: create table room(R_number int,Capacity int,Building varchar(20),primary key (R_number));

Output:

```
mysql> create table room(R_number int,Capacity int,Building varchar(20),primary key (R_number) );
Query OK, 0 rows affected (0.03 sec)
```

Show table:

```
mysql> desc room;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| R_number | int         | NO   | PRI | NULL    |       |
| Capacity | int         | YES  |     | NULL    |       |
| Building | varchar(20) | YES  |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
3 rows in set (0.04 sec)
```

Step-3: Create table Course:

Query: create table course(C_number int,C_name varchar (30),Department varchar (20),primary key (C_name));

Output:

```
mysql> create table course(C_number int,C_name varchar (30),Department varchar (20),primary key (C_name) );
Query OK, 0 rows affected (0.04 sec)
```

Show table:

```
mysql> desc course;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| C_number   | int         | YES  |     | NULL    |       |
| C_name     | varchar(30) | NO   | PRI | NULL    |       |
| Department | varchar(20) | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
3 rows in set (0.03 sec)
```

Step-4: Create table Section:

Query: create table section(S_number int,Enrollment int,C_name varchar(30), primary key(S_number), foreign key(C_name) references section(C_number));

Output:

```
mysql> create table section(S_number int,Enrollment int, primary key(S_number) );
Query OK, 0 rows affected (0.03 sec)
```

Show table:

```
mysql> desc section;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| S_number   | int         | NO   | PRI | NULL    |       |
| Enrollment | int         | YES  |     | NULL    |       |
| C_name     | varchar(30) | YES  | MUL | NULL    |       |
+------------+-------------+------+-----+---------+-------+
3 rows in set (0.01 sec)
```

Step-5: Create table Exam:

Query: create table exam(time time,C_name varchar(30),R_number int,S_number int, foreign key(C_name) references course(C_name),foreign key(R_number) referen

ces room(R_number), foreign key(S_number) references section(S_number));

Output:

```
mysql> create table exam(time time,C_name varchar(30),R_number int,S_number int, foreign key(C_name) references course(C_name),foreign key(R_number) referen
ces room(R_number), foreign key(S_number) references section(S_number));
Query OK, 0 rows affected (0.07 sec)
```

Show table:

```
mysql> desc exam;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| time     | time        | YES  |     | NULL    |       |
| C_name   | varchar(30) | YES  | MUL | NULL    |       |
| R_number | int         | YES  | MUL | NULL    |       |
| S_number | int         | YES  | MUL | NULL    |       |
+----------+-------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

Insert Data:-

1. Room

Query: INSERT INTO room VALUES

-> (101, 50, 'Building A'),

-> (102, 40, 'Building B'),

-> (103, 30, 'Building A');

Output:

```
mysql> INSERT INTO room VALUES
    -> (101, 50, 'Building A'),
    -> (102, 40, 'Building B'),
    -> (103, 30, 'Building A');
Query OK, 3 rows affected (0.02 sec)
Records: 3  Duplicates: 0  Warnings: 0
```

2. Course

Query: INSERT INTO course VALUES

-> (238, 'Database Management System', 'Computer Science'),

-> (305, 'Core JAVA Programing', 'Computer Science'),

-> (251, 'Operating System', 'Computer Science');

Output:

```
mysql> INSERT INTO course VALUES
    -> (238, 'Database Management System', 'Computer Science'),
    -> (305, 'Core JAVA Programing', 'Computer Science'),
    -> (251, 'Operating System', 'Computer Science');
Query OK, 3 rows affected (0.01 sec)
Records: 3  Duplicates: 0  Warnings: 0
```

3. Section

Query: INSERT INTO section VALUES

-> (1, 409, 'Database Management System'),

-> (2, 404, 'Database Management System'),

-> (3, 407, 'Database Management System');

Output:

