Project Work Of Database Management System

Tribhuvan University

Amrit Science Campus

Thamel, Kathmandu

Submitted By: Arjun Mijar

Faculty: CSIT

Roll No.: 18

Section: A

Submitted To: Er. Yograj Joshi

Internal Examiner:

Signature: _____

External Examiner

Signature: _____

Database and Database Management System

- Database: A **database** is an organized collection of structured data stored electronically. It allows for efficient data storage, retrieval, and management, typically in tables. For examples: MySQL, Oracle, MongoDB

- Database Management System (DBMS): A **DBMS** is software that manages databases, providing tools for defining, querying, updating, and securing data. It acts as an interface between the database and users or applications, ensuring data integrity, security, and efficient access. For examples: SQL Server, PostgreSQL, MongoDB

Constraints

Constraints in a database are rules applied to columns in a table to enforce data integrity, accuracy, and reliability. They ensure that the data entered into the database adheres to certain conditions or rules.

Types of Constraints:

1. **Primary Key Constraint**:
   - Ensures each row in the table is unique and not null.
   - Example: PRIMARY KEY (ID)
2. **Foreign Key Constraint**:
   - Enforces a link between two tables by ensuring the value in a column matches a value in the referenced table.
   - Example: FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)
3. **Unique Constraint**:
   - Ensures that all values in a column or a set of columns are unique across the table.
   - Example: UNIQUE (Email)
4. **Not Null Constraint**:
   - Ensures that a column cannot contain NULL values.
   - Example: Name VARCHAR(100) NOT NULL
5. **Check Constraint**:
   - Ensures that all values in a column satisfy a specific condition.
   - Example: CHECK (Age >= 18)
6. **Default Constraint**:
   - Assign a column default value if no value is specified during data insertion.
   - Example: Salary DECIMAL(10, 2) DEFAULT 50000

Primary key

A **primary key** is a column or a combination of columns in a database table that uniquely identifies each row in that table. It ensures that no two rows have the same primary key value, maintaining the integrity and uniqueness of the data.

Key Characteristics:

- **Uniqueness**: Each value in the primary key column(s) must be unique.
- **Not Null**: A primary key cannot contain NULL values.
- **Single or Composite**: It can be a single column (e.g., ID) or a combination of columns (composite key, e.g., FirstName + LastName).

Example:

In a Student table:

```
CREATE TABLE Student (
   UserID INT PRIMARY KEY,
   Username VARCHAR(50),
   Email VARCHAR(100)
);
```

Foreign Key

A **foreign key** is a column or a set of columns in a database table that creates a relationship between two tables. It links the data in one table (the child table) to a primary key or unique key in another table (the parent table), enforcing referential integrity.

Key Characteristics:

- **Relationship**: The foreign key establishes a link between the child table and the parent table.
- **Referential Integrity**: Ensures that the values in the foreign key column must match values in the primary key or unique key column of the parent table, or be `NULL`.
- **Cascading Actions**: When a referenced row in the parent table is updated or deleted, the changes can cascade to the child table, depending on the defined rules.

View: In SQL, a view is a virtual table based on the result set of an SQL statement. A view contains rows and columns, just like a real table. The fields in a view are fields from one or more real tables in the database. You can add SQL statements and functions to a view and present the data as if the data were coming from one single table. A view is created with the CREATE VIEW statement.

CREATE VIEW Syntax

- CREATE VIEW view_name AS SELECT column1, column2, …. FROM table_name WHERE condition;
- eg. CREATE VIEW IT_Employee AS SELECT id, name, department FROM Employee WHERE department = "IT";

Few Commands of SQL

1. To Create a database:
   - CREATE DATABASE databasename;
   - Example: CREATE DATABASE college;

2. To Delete a database:
   - DROP DATABASE databasename;
   - Example: DROP DATABASE college;

3. To Create a Table:
   - CREATE TABLE table_name(
     column1 datatype,
     column2 datatype,
     colulmn3 datatype,
     ……..
     );
   - Example: CREATE TABLE student(
     Roll_no int primary key,
     Name varchar(50),
     Class int,
     College_name varchar(40)
     );

4. To Delete a Table:
   - DROP TABLE table_name;
   - Example: DROP TABLE student;

5. To Alter Table:
   - The ALTER TABLE statement is used to add, delete or modify columns in an existing table. It is also used to add and drop various constraints on an existing table.

   For adding the primary key

   ALTER TABEL table_name ADD PRIMARY KEY column_name;

   For adding foreign key

   ALTER TABEL table_name ADD FOREIGN KEY(column_name) REFERENCES table_name(column_name);

   For changing datatype

   ALTER TABEL table_name ALTER COLUMN column_name TYPE new_data_type;

6. To display the schema of table
   - desc table_name;

7. To display the data of the table
   - select * from table_name;

8. SELECT DISTINCT: The SELECT DISTINCT statement is used to return only distinct (different) values.
   - Syntax: SELECT DISTINCT column1, column2, ... FROM table_name;
   - eg. SELECT DISTINCT country from employee;

9. WHERE: The WHERE clause is used to filter records. It is used to extract only those records that fulfill a specified condition.
   - Syntax: SELECT column1, column2, ……. from table_name WHERE condition;
   - eg. SELECT * FROM customers WHERE country="Nepal";

10. ORDER BY : The ORDER BY keyword is used to sort the result set in ascending or descending order.
    - Syntax: SELECT column1, column2, …. FROM table_name ORDER BY column1, column2, ….. ASC|DESC;
    - eg. SELECT * FROM products ORDER BY price;

11. AND Operator: The AND operator is used to filter records based on more than one condition.
    - Syntax: SELECT column1, column2, …. FROM table_name WHERE condition1 AND condtion2 AND condtion3 …..;
    - eg. SELECT * FROM customers WHERE country = "Nepal" AND city = "Kathmandu" AND postalcode > 12000;

12. OR Operator: The OR operator is used to filter records based on more than one condtion.
    - Syntax: SELECT column1, column2, … FROM table_name WHRE condition1 OR condition2 OR condition3 ….;
    - eg. SELECT FROM * customers WHRE country="Nepal" OR country ="India";

13. NOT Operator: The NOT operator is used in combination with other operators to give the opposite result, also called the negative result.
    - Syntax: SELECT column1, column2, … FROM table_name where NOT condtion;
    - eg. SELECT * FROM customers where NOT country = "India";

14. INSERT INTO: It is used to insert new records in a table.
    - Syntax 1: INSERT INTO table_name (column1, column2, column3, ….) values(value1, value2, vlaue3,….);
    - eg. INSERT INTO employee(name, age,sex) values("Arjun", 22, "M");
    - Syntax 2: INSERT INTO table_name values(value1, value2, vlaue3,….);
    - eg. INSERT INTO employee values("Arjun", 22, "M");

15. NULL Values: A field with a NULL value is a field with no value. If a field in a table is optional, it is possible to insert a new record or update a record without adding a value to this field. Then the field will be saved with a NULL value.
    - IS NULL Syntax: SELECT column_name FROM table_name WHRE column_name IS NULL;
    - IS NOT NULL Syntax: SELECT column_name FROM table_name WHRE column_name IS NOT NULL;
16. UPDATE: The UPDATE statement is used to modify the existing records in a table.
    - Syntax: UPDATE table_name SET column1=value1, column2=value2, …. WHERE condition;
    - eg. UPDATE employee SET address="Kathmandu" where id=2;

17. DELETE: The DELETE statement is used to delete existing records in a table.
    - Syntax: DELETE FROM table_name WHERE condition;
    - eg. DELETE FROM employee WHERE id=2;

18. MIN() and MAX() Functions: The MIN() function returns the smallest value of the selected column. And the MAX() function returns the largest value of the selected column.
    - Syntax for MIN(): SELECT MIN(column_name) FROM table_name WHERE condition;
    - Syntax for MAX(): SELECT MAX(column_name) FROM table_name WHERE condition;

19. COUNT() Function: The COUNT() function returns the number of rows that matches a specified criterion.
    - SELECT COUNT(column_name) FROM table_name WHERE condtion;
    - eg. SELECT COUNT(*) FROM Products;

20. SUM() Function: The SUM() function returns the total sum of a numeric column.
    - Syntax: SELECT SUM(column_name) FROM table_name WHERE condition;
    - eg. SELECT SUM(quantity) FROM orderDetails;

21. AVG () Function: The AVG() function returns the average value of a numeric column.
    - SELECT AVG(column_name) FROM table_name;
    - eg. SELECT AVG(price) FROM products;

22. IN Operator: The IN operator allows you to specify multiple values in a WHERE clause. The IN operator is a shorthand for multiple OR conditions.
    - Syntax: SELECT column_name(S) FROM table_name WHERE column_name IN (value1, value2, …..);
    - eg. SELECT * FROM customers WHERE country IN ("Nepal", "China", "India");

23. BETWEEN Operator: The BETWEEN operator selects values within a given range. The values can be numbers, text, or dates. The BETWEEN operator is inclusive: begin and end values are included.
    - SELECT column_name(s) FROM table_name WHRE column_name BETWEEN value1 AND value2;
    - eg. SELECT * FROM products WHERE price BETWEEN 10 AND 20;

24. JOIN: The JOIN clause is used to combine rows from two or more tables, based on a related column between them.
    - Syntax: SELECT column_name(s) FROM table1 JOIN table2 ON table1.column_name = table2.column_name;

25. LEFT JOIN: The LEFT JOIN keyword returns all records from the left table (table1), and the matching records from the right table (table2). The result is 0 records from the right side, if there is no match.
    - Syntax: SELECT column_name(s) FROM table1 LEFT JOIN table2 ON tabel1.column_name = table2.column_name;
    - eg. SELECT customers.customerName, orders.OrderID FROM customers LEFT JOIN orders ON customers.CustomerID = order.customerID ORDER BY customers. customerName;

26. RIGHT JOIN: The RIGHT JOIN keyword returns all records from the right table (table2), and the matching records from the left table (table1). The result is 0 records from the left side, if there is no match.
    - Syntax: SELECT column_name(s) FROM table1 RIGHT JOIN table2 ON tabel1.column_name = table2.column_name;

27. GROUP BY: The GROUP BY statement groups rows that have the same values into summary rows, like "find the number of customers in each country". The GROUP BY statement is often used with aggregate functions (COUNT() , MAX() , MIN() , SUM(), AVG()) to group the result-set by one or more columns.
    - Syntax: SELECT column_name(s) FROM table_name WHERE condition GROUP BY column_name(s) ORDER BY column_name(s);
    - eg. SELECT COUNT(customerID), country FROM customers GROUP BY country;

28. HAVING: The HAVING clauses was added to SQL because the WHERE keyword cannot be used with aggregate functions.
    - Syntax: SELECT column_name(s) FROM table_name WHERE condition GROUP BY column_name(s) HAVING condition ORDER BY column_name(s);
    - eg. SELECT COUNT(customerID), country FROM customers GROUP BY country HAVING COUNT(customerID) > 5;

1. Consider the following relational database, where the primary keys are underlined.
   Employee (person-name, street, city)
   Works (person-name, company-name, salary)
   Company (company-name, city)
   Manages (person-name, manager-name)

a)Create the above tables in MySQL

Commands:

```
MariaDB [database_one]> create table employee(person_name varchar(40) primary key, street varch
ar(50), city varchar(50));
Query OK, 0 rows affected (0.011 sec)

MariaDB [database_one]> create table company(company_name varchar(30) primary key, city varchar
(40));
Query OK, 0 rows affected (0.009 sec)

MariaDB [database_one]> create table works(person_name varchar(30), company_name varchar(40), s
alary decimal(10,2), primary key(person_name), foreign key(person_name) references employee(per
son_name), foreign key(company_name) references company(company_name));
Query OK, 0 rows affected (0.020 sec)

MariaDB [database_one]> create table manages(person_name varchar(40), manager_name varchar(50),
 primary key(person_name), foreign key (person_name) references employee(person_name), foreign
key(manager_name) references employee(person_name));
Query OK, 0 rows affected (0.010 sec)
```

```
MariaDB [database_one]> desc employee;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| person_name | varchar(40) | NO   | PRI | NULL    |       |
| street      | varchar(50) | YES  |     | NULL    |       |
| city        | varchar(50) | YES  |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
3 rows in set (0.020 sec)

MariaDB [database_one]> desc company;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| company_name | varchar(30) | NO   | PRI | NULL    |       |
| city         | varchar(40) | YES  |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
2 rows in set (0.032 sec)
```

```
MariaDB [database_one]> desc works;
+--------------+---------------+------+-----+---------+-------+
| Field        | Type          | Null | Key | Default | Extra |
+--------------+---------------+------+-----+---------+-------+
| person_name  | varchar(30)   | NO   | PRI | NULL    |       |
| company_name | varchar(40)   | YES  | MUL | NULL    |       |
| salary       | decimal(10,2) | YES  |     | NULL    |       |
+--------------+---------------+------+-----+---------+-------+
3 rows in set (0.023 sec)

MariaDB [database_one]> desc manages;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| person_name  | varchar(40) | NO   | PRI | NULL    |       |
| manager_name | varchar(50) | YES  | MUL | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
2 rows in set (0.021 sec)
```