```
mysql> INSERT INTO section VALUES
    -> (1, 409, 'Database Management System'),
    -> (2, 404, 'Database Management System'),
    -> (3, 407, 'Database Management System');
Query OK, 3 rows affected (0.01 sec)
Records: 3  Duplicates: 0  Warnings: 0
```

4. Exam

Query: INSERT INTO exam VALUES

-> ('09:00:00', 'Database Management System', 101, 1),

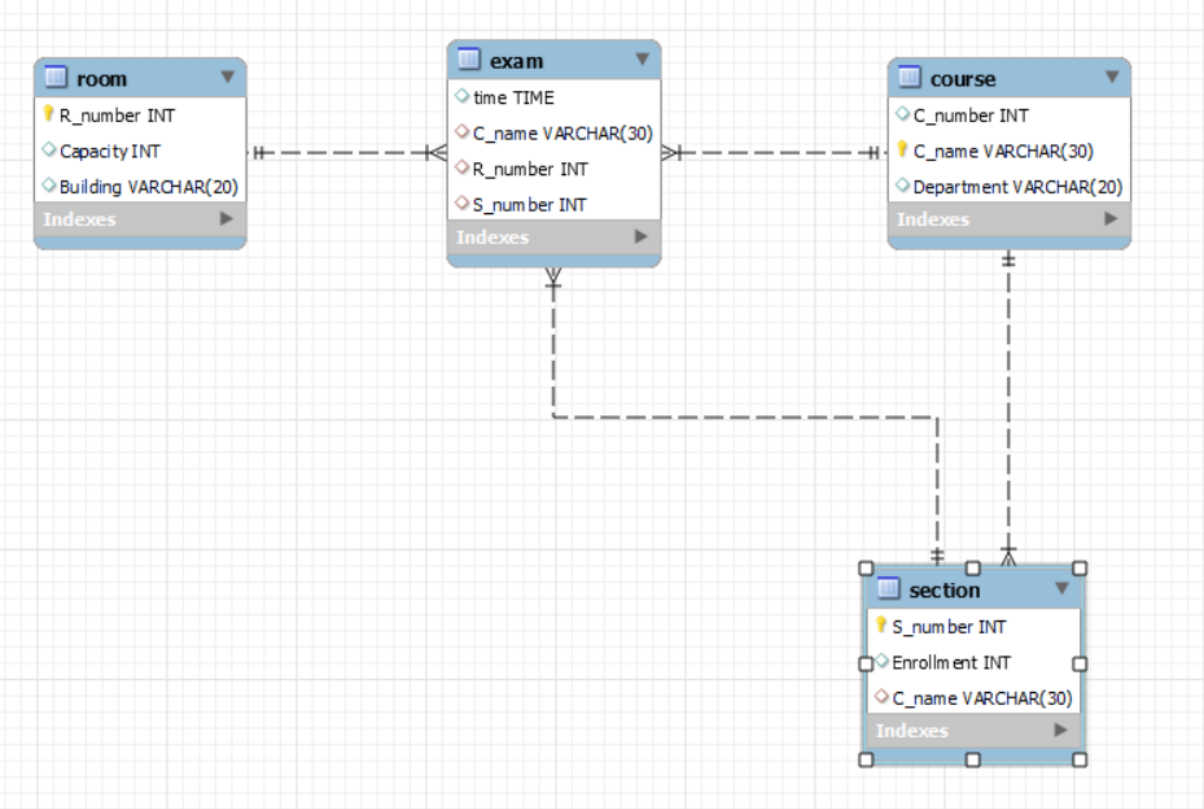
-> ('11:00:00', 'Core JAVA Programing', 102, 2),

-> ('13:00:00', 'Operating System', 103, 3);

Output:

```
mysql> INSERT INTO exam VALUES
    -> ('09:00:00', 'Database Management System', 101, 1),
    -> ('11:00:00', 'Core JAVA Programing', 102, 2),
    -> ('13:00:00', 'Operating System', 103, 3);
Query OK, 3 rows affected (0.01 sec)
Records: 3  Duplicates: 0  Warnings: 0
```

Reduction of ER diagram to Table



Queries

Query1: SELECT * FROM room;

Output:

```
mysql> SELECT * FROM room;
+----------+----------+------------+
| R_number | Capacity | Building   |
+----------+----------+------------+
|      101 |       50 | Building A |
|      102 |       40 | Building B |
|      103 |       30 | Building A |
+----------+----------+------------+
3 rows in set (0.00 sec)
```

Query2: SELECT C_name, Department FROM course;

Output:

```
mysql> SELECT C_name, Department FROM course;
+----------------------------+------------------+
| C_name                     | Department       |
+----------------------------+------------------+
| Core JAVA Programing       | Computer Science |
| Database Management System | Computer Science |
| Operating System           | Computer Science |
+----------------------------+------------------+
3 rows in set (0.00 sec)
```

Query3: SELECT DISTINCT Department FROM course;

Output:

```
mysql> SELECT DISTINCT Department FROM course;
+------------------+
| Department       |
+------------------+
| Computer Science |
+------------------+
1 row in set (0.01 sec)
```

Query4: SELECT DISTINCT C_name FROM course;

Output:

```
mysql> SELECT DISTINCT C_name FROM course;
+---------------------------+
| C_name                    |
+---------------------------+
| Core JAVA Programing      |
| Database Management System |
| Operating System          |
+---------------------------+
3 rows in set (0.00 sec)
```

Query5: SELECT R_number FROM room WHERE Capacity = 40;

Output:

```
mysql> SELECT R_number FROM room WHERE Capacity = 40;
+----------+
| R_number |
+----------+
|      102 |
+----------+
1 row in set (0.00 sec)
```

Query6: SELECT R_number FROM room WHERE Building = 'Building A';

Output:

```
mysql> SELECT R_number FROM room WHERE Building = 'Building A';
+----------+
| R_number |
+----------+
|      101 |
|      103 |
+----------+
2 rows in set (0.00 sec)
```

Query7: SELECT C_name FROM course ORDER BY C_name;

Output:

```
mysql> SELECT C_name FROM course ORDER BY C_name;
+----------------------------+
| C_name                     |
+----------------------------+
| Core JAVA Programing       |
| Database Management System |
| Operating System           |
+----------------------------+
3 rows in set (0.00 sec)
```

Query 8: SELECT S_number FROM section ORDER BY Enrollment DESC;

Output:

```
mysql> SELECT S_number FROM section ORDER BY Enrollment DESC;
+----------+
| S_number |
+----------+
|        1 |
|        3 |
|        2 |
+----------+
3 rows in set (0.01 sec)
```

Query 9: SELECT R_number FROM room WHERE Capacity > 30 AND Building = 'Building B';

Output:

```
mysql> SELECT R_number FROM room WHERE Capacity > 30 AND Building = 'Building B';
+----------+
| R_number |
+----------+
|      102 |
+----------+
1 row in set (0.00 sec)
```

Query 10: SELECT R_number FROM room WHERE Capacity = 40 OR Capacity = 50;

Output:

```
mysql> SELECT R_number FROM room WHERE Capacity = 40 OR Capacity = 50;
+----------+
| R_number |
+----------+
|      101 |
|      102 |
+----------+
2 rows in set (0.00 sec)
```

Query11: SELECT C_name FROM course WHERE Department = 'Computer Science' OR Department = 'Database Management System';

Output:

```
mysql> SELECT C_name FROM course WHERE Department = 'Computer Science' OR Department = 'Database Management System';
+--------------------------+
| C_name                   |
+--------------------------+
| Core JAVA Programing     |
| Database Management System |
| Operating System         |
+--------------------------+
3 rows in set (0.00 sec)
```

Query12: SELECT R_number FROM room WHERE NOT Capacity = 50;

Output:

```
mysql> SELECT R_number FROM room WHERE NOT Capacity = 50;
+----------+
| R_number |
+----------+
|      102 |
|      103 |
+----------+
2 rows in set (0.00 sec)
```

Query13: SELECT S_number FROM section WHERE NOT Enrollment = 407;

Output:

```
mysql> SELECT S_number FROM section WHERE NOT Enrollment = 407;
+----------+
| S_number |
+----------+
|        1 |
|        2 |
+----------+
2 rows in set (0.00 sec)
```

Query14: SELECT * FROM room WHERE NOT Building = 'Building B';

Output:

```
mysql> SELECT * FROM room WHERE NOT Building = 'Building B';
+----------+----------+------------+
| R_number | Capacity | Building   |
+----------+----------+------------+
|      101 |       50 | Building A |
|      103 |       30 | Building A |
+----------+----------+------------+
2 rows in set (0.00 sec)
```

Query15: SELECT * FROM course WHERE NOT C_name IN ('Database Management System', 'Core JAVA Programing');

Output:

```
mysql> SELECT * FROM course WHERE NOT C_name IN ('Database Management System', 'Core JAVA Programing');
+----------+------------------+------------------+
| C_number | C_name           | Department       |
+----------+------------------+------------------+
|      251 | Operating System | Computer Science |
+----------+------------------+------------------+
1 row in set (0.01 sec)
```

Query16: UPDATE room SET Capacity = 60 WHERE R_number = 101;

Output:

```
mysql> select * from room;
+----------+----------+------------+
| R_number | Capacity | Building   |
+----------+----------+------------+
|      101 |       60 | Building A |
|      102 |       40 | Building B |
|      103 |       30 | Building A |
+----------+----------+------------+
3 rows in set (0.00 sec)
```

Query17: UPDATE exam SET time = '08:00:00' WHERE C_name = 'Operating System';

Output:

```
mysql> select * from exam;
+----------+----------------------------+----------+----------+
| time     | C_name                     | R_number | S_number |
+----------+----------------------------+----------+----------+
| 09:00:00 | Database Management System |      101 |        1 |
| 11:00:00 | Core JAVA Programing       |      102 |        2 |
| 08:00:00 | Operating System           |      103 |        3 |
+----------+----------------------------+----------+----------+
3 rows in set (0.00 sec)
```

Query18: SELECT MIN(Capacity) AS Min_Capacity FROM room;

Output:

```
mysql> SELECT MIN(Capacity) AS Min_Capacity FROM room;
+--------------+
| Min_Capacity |
+--------------+
|           30 |
+--------------+
1 row in set (0.01 sec)
```

Query19: SELECT MAX(C_number) AS Max_Course_Number FROM course;

Output:

```
mysql> SELECT MAX(C_number) AS Max_Course_Number FROM course;
+-------------------+
| Max_Course_Number |
+-------------------+
|               305 |
+-------------------+
1 row in set (0.00 sec)
```

Query20: SELECT COUNT(*) AS Total_Rooms FROM room;

Output:

```
mysql> SELECT COUNT(*) AS Total_Rooms FROM room;
+--------------+
| Total_Rooms  |
+--------------+
|            3 |
+--------------+
1 row in set (0.01 sec)
```

Query21: SELECT COUNT(*) AS Total_Sections FROM section;

Output: :

```
mysql> SELECT COUNT(*) AS Total_Sections FROM section;
+----------------+
| Total_Sections |
+----------------+
|              3 |
+----------------+
1 row in set (0.01 sec)
```

Query 22: SELECT COUNT(*) AS Total_Exams FROM exam;

Output:

```
mysql> SELECT COUNT(*) AS Total_Exams FROM exam;
+-------------+
| Total_Exams |
+-------------+
|           3 |
+-------------+
1 row in set (0.00 sec)
```

Query 23: SELECT SUM(Enrollment) AS Total_Enrollment FROM section;

Output:

```
mysql> SELECT SUM(Enrollment) AS Total_Enrollment FROM section;
+------------------+
| Total_Enrollment |
+------------------+
|             1220 |
+------------------+
1 row in set (0.01 sec)
```

Query 24: SELECT SUM(Capacity) AS Building_A_Capacity FROM room WHERE Building = 'Building A';

Output:

```
mysql> SELECT SUM(Capacity) AS Building_A_Capacity FROM room WHERE Building = 'Building A';
+--------------------+
| Building_A_Capacity |
+--------------------+
|                 90 |
+--------------------+
1 row in set (0.01 sec)
```

Query 25: SELECT AVG(Capacity) AS Average_Capacity FROM room;

Output:

```
mysql> SELECT AVG(Capacity) AS Average_Capacity FROM room;
+------------------+
| Average_Capacity |
+------------------+
|          43.3333 |
+------------------+
1 row in set (0.01 sec)
```

Query 26: SELECT AVG(Enrollment) AS Average_CS_Enrollment FROM section WHERE C_name IN (SELECT C_name FROM course WHERE Department = 'Computer Science');

Output:

```
mysql> SELECT AVG(Enrollment) AS Average_CS_Enrollment FROM section WHERE C_name IN (SELECT C_name FROM course WHERE Department = 'Computer Science');
+----------------------+
| Average_CS_Enrollment |
+----------------------+
|              406.6667 |
+----------------------+
1 row in set (0.01 sec)
```

Query 27: SELECT * FROM course WHERE C_name LIKE 'D%';

Output:

```
mysql> SELECT * FROM course WHERE C_name LIKE 'D%';
+----------+-----------------------------+------------------+
| C_number | C_name                      | Department       |
+----------+-----------------------------+------------------+
|      238 | Database Management System | Computer Science |
+----------+-----------------------------+------------------+
1 row in set (0.00 sec)
```

Query 28: SELECT * FROM room WHERE Building LIKE 'Building%';

Output:

```
mysql> SELECT * FROM room WHERE Building LIKE 'Building%';
+----------+----------+------------+
| R_number | Capacity | Building   |
+----------+----------+------------+
|      101 |       60 | Building A |
|      102 |       40 | Building B |
|      103 |       30 | Building A |
+----------+----------+------------+
3 rows in set (0.00 sec)
```

Query 29: SELECT * FROM room WHERE Capacity LIKE '%6%';

Output:

```
mysql> SELECT * FROM room WHERE Capacity LIKE '%6%';
+----------+----------+------------+
| R_number | Capacity | Building   |
+----------+----------+------------+
|      101 |       60 | Building A |
+----------+----------+------------+
1 row in set (0.00 sec)
```

Query 30: SELECT * FROM room WHERE Capacity IN (40, 50);

Output:

```
mysql> SELECT * FROM room WHERE Capacity IN (40, 50);
+----------+----------+------------+
| R_number | Capacity | Building   |
+----------+----------+------------+
|      102 |       40 | Building B |
+----------+----------+------------+
1 row in set (0.00 sec)
```

Query 31: SELECT * FROM course WHERE Department IN ('Computer Science');

Output:

```
mysql> SELECT * FROM course WHERE Department IN ('Computer Science');
+----------+---------------------------+------------------+
| C_number | C_name                    | Department       |
+----------+---------------------------+------------------+
|      305 | Core JAVA Programing      | Computer Science |
|      238 | Database Management System | Computer Science |
|      251 | Operating System          | Computer Science |
+----------+---------------------------+------------------+
3 rows in set (0.00 sec)
```

Query 32: SELECT * FROM room WHERE Capacity BETWEEN 40 AND 50;

Output:

```
mysql> SELECT * FROM room WHERE Capacity BETWEEN 40 AND 50;
+----------+----------+------------+
| R_number | Capacity | Building   |
+----------+----------+------------+
|      102 |       40 | Building B |
+----------+----------+------------+
1 row in set (0.01 sec)
```

Query 33: SELECT * FROM course WHERE C_number BETWEEN 200 AND 300;

Output:

```
mysql> SELECT * FROM course WHERE C_number BETWEEN 200 AND 300;
+----------+----------------------------+------------------+
| C_number | C_name                     | Department       |
+----------+----------------------------+------------------+
|      238 | Database Management System | Computer Science |
|      251 | Operating System           | Computer Science |
+----------+----------------------------+------------------+
2 rows in set (0.00 sec)
```

Query 34: SELECT R_number AS Room_Number, Capacity AS Room_Capacity FROM room;

Output:

```
mysql> SELECT R_number AS Room_Number, Capacity AS Room_Capacity FROM room;
+-------------+---------------+
| Room_Number | Room_Capacity |
+-------------+---------------+
|         101 |            60 |
|         102 |            40 |
|         103 |            30 |
+-------------+---------------+
3 rows in set (0.00 sec)
```

Query 35: SELECT S_number AS Section_Number, Enrollment AS Section_Enrollment FROM section;

Output:

```
mysql> SELECT S_number AS Section_Number, Enrollment AS Section_Enrollment FROM section;
+---------------+-------------------+
| Section_Number | Section_Enrollment |
+---------------+-------------------+
|             1 |               409 |
|             2 |               404 |
|             3 |               407 |
+---------------+-------------------+
3 rows in set (0.00 sec)
```

Query36: SELECT exam.*, room.*

FROM exam

INNER JOIN room ON exam.R_number = room.R_number;

Output:

```
mysql> SELECT exam.*, room.*
    -> FROM exam
    -> INNER JOIN room ON exam.R_number = room.R_number;
+----------+----------------------------+----------+----------+----------+----------+------------+
| time     | C_name                     | R_number | S_number | R_number | Capacity | Building   |
+----------+----------------------------+----------+----------+----------+----------+------------+
| 09:00:00 | Database Management System |      101 |        1 |      101 |       60 | Building A |
| 11:00:00 | Core JAVA Programing       |      102 |        2 |      102 |       40 | Building B |
| 08:00:00 | Operating System           |      103 |        3 |      103 |       30 | Building A |
+----------+----------------------------+----------+----------+----------+----------+------------+
3 rows in set (0.00 sec)
```

Query 37: SELECT section.*, course.*

FROM section

INNER JOIN course ON section.C_name = course.C_name;

Output:

```
mysql> SELECT section.*, course.*
    -> FROM section
    -> INNER JOIN course ON section.C_name = course.C_name;
+----------+------------+----------------------------+----------+----------------------------+------------------+
| S_number | Enrollment | C_name                     | C_number | C_name                     | Department       |
+----------+------------+----------------------------+----------+----------------------------+------------------+
|        1 |        409 | Database Management System |      238 | Database Management System | Computer Science |
|        2 |        404 | Database Management System |      238 | Database Management System | Computer Science |
|        3 |        407 | Database Management System |      238 | Database Management System | Computer Science |
+----------+------------+----------------------------+----------+----------------------------+------------------+
3 rows in set (0.00 sec)
```

Query 38: SELECT room.*, exam.*

FROM room

RIGHT JOIN exam ON room.R_number = exam.R_number;

Output:

```
mysql> SELECT room.*, exam.*
    -> FROM room
    -> RIGHT JOIN exam ON room.R_number = exam.R_number;
+----------+----------+------------+----------+----------------------------+----------+----------+
| R_number | Capacity | Building   | time     | C_name                     | R_number | S_number |
+----------+----------+------------+----------+----------------------------+----------+----------+
|      101 |       60 | Building A | 09:00:00 | Database Management System |      101 |        1 |
|      102 |       40 | Building B | 11:00:00 | Core JAVA Programing       |      102 |        2 |
|      103 |       30 | Building A | 08:00:00 | Operating System           |      103 |        3 |
+----------+----------+------------+----------+----------------------------+----------+----------+
3 rows in set (0.00 sec)
```

Query 39: SELECT course.*, section.*

FROM course

RIGHT JOIN section ON course.C_name = section.C_name;

Output:

```
mysql> SELECT course.*, section.*
    -> FROM course
    -> RIGHT JOIN section ON course.C_name = section.C_name;
+----------+---------------------------+-----------------+----------+------------+---------------------------+
| C_number | C_name                    | Department      | S_number | Enrollment | C_name                    |
+----------+---------------------------+-----------------+----------+------------+---------------------------+
|      238 | Database Management System | Computer Science |        1 |        409 | Database Management System |
|      238 | Database Management System | Computer Science |        2 |        404 | Database Management System |
|      238 | Database Management System | Computer Science |        3 |        407 | Database Management System |
+----------+---------------------------+-----------------+----------+------------+---------------------------+
3 rows in set (0.00 sec)
```

Query 40: SELECT exam.*, room.*

FROM exam

LEFT JOIN room ON exam.R_number = room.R_number;

Output:

```
mysql> SELECT exam.*, room.*
    -> FROM exam
    -> LEFT JOIN room ON exam.R_number = room.R_number;
+----------+---------------------------+----------+----------+----------+----------+------------+
| time     | C_name                    | R_number | S_number | R_number | Capacity | Building   |
+----------+---------------------------+----------+----------+----------+----------+------------+
| 09:00:00 | Database Management System |      101 |        1 |      101 |       60 | Building A |
| 11:00:00 | Core JAVA Programing       |      102 |        2 |      102 |       40 | Building B |
| 08:00:00 | Operating System           |      103 |        3 |      103 |       30 | Building A |
+----------+---------------------------+----------+----------+----------+----------+------------+
3 rows in set (0.00 sec)
```

Query 41: SELECT section.*, course.*

FROM section

LEFT JOIN course ON section.C_name = course.C_name;

Output:

```
mysql> SELECT section.*, course.*
    -> FROM section
    -> LEFT JOIN course ON section.C_name = course.C_name;
+----------+------------+---------------------------+----------+---------------------------+-----------------+
| S_number | Enrollment | C_name                    | C_number | C_name                    | Department      |
+----------+------------+---------------------------+----------+---------------------------+-----------------+
|        1 |        409 | Database Management System |      238 | Database Management System | Computer Science |
|        2 |        404 | Database Management System |      238 | Database Management System | Computer Science |
|        3 |        407 | Database Management System |      238 | Database Management System | Computer Science |
+----------+------------+---------------------------+----------+---------------------------+-----------------+
3 rows in set (0.00 sec)
```

Query 42: SELECT room.*, exam.*, section.*

FROM room

LEFT JOIN exam ON room.R_number = exam.R_number

LEFT JOIN section ON exam.S_number = section.S_number

UNION

SELECT room.*, exam.*, section.*

FROM room

RIGHT JOIN exam ON room.R_number = exam.R_number

RIGHT JOIN section ON exam.S_number = section.S_number;

Output:

```
mysql> SELECT room.*, exam.*, section.*
    -> FROM room
    -> LEFT JOIN exam ON room.R_number = exam.R_number
    -> LEFT JOIN section ON exam.S_number = section.S_number
    -> UNION
    -> SELECT room.*, exam.*, section.*
    -> FROM room
    -> RIGHT JOIN exam ON room.R_number = exam.R_number
    -> RIGHT JOIN section ON exam.S_number = section.S_number;
+----------+----------+------------+----------+--------------------------+----------+----------+----------+------------+--------------------------+
| R_number | Capacity | Building   | time     | C_name                   | R_number | S_number | S_number | Enrollment | C_name                   |
+----------+----------+------------+----------+--------------------------+----------+----------+----------+------------+--------------------------+
|      101 |       60 | Building A | 09:00:00 | Database Management System |      101 |        1 |        1 |        409 | Database Management System |
|      102 |       40 | Building B | 11:00:00 | Core JAVA Programing       |      102 |        2 |        2 |        404 | Database Management System |
|      103 |       30 | Building A | 08:00:00 | Operating System           |      103 |        3 |        3 |        407 | Database Management System |
+----------+----------+------------+----------+--------------------------+----------+----------+----------+------------+--------------------------+
3 rows in set (0.01 sec)
```

Query 43: ALTER TABLE Course

MODIFY COLUMN C_name VARCHAR(30) NOT NULL;

Output:

```
mysql> ALTER TABLE Course
    -> MODIFY COLUMN C_name VARCHAR(30) NOT NULL;
Query OK, 0 rows affected (0.04 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

Query 44: CREATE TABLE Employee (Employee_ID INT NOT NULL,Name VARCHAR(50),Department VARCHAR(50));

Output:

```
mysql> CREATE TABLE Employee (
    ->     Employee_ID INT NOT NULL,
    ->     Name VARCHAR(50),
    ->     Department VARCHAR(50)
    -> );
Query OK, 0 rows affected (0.05 sec)
```

Query 45: SELECT * FROM Section WHERE Enrollment IS NOT NULL;

Output:

```
mysql> SELECT * FROM Section
    -> WHERE Enrollment IS NOT NULL;
+----------+------------+----------------------------+
| S_number | Enrollment | C_name                     |
+----------+------------+----------------------------+
|        1 |        409 | Database Management System |
|        2 |        404 | Database Management System |
|        3 |        407 | Database Management System |
+----------+------------+----------------------------+
3 rows in set (0.00 sec)
```

Query 46: ALTER TABLE Room ADD CONSTRAINT check_capacity CHECK (Capacity > 0);

Output:

```
mysql> ALTER TABLE Room
    -> ADD CONSTRAINT check_capacity CHECK (Capacity > 0);
Query OK, 3 rows affected (0.06 sec)
Records: 3  Duplicates: 0  Warnings: 0
```

Query 47: UPDATE Section SET Enrollment = ABS(Enrollment)

WHERE Enrollment < 0;

Output:

```
mysql> UPDATE Section
    -> SET Enrollment = ABS(Enrollment)
    -> WHERE Enrollment < 0;
Query OK, 0 rows affected (0.00 sec)
Rows matched: 0  Changed: 0  Warnings: 0
```

Query 48: ALTER TABLE Section ALTER COLUMN Enrollment SET DEFAULT 0;

Output:

```
mysql> ALTER TABLE Section
    -> ALTER COLUMN Enrollment SET DEFAULT 0;
Query OK, 0 rows affected (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

Query 49: SELECT c1.C_name, c2.C_name

FROM course c1

INNER JOIN course c2 ON c1.Department = c2.Department AND c1.C_name <> c2.C_name;

Output:

```
mysql> SELECT c1.C_name, c2.C_name
    -> FROM course c1
    -> INNER JOIN course c2 ON c1.Department = c2.Department AND c1.C_name <> c2.C_name;
+--------------------------+--------------------------+
| C_name                   | C_name                   |
+--------------------------+--------------------------+
| Operating System         | Core JAVA Programing     |
| Database Management System | Core JAVA Programing   |
| Operating System         | Database Management System |
| Core JAVA Programing     | Database Management System |
| Database Management System | Operating System       |
| Core JAVA Programing     | Operating System         |
+--------------------------+--------------------------+
6 rows in set (0.00 sec)
```

Query 50: SELECT s1.S_number, s2.S_number

FROM section s1

INNER JOIN section s2 ON s1.C_name = s2.C_name AND s1.S_number <> s2.S_number;

Output:

```
mysql> SELECT s1.S_number, s2.S_number
    -> FROM section s1
    -> INNER JOIN section s2 ON s1.C_name = s2.C_name AND s1.S_number <> s2.S_number;
+----------+----------+
| S_number | S_number |
+----------+----------+
|        1 |        2 |
|        1 |        3 |
|        2 |        1 |
|        2 |        3 |
|        3 |        1 |
|        3 |        2 |
+----------+----------+
6 rows in set (0.00 sec)
```

Query51: SELECT C_name, COUNT(*) AS Section_Count

FROM Section

GROUP BY C_name;

Output:

```
mysql> SELECT C_name, COUNT(*) AS Section_Count
    -> FROM Section
    -> GROUP BY C_name;
+----------------------------+---------------+
| C_name                     | Section_Count |
+----------------------------+---------------+
| Database Management System |             3 |
+----------------------------+---------------+
1 row in set (0.01 sec)
```

❖