b)Insert any three records in each above table and display them.

```
MariaDB [database_one]> insert into employee values("Arjun","Devinagar 10","Kathmandu"),("Sitar
am","Nayaneshwor 21","Lalitpur"),("Roman","Madhevstan 4","Bhaktapur");
Query OK, 3 rows affected (0.003 sec)
Records: 3  Duplicates: 0  Warnings: 0

MariaDB [database_one]> insert into company values("First Bank Corporation",
"Kathmandu"),("Logix Computer","Bhaktapur"),("Tech Innovators","Lalitpur");
Query OK, 3 rows affected (0.004 sec)
Records: 3  Duplicates: 0  Warnings: 0

MariaDB [database_one]> insert into works values("Arjun","First Bank Corporation",40000),("Sita
ram","Logix Computer", 35000),("Roman","Tech Innovators",30000);
Query OK, 3 rows affected (0.004 sec)
Records: 3  Duplicates: 0  Warnings: 0

MariaDB [database_one]> insert into manages values("Arjun","Sitaram"),("Roman",
"Arjun"),("Sitaram","Roman");
Query OK, 3 rows affected (0.003 sec)
Records: 3  Duplicates: 0  Warnings: 0
```

```
MariaDB [database_one]> select * from employee;
+-------------+----------------+-----------+
| person_name | street         | city      |
+-------------+----------------+-----------+
| Arjun       | Devinagar 10   | Kathmandu |
| Roman       | Madhevstan 4   | Bhaktapur |
| Sitaram     | Nayaneshwor 21 | Lalitpur  |
+-------------+----------------+-----------+
3 rows in set (0.001 sec)

MariaDB [database_one]> select * from company;
+-----------------------+-----------+
| company_name          | city      |
+-----------------------+-----------+
| First Bank Corporation | Kathmandu |
| Logix Computer        | Bhaktapur |
| Tech Innovators       | Lalitpur  |
+-----------------------+-----------+
3 rows in set (0.001 sec)

MariaDB [database_one]> select * from works;
+-------------+-----------------------+----------+
| person_name | company_name          | salary   |
+-------------+-----------------------+----------+
| Arjun       | First Bank Corporation | 40000.00 |
| Roman       | Tech Innovators       | 30000.00 |
| Sitaram     | Logix Computer        | 35000.00 |
+-------------+-----------------------+----------+
3 rows in set (0.001 sec)

MariaDB [database_one]> select * from manages;
+-------------+--------------+
| person_name | manager_name |
+-------------+--------------+
| Roman       | Arjun        |
| Sitaram     | Roman        |
| Arjun       | Sitaram      |
+-------------+--------------+
3 rows in set (0.000 sec)
```

c)Write the SQL for each of the following queries.

a). Find the names of all employees who work for First Bank Corporation.

```
MariaDB [database_one]> select person_name from works where company_name="First
 Bank Corporation";
+-------------+
| person_name |
+-------------+
| Arjun       |
+-------------+
1 row in set (0.001 sec)
```

b). Find the names and cities of residence of all employees who work for First Bank Corporation.

```
MariaDB [database_one]> select E.person_name, E.city from employee E, works W
where E.person_name=W.person_name and W.company_name="First Bank Corporation";
+-------------+-----------+
| person_name | city      |
+-------------+-----------+
| Arjun       | Kathmandu |
+-------------+-----------+
1 row in set (0.001 sec)
```

c). Find the names, street addresses, and cities of residence of all employees who work for First Bank Corporation and earn more than Rs 10,000 per annum.

```
MariaDB [database_one]> select E.person_name, E.street, E.city from employee E
, works W where E.person_name=W.person_name and W.company_name="First Bank Cor
poration" and W.salary > 10000;
+-------------+-------------+-----------+
| person_name | street      | city      |
+-------------+-------------+-----------+
| Arjun       | Devinagar 10 | Kathmandu |
+-------------+-------------+-----------+
1 row in set (0.001 sec)
```

d) Find the names of all employees in this database who do not work for First Bank Corporation.

```
MariaDB [database_one]> select E.person_name from employee E where E.person_name
not in(select person_name from works where company_name ="First Bank Corporation"
);
+-------------+
| person_name |
+-------------+
| Roman       |
| Sitaram     |
+-------------+
2 rows in set (0.001 sec)
```

e) Find all employees in the database who live in the same cities as the companies for which they work.

```
MariaDB [database_one]> select E.person_name from employee E, works W, company C
where E.person_name=W.person_name and W.company_name=C.company_name and E.city=C.
city;
+-------------+
| person_name |
+-------------+
| Arjun       |
+-------------+
1 row in set (0.001 sec)
```

f) Find all companies in which the average salary of an employee is more than 5000.

```
MariaDB [database_one]> select company_name from works group by company_name
 having AVG(salary)>5000;
+------------------------+
| company_name           |
+------------------------+
| First Bank Corporation |
| Logix Computer         |
| Tech Innovators        |
+------------------------+
3 rows in set (0.001 sec)
```

g) Update the salary of all the employees who work for First Bank Corporation by 10%.

```
MariaDB [database_one]> update works set salary=salary*1.10 where company_na
me="First Bank Corporation";
Query OK, 1 row affected (0.003 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [database_one]> select * from works;
+-------------+------------------------+----------+
| person_name | company_name           | salary   |
+-------------+------------------------+----------+
| Arjun       | First Bank Corporation | 44000.00 |
| Roman       | Tech Innovators        | 30000.00 |
| Sitaram     | Logix Computer         | 35000.00 |
+-------------+------------------------+----------+
3 rows in set (0.000 sec)
```

h) Delete the records of all employees who work for First Bank Corporation.

```
MariaDB [database_one]> delete from works where company_name="First Bank Cor
poration";
Query OK, 1 row affected (0.003 sec)

MariaDB [database_one]> select * from works;
+-------------+-----------------+----------+
| person_name | company_name    | salary   |
+-------------+-----------------+----------+
| Roman       | Tech Innovators | 30000.00 |
| Sitaram     | Logix Computer  | 35000.00 |
+-------------+-----------------+----------+
2 rows in set (0.001 sec)
```

i) Create a view to find the names, street addresses, and cities of residence of all employees who work for First Bank Corporation and earn more than Rs 10,000 per annum.

```
MariaDB [database_one]> create view HighEarners as select E.person_name, E.street,
 E.city from employee E, works W where E.person_name=W.person_name and
 W.company_name="First Bank Corporation" and W.salary > 10000;
Query OK, 0 rows affected (0.003 sec)

MariaDB [database_one]> select * from HighEarners;
Empty set (0.001 sec)
```

After inserting values:

```
MariaDB [database_one]> insert into employee values("Arjun Mijar","Madan Bha
ndari Margh 08", "Kathmandu");
Query OK, 1 row affected (0.003 sec)

MariaDB [database_one]> insert into works values("Arjun Mijar", "First Bank
Corporation", 21500);
Query OK, 1 row affected (0.004 sec)

MariaDB [database_one]> select * from HighEarners;
+-------------+-------------------------+-----------+
| person_name | street                  | city      |
+-------------+-------------------------+-----------+
| Arjun Mijar | Madan Bhandari Margh 08 | Kathmandu |
+-------------+-------------------------+-----------+
1 row in set (0.001 sec)
```

2. For the following relations
   Members (mid, name, design, age)
   Books (Bid, Btitle, BAuthor, Bpublisher, Bprice)
   Reserves (mid, Bid, date)
   Where Bid is book identification, Btitle is Book title, Bpublisher is book publisher, Bprice is Book price, mid is Members identification, and Desig is designation.

   a)Create the above tables in MySQL

Commands:

- create table members(mid int primary key, name varchar(30), design varchar(40), age int);

```
MariaDB [database_two]> desc members;
+--------+-------------+------+-----+---------+-------+
| Field  | Type        | Null | Key | Default | Extra |
+--------+-------------+------+-----+---------+-------+
| mid    | int(11)     | NO   | PRI | NULL    |       |
| name   | varchar(30) | YES  |     | NULL    |       |
| design | varchar(40) | YES  |     | NULL    |       |
| age    | int(11)     | YES  |     | NULL    |       |
+--------+-------------+------+-----+---------+-------+
4 rows in set (0.015 sec)
```

- create table books(Bid int primary key, Btitle varchar(40), BAuther varchar(50), Bpublisher varchar(50), Bprice decimal(10,2));

```
MariaDB [database_two]> desc books;
+------------+---------------+------+-----+---------+-------+
| Field      | Type          | Null | Key | Default | Extra |
+------------+---------------+------+-----+---------+-------+
| Bid        | int(11)       | NO   | PRI | NULL    |       |
| Btitle     | varchar(40)   | YES  |     | NULL    |       |
| BAuther    | varchar(50)   | YES  |     | NULL    |       |
| Bpublisher | varchar(50)   | YES  |     | NULL    |       |
| Bprice     | decimal(10,2) | YES  |     | NULL    |       |
+------------+---------------+------+-----+---------+-------+
5 rows in set (0.015 sec)
```

- create table Reserves(mid int, Bid int, date DATE, primary key(mid,Bid), foreign key(mid) references members(mid), foreign key(Bid) references books(Bid));

```
MariaDB [database_two]> desc reserves;
+-------+---------+------+-----+---------+-------+
| Field | Type    | Null | Key | Default | Extra |
+-------+---------+------+-----+---------+-------+
| mid   | int(11) | NO   | PRI | NULL    |       |
| Bid   | int(11) | NO   | PRI | NULL    |       |
| date  | date    | YES  |     | NULL    |       |
+-------+---------+------+-----+---------+-------+
3 rows in set (0.017 sec)
```

b)Insert any three records in each of the above tables and display them.

Commands:

- insert into members values(1, "Arjun Rokka", "Professor", 40),(2, "Arun Chaudary", "Student", 25),(3, "Aayush Gautam", "Professor",35);

```
MariaDB [database_two]> select * from members;
+-----+--------------+-----------+------+
| mid | name         | design    | age  |
+-----+--------------+-----------+------+
|   1 | Arjun Rokka  | Professor |   46 |
|   2 | Arun Chaudary | Student  |   25 |
|   3 | Aayush Gautam | Professor |   35 |
+-----+--------------+-----------+------+
3 rows in set (0.000 sec)
```

- insert into books values(101, "Database System", "Raghu Lal", "Read More", 2500),(102, "Introduction to Computer", "Gyani Ray", "KEC", 500),(103, "Think and Grow Rich", "Oliver Napoleon", "Good Readers",1200);

```
MariaDB [database_two]> select * from books;
+-----+-------------------------+----------------+-----------------+---------+
| Bid | Btitle                  | BAuther        | Bpublisher      | Bprice  |
+-----+-------------------------+----------------+-----------------+---------+
| 101 | Database System         | Raghu Lal      | Read More       | 2500.00 |
| 102 | Introduction to Computer | Gyani Ray     | KEC             |  500.00 |
| 103 | Think and Grow Rich     | Oliver Napoleon | Asia Publication | 1200.00 |
+-----+-------------------------+----------------+-----------------+---------+
3 rows in set (0.000 sec)
```

- insert into reserves values(1, 101, '2023-09-12'),(2, 102, '2024-05-31'),(3, 103, '2020-02-25');

```
MariaDB [database_two]> select * from reserves;
+-----+-----+------------+
| mid | Bid | date       |
+-----+-----+------------+
|   1 | 101 | 2023-09-12 |
|   2 | 102 | 2024-05-31 |
|   3 | 103 | 2020-02-25 |
+-----+-----+------------+
3 rows in set (0.001 sec)
```

c)Write the SQL for each of the following queries.

a) List the titles of books reserved by professors older than 45 years

Commands:

- select B.Btitle from books B join reserves R on B.Bid=R.Bid join members M on M.mid=R.mid where M.design= "Professor" and M.age>45;

```
MariaDB [database_two]> select B.Btitle from books B join reserves R on B.Bid=R.Bid
 join members M on M.mid=R.mid where M.design= "Professor" and M.age>45;
+-----------------+
| Btitle          |
+-----------------+
| Database System |
+-----------------+
1 row in set (0.001 sec)
```

b) Find IDs of members who have not reserved books costing more than Rs. 500.

- select M.mid from members M where M.mid not in (select R.mid from reserves R join books B on R.Bid=B.Bid where B.Bprice>500);

```
MariaDB [database_two]> select M.mid from members M where M.mid not in (select R.mi
d from reserves R join books B on R.Bid=B.Bid where B.Bprice>500);
+-----+
| mid |
+-----+
|   2 |
+-----+
1 row in set (0.001 sec)
```

c) Find the author and title of books reserved on 27-May-2007.

- select B.BAuther, B.Btitle from books B join reserves R on B.Bid = R.Bid where R.date = '2007-05-27';

```
MariaDB [database_two]> select M.mid from members M where M.mid not in (select
R.mid from reserves R join books B on R.Bid=B.Bid where B.Bprice>500);
+-----+
| mid |
+-----+
|   2 |
+-----+
1 row in set (0.001 sec)
```

d) Find the names of members who have reserved all books.

- select M.name from members M where not exists(select B.Bid from books B where not exists(select R.Bid from reserves R where R.mid=M.mid and R.Bid=B.Bid));

```
MariaDB [database_two]> select M.name from members M where not exists(select B.
Bid from books B where not exists(select R.Bid from reserves R where R.mid=M.mi
d and R.Bid=B.Bid));
Empty set (0.001 sec)
```

e) Update the price of all the books by Rs 100 whose publisher name is ‘Asia Publication’

- update books set Bprice=Bprice+100 where Bpublisher = 'Asia Publication';

```
MariaDB [database_two]> select * from books;
+-----+-------------------------+----------------+------------------+---------+
| Bid | Btitle                  | BAuther        | Bpublisher       | Bprice  |
+-----+-------------------------+----------------+------------------+---------+
| 101 | Database System         | Raghu Lal      | Read More        | 2500.00 |
| 102 | Introduction to Computer | Gyani Ray     | KEC              |  500.00 |
| 103 | Think and Grow Rich     | Oliver Napoleon | Asia Publication | 1300.00 |
+-----+-------------------------+----------------+------------------+---------+
3 rows in set (0.000 sec)
```

f) Delete the records of all members whose age is less than 18.

- delete from members where age<18;

```
MariaDB [database_two]> select * from members;
+-----+---------------+-----------+------+
| mid | name          | design    | age  |
+-----+---------------+-----------+------+
|   1 | Arjun Rokka   | Professor |   46 |
|   2 | Arun Chaudary | Student   |   25 |
|   3 | Aayush Gautam | Professor |   35 |
+-----+---------------+-----------+------+
3 rows in set (0.000 sec)
```

3) Consider the following relational schema and write the relational algebra expression and SQL for the following.

Supplier
(supplier-id, supplier-name, city)
Supplies
(supplier-id, part-id, quantity)
Parts
(part-id, part-name, color, weight)

Creating Tables

Commands:

- create table supplies(supplier_id int, part_id int, quantity int, primary key(supplier_id,part_id));

```
MariaDB [database_three]> desc supplier;
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| supplier_id   | int(11)     | NO   | PRI | NULL    |       |
| supplier_name | varchar(50) | YES  |     | NULL    |       |
| city          | varchar(50) | YES  |     | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
3 rows in set (0.012 sec)
```

- create table parts(part_id int primary key, part_name varchar(50), color varchar(30), weight decimal(10,2));

```
MariaDB [database_three]> desc parts;
+-----------+---------------+------+-----+---------+-------+
| Field     | Type          | Null | Key | Default | Extra |
+-----------+---------------+------+-----+---------+-------+
| part_id   | int(11)       | NO   | PRI | NULL    |       |
| part_name | varchar(50)   | YES  |     | NULL    |       |
| color     | varchar(30)   | YES  |     | NULL    |       |
| weight    | decimal(10,2) | YES  |     | NULL    |       |
+-----------+---------------+------+-----+---------+-------+
4 rows in set (0.012 sec)
```

- create table supplies(supplier_id int, part_id int, quantity int, primary key(supplier_id,part_id), foreign key(supplier_id) references supplier(supplier_id), foreign key(part_id) references parts(part_id));

```
MariaDB [database_three]> desc supplies;
+-------------+---------+------+-----+---------+-------+
| Field       | Type    | Null | Key | Default | Extra |
+-------------+---------+------+-----+---------+-------+
| supplier_id | int(11) | NO   | PRI | NULL    |       |
| part_id     | int(11) | NO   | PRI | NULL    |       |
| quantity    | int(11) | YES  |     | NULL    |       |
+-------------+---------+------+-----+---------+-------+
3 rows in set (0.011 sec)
```

Value Insertion

Commands:

- insert into supplier values(1, "ABC Company", "Kathmandu"), (2, "Banapali Traders", "Kathmandu"), (3, "Satko Company", "Lalitpur");

```
MariaDB [database_three]> select * from supplier;
+-------------+------------------+-----------+
| supplier_id | supplier_name    | city      |
+-------------+------------------+-----------+
|           1 | ABC Company      | Kathmandu |
|           2 | Banapali Traders | Kathmandu |
|           3 | Satko Company    | Lalitpur  |
+-------------+------------------+-----------+
3 rows in set (0.001 sec)
```

- insert into parts values(101, "Head Light", "Black", 300.45),(102, "Iron", "Light Blue", 110.65),(103, "Coffee Machine", "Silver White", 15400.34);

```
MariaDB [database_three]> select * from parts;
+---------+----------------+--------------+----------+
| part_id | part_name      | color        | weight   |
+---------+----------------+--------------+----------+
|     101 | Head Light     | Black        |   300.45 |
|     102 | Iron           | Light Blue   |   110.65 |
|     103 | Coffee Machine | Silver White | 15400.34 |
+---------+----------------+--------------+----------+
3 rows in set (0.000 sec)
```

- insert into supplies values(1,101,301), (2,102,50), (3,103,110);

```
MariaDB [database_three]> select * from supplies;
+-------------+---------+----------+
| supplier_id | part_id | quantity |
+-------------+---------+----------+
|           1 |     101 |      301 |
|           2 |     102 |       50 |
|           3 |     103 |      110 |
+-------------+---------+----------+
3 rows in set (0.000 sec)
```

a)Find the name of all suppliers located in the city "Kathmandu".

Commands:

- select supplier_name from supplier where city="Kathmandu";

```
MariaDB [database_three]> select supplier_name from supplier where city="Kathmandu";
+------------------+
| supplier_name    |
+------------------+
| ABC Company      |
| Banapali Traders |
+------------------+
2 rows in set (0.001 sec)
```

b)Find the name of all parts supplied by "ABC Company".

- select P.part_name from parts P join supplies S on P.part_id=S.part_id join supplier SP on S.supplier_id=SP.supplier_id where SP.supplier_name="ABC Company";

```
MariaDB [database_three]> select P.part_name from parts P join supplies S on
 P.part_id=S.part_id join supplier SP on S.supplier_id=SP.supplier_id where
SP.supplier_name="ABC Company";
+------------+
| part_name  |
+------------+
| Head Light |
+------------+
1 row in set (0.001 sec)
```

c)Find the name of all parts that are supplied in quantity greater than 300.

- select P.part_name from parts P join supplies S on P.part_id = S.part_id where S.quantity>300;

```
MariaDB [database_three]> select P.part_name from parts P join supplies S on
 P.part_id = S.part_id where S.quantity>300;
+------------+
| part_name  |
+------------+
| Head Light |
+------------+
1 row in set (0.001 sec)
```

d)Find the number of parts supplied by "ABC Company'.

- select count(distinct S.part_id) AS number_of_parts from supplies S join supplier SP ON S.supplier_id = SP.supplier_id where SP.supplier_name = "ABC Company";

```
MariaDB [database_three]> select count(distinct S.part_id) AS number_of_part
s from supplies S join supplier SP ON S.supplier_id = SP.supplier_id where S
P.supplier_name = "ABC Company";
+-----------------+
| number_of_parts |
+-----------------+
|               1 |
+-----------------+
1 row in set (0.001 sec)
```

e)Find the names of all suppliers who supply more than 30 different parts.

- select SP.supplier_name from supplier SP join supplies S ON SP.supplier_id = S.supplier_id GROUP BY SP.supplier_name HAVING count(distinct S.part_id) > 30;

```
MariaDB [database_three]> select SP.supplier_name from supplier SP join supp
lies S ON SP.supplier_id = S.supplier_id GROUP BY SP.supplier_name HAVING co
unt(distinct S.part_id) > 30;
Empty set (0.002 sec)
```

4) Consider the following relational schema and write the SQL for the following.

student(id, name)
enrolledIn(id, code)
subject(code, lecturer)

Command:

- create table student(id int primary key, name varchar(50));

```
MariaDB [database_four]> desc student;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| id    | int(11)     | NO   | PRI | NULL    |       |
| name  | varchar(50) | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
2 rows in set (0.011 sec)
```

- create table subject(code varchar(15) primary key, lecturer varchar(25));

```
MariaDB [database_four]> desc subject;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| code     | varchar(15) | NO   | PRI | NULL    |       |
| lecturer | varchar(25) | YES  |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
2 rows in set (0.009 sec)
```

- create table enrolledIn(id int, code varchar(15), primary key(id,code), foreign key(id) references student(id), foreign key(code) references subject(code));

```
MariaDB [database_four]> desc enrolledIn;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| id    | int(11)     | NO   | PRI | NULL    |       |
| code  | varchar(15) | NO   | PRI | NULL    |       |
+-------+-------------+------+-----+---------+-------+
2 rows in set (0.013 sec)
```

Commands:

- insert into student values(1, "Arjun"),(2, "Sitara"),(3, "Aayush"),(4, "Bijaya");

```
MariaDB [database_four]> select * from student;
+----+--------+
| id | name   |
+----+--------+
|  1 | Arjun  |
|  2 | Sitara |
|  3 | Aayush |
|  4 | Bijaya |
+----+--------+
4 rows in set (0.000 sec)
```

- insert into subject values("cs1500", "Yograj"),("cs3010", "Hector"),("cs3020", "Roger"),("cs1200", "Nabaraj");

```
MariaDB [database_four]> select * from subject;
+--------+----------+
| code   | lecturer |
+--------+----------+
| cs1200 | Nabaraj  |
| cs1500 | Yograj   |
| cs3010 | Hector   |
| cs3020 | Roger    |
+--------+----------+
4 rows in set (0.000 sec)
```

- insert into enrolledIn values(1, "cs1500"),(2, "cs3010"),(3, "cs3020"),(4, "cs1200");

```
MariaDB [database_four]> select * from enrolledIn;
+----+--------+
| id | code   |
+----+--------+
|  1 | cs1500 |
|  2 | cs3010 |
|  3 | cs3020 |
|  4 | cs1200 |
+----+--------+
4 rows in set (0.000 sec)
```

a. What are the names of students enrolled in cs3020?

- select S.name from student S join enrolledIn E ON S.id=E.id where E.code="cs3020";

```
MariaDB [database_four]> select S.name from student S join enrolledIn E ON S
.id=E.id where E.code="cs3020";
+--------+
| name   |
+--------+
| Aayush |
+--------+
1 row in set (0.001 sec)
```

b. Which subjects are Hector taking?

- select E.code from subject S join enrolledIn E ON S.code=E.code where lecturer = "Hector";

```
MariaDB [database_four]> select E.code from subject S join enrolledIn E ON S.c
ode=E.code where lecturer = "Hector";
+--------+
| code   |
+--------+
| cs3010 |
+--------+
1 row in set (0.001 sec)
```

c. Who teaches cs1500?

- select lecturer from subject where code="cs1500";

```
MariaDB [database_four]> select lecturer from subject where code="cs1500";
+----------+
| lecturer |
+----------+
| Yograj   |
+----------+
1 row in set (0.001 sec)
```

d. Who teaches cs1500 or cs3020?

- select distinct lecturer from subject where code="cs1500" or code="cs3020";

```
MariaDB [database_four]> select distinct lecturer from subject where code="cs15
00" or code="cs3020";
+----------+
| lecturer |
+----------+
| Yograj   |
| Jalaudin |
+----------+
2 rows in set (0.001 sec)
```

e. Who teaches at least two different subjects?

- select lecturer from subject group by lecturer having count(distinct code) >= 2;

```
MariaDB [database_four]> select lecturer from subject group by lecturer having
count(distinct code) >= 2;
Empty set (0.001 sec)
```

f. What are the names of students in cs1500 or cs3010?

- select distinct S.name from student S join enrolledIn E ON S.id=E.id where e.code="cs1500" OR e.code="cs3010";

```
MariaDB [database_four]> select distinct S.name from student S join enrolledIn
E ON S.id=E.id where e.code="cs1500" OR e.code="cs3010";
+--------+
| name   |
+--------+
| Arjun  |
| Sitara |
+--------+
2 rows in set (0.001 sec)
```

g. What are the names of students in both cs1500 and cs1200?

- select S.name from student S where S.id IN (select E1.id from enrolledIn E1 join enrolledIn E2 ON E1.id=E2.id where E1.code="cs1500" AND E2.code="cs1200");

```
MariaDB [database_four]> select S.name from student S where S.id IN (select E1.
id from enrolledIn E1 join enrolledIn E2 ON E1.id=E2.id where E1.code="cs1500"
AND E2.code="cs1200");
Empty set (0.001 sec)
```

h. What are the names of students in at least two different subjects?

- select S.name from student S join enrolledIn E ON S.id=E.id group by S.id, S.name having count(distinct E.code)>=2;

```
MariaDB [database_four]> select S.name from student S join enrolledIn E ON S.id
=E.id group by S.id, S.name having count(distinct E.code)>=2;
Empty set (0.001 sec)
```

i. What are the codes of all the subjects taught?

- select code from subject;

```
MariaDB [database_four]> select code from subject;
+--------+
| code   |
+--------+
| cs1200 |
| cs1500 |
| cs3010 |
| cs3020 |
+--------+
4 rows in set (0.000 sec)
```

j. What are the names of all the students?

- select name from student;

```
MariaDB [database_four]> select name from student;
+--------+
| name   |
+--------+
| Arjun  |
| Sitara |
| Aayush |
| Bijaya |
+--------+
4 rows in set (0.001 sec)
```

k. What are the names of all the students in cs1500?

- select S.name from student S join enrolledIn E ON S.id=E.id where E.code="cs1500";

```
MariaDB [database_four]> select S.name from student S join enrolledIn E ON S.id
=E.id where E.code="cs1500";
+-------+
| name  |
+-------+
| Arjun |
+-------+
1 row in set (0.000 sec)
```

l. What are the names of students taking a subject taught by Roger?

- select distinct S.name from student S join enrolledIn E ON S.id=E.id join subject SU ON E.code=SU.code where SU.lecturer="Roger";

```
MariaDB [database_four]> select distinct S.name from student S join enrolledIn
E ON S.id=E.id join subject SU ON E.code=SU.code where SU.lecturer="Roger";
+--------+
| name   |
+--------+
| Aayush |
+--------+
1 row in set (0.001 sec)
```

m. What are the names of students who are taking a subject not taught by Roger?

- select distinct S.name from student S join enrolledIn E ON S.id = E.id join subject SU ON E.code = SU.code where SU.lecturer != "Roger";

```
MariaDB [database_four]> select distinct S.name from student S join enrolledIn
E ON S.id = E.id join subject SU ON E.code = SU.code where SU.lecturer != "Roge
r";
+--------+
| name   |
+--------+
| Bijaya |
| Arjun  |
| Sitara |
+--------+
3 rows in set (0.001 sec)
```

5) Consider the following relational database.

Student(snum: integer, sname: string, major: string, level: string, age: integer)

Class(name: string, meets at: time, room: string, fid: integer)

Enrolled(snum: integer, cname: string)

Faculty(fid: integer, fname: string, deptid: integer)

Commands:

- create table student(snum int primary key, sname varchar(40), major varchar(40), level varchar(40), age int);

```
MariaDB [database_five]> desc student;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| snum  | int(11)     | NO   | PRI | NULL    |       |
| sname | varchar(40) | YES  |     | NULL    |       |
| major | varchar(40) | YES  |     | NULL    |       |
| level | varchar(40) | YES  |     | NULL    |       |
| age   | int(11)     | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
5 rows in set (0.012 sec)
```

- create table faculty(fid int primary key, fname varchar(50), deptid int);

```
MariaDB [database_five]> desc faculty;
+--------+-------------+------+-----+---------+-------+
| Field  | Type        | Null | Key | Default | Extra |
+--------+-------------+------+-----+---------+-------+
| fid    | int(11)     | NO   | PRI | NULL    |       |
| fname  | varchar(50) | YES  |     | NULL    |       |
| deptid | int(11)     | YES  |     | NULL    |       |
+--------+-------------+------+-----+---------+-------+
3 rows in set (0.013 sec)
```

- create table class(name varchar(50) primary key, meets_at time, room varchar(50), fid int, foreign key(fid) references faculty(fid));

```
MariaDB [database_five]> desc class;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| name     | varchar(50) | NO   | PRI | NULL    |       |
| meets_at | time        | YES  |     | NULL    |       |
| room     | varchar(50) | YES  |     | NULL    |       |
| fid      | int(11)     | YES  | MUL | NULL    |       |
+----------+-------------+------+-----+---------+-------+
4 rows in set (0.010 sec)
```

- create table enrolled(snum int, cname varchar(40), primary key(snum,cname));

```
MariaDB [database_five]> desc enrolled;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| snum  | int(11)     | NO   | PRI | NULL    |       |
| cname | varchar(40) | NO   | PRI | NULL    |       |
+-------+-------------+------+-----+---------+-------+
2 rows in set (0.019 sec)
```

Commands:

- insert into student values(1, "Arjun", "Database", "JR", 22),(2, "Simson", "Math", "SR", 21),(3, "Roman", "History", "FR", 23),(4, "Rohan", "Computer", "SR, 22),(5, "Manander", "Physics", "JR", 30);

```
MariaDB [database_five]> select * from student;
+------+----------+----------+-------+------+
| snum | sname    | major    | level | age  |
+------+----------+----------+-------+------+
|    1 | Arjun    | Database | JR    |   22 |
|    2 | Simson   | Math     | SR    |   21 |
|    3 | Roman    | History  | FR    |   23 |
|    4 | Rohan    | Computer | SR    |   22 |
|    5 | Manander | Physics  | JR    |   30 |
+------+----------+----------+-------+------+
5 rows in set (0.000 sec)
```

- insert into faculty values(1, "I.Teach", 101),(2, "Dr. Math", 102),(3, "Prof. History", 103),(4, "Dr. Physics", 104);

```
MariaDB [database_five]> select * from faculty;
+-----+--------------+--------+
| fid | fname        | deptid |
+-----+--------------+--------+
|   1 | I.Teach      |    101 |
|   2 | Dr.Math      |    102 |
|   3 | Prof. History |    103 |
|   4 | Dr. Physics  |    104 |
+-----+--------------+--------+
4 rows in set (0.000 sec)
```

- into class values("CS101", "10:00", "R128", 1),("Math101", "11:00", "R130", 2),("Hist201", "12:00", "R128", 3),("CS302", "14:00", "R128", 3),("Phys101", "15:00", "R140", 4);

```
MariaDB [database_five]> select * from class;
+---------+----------+------+------+
| name    | meets_at | room | fid  |
+---------+----------+------+------+
| CS101   | 10:00:00 | R128 |    1 |
| CS302   | 14:00:00 | R128 |    3 |
| Hist201 | 12:00:00 | R128 |    3 |
| Math101 | 11:00:00 | R130 |    2 |
| Phys101 | 15:00:00 | R140 |    4 |
+---------+----------+------+------+
5 rows in set (0.000 sec)
```

- insert into enrolled values(1, "CS101"),(2,"Math101"), (3, "Hist201")(4, "CS101"),(5,"Phys101"),(1, "CS302"),(2, "CS101");

```
MariaDB [database_five]> select * from enrolled;
+------+---------+
| snum | cname   |
+------+---------+
|    1 | CS101   |
|    1 | CS302   |
|    2 | CS101   |
|    2 | Math101 |
|    3 | Hist201 |
|    4 | CS101   |
|    5 | Phys101 |
+------+---------+
7 rows in set (0.001 sec)
```

Write the SQL for each of the following queries.

a. Find the names of all Juniors (Level = JR) who are enrolled in a class taught by I. Teach.

- select S.sname from student S join enrolled E ON S.snum=E.snum join class C ON E.cname=C.name join faculty F ON C.fid=F.fid where S.level = "JR" AND F.fname="I. Teach";

```
MariaDB [database_five]> select S.sname from student S join enrolled E ON S.snum
=E.snum join class C ON E.cname=C.name join faculty F ON C.fid=F.fid where S.lev
el = "JR" AND F.fname="I. Teach";
Empty set (0.001 sec)
```

b. Find the age of the oldest student who is either a History major or is enrolled in a course taught by I. Teach.

- select MAX(S.age) AS oldest_age from student S where S.major = "History" OR S.snum IN (select E.snum from enrolled E join class C ON E.cname = C.name join faculty F ON C.fid = F.fid where F.fname = "I. Teach");

```
MariaDB [database_five]> select MAX(S.age) AS oldest_age from student S where S
.major = "History" OR S.snum IN (select E.snum from enrolled E join class C ON
E.cname = C.name join faculty F ON C.fid = F.fid where F.fname = "I. Teach");
+------------+
| oldest_age |
+------------+
|         23 |
+------------+
1 row in set (0.001 sec)
```

c. Find the names of all classes that either meet in room R128 or have five or more students enrolled.

- select C.name from class C where C.room="R128" OR C.name IN (select E.cname from enrolled E group by E.cname having count(E.snum)>=5);

```
MariaDB [database_five]> select C.name from class C where C.room="R128" OR C.na
me IN (select E.cname from enrolled E group by E.cname having count(E.snum)>=5)
;
+---------+
| name    |
+---------+
| CS101   |
| CS302   |
| Hist201 |
+---------+
3 rows in set (0.001 sec)
```

d. Print the Level and the average age of students for that Level, for each Level.

- select S.level, AVG(S.age) AS average_age from student S group by S.level;

```
MariaDB [database_five]> select S.level, AVG(S.age) AS average_age from student
 S group by S.level;
+-------+-------------+
| level | average_age |
+-------+-------------+
| FR    |     23.0000 |
| JR    |     26.0000 |
| SR    |     21.5000 |
+-------+-------------+
3 rows in set (0.001 sec)
```

e. Find the names of students who are not enrolled in any class.

- select S.sname from student S where S.snum not in (select E.snum from enrolled E);

```
MariaDB [database_five]> select S.sname from student S where S.snum not in ( se
lect E.snum from enrolled E);
Empty set (0.001 sec)
```

6) Consider the following database schema. What are the referential integrity constraints that should be held on the schema? Write appropriate SQL DDL statements to define the database.

EMPLOYEE

FNAME	MINIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO

DEPARTMENT

DNAME	DNUMBER	MGRSSN	MGRSTARTDATE

DEPT_LOCATIONS

DNUMBER	DLOCATION

PROJECT

PNAME	PNUMBER	PLOCATION	DNUM

WORKS_ON

ESSN	PNO	HOURS

DEPENDENT

ESSN	DEPENDENT_NAME	SEX	BDATE	RELATIONSHIP

Commands: (DDL)

- create table employee(fname varchar(50), minit varchar(50), lname varchar(50), ssn char(10) primary key, bdate date, address varchar(100), sex char(1), salary decimal(10,2), superssn char(10), dno int, foreign key(superssn) references employee(ssn));

```
MariaDB [database_six]> desc employee;
+---------+---------------+------+-----+---------+-------+
| Field   | Type          | Null | Key | Default | Extra |
+---------+---------------+------+-----+---------+-------+
| fname   | varchar(50)   | YES  |     | NULL    |       |
| minit   | varchar(50)   | YES  |     | NULL    |       |
| lname   | varchar(50)   | YES  |     | NULL    |       |
| ssn     | char(10)      | NO   | PRI | NULL    |       |
| bdate   | date          | YES  |     | NULL    |       |
| address | varchar(100)  | YES  |     | NULL    |       |
| sex     | char(1)       | YES  |     | NULL    |       |
| salary  | decimal(10,2) | YES  |     | NULL    |       |
| superssn| char(10)      | YES  | MUL | NULL    |       |
| dno     | int(11)       | YES  |     | NULL    |       |
+---------+---------------+------+-----+---------+-------+
10 rows in set (0.019 sec)
```

- create table department(dname varchar(50), dnumber int primary key, mgrssn char(10), mgrstartdate date, foreign key(mgrssn) references employee(ssn));

```
MariaDB [database_six]> desc department;
+--------------+------------+------+-----+---------+-------+
| Field        | Type       | Null | Key | Default | Extra |
+--------------+------------+------+-----+---------+-------+
| dname        | varchar(50) | YES  |     | NULL    |       |
| dnumber      | int(11)    | NO   | PRI | NULL    |       |
| mgrssn       | char(10)   | YES  | MUL | NULL    |       |
| mgrstartdate | date       | YES  |     | NULL    |       |
+--------------+------------+------+-----+---------+-------+
4 rows in set (0.021 sec)
```

- alter table employee add constraint fk_dno foreign key(dno) references department(dnumber);

```
MariaDB [database_six]> desc employee;
+----------+--------------+------+-----+---------+-------+
| Field    | Type         | Null | Key | Default | Extra |
+----------+--------------+------+-----+---------+-------+
| fname    | varchar(50)  | YES  |     | NULL    |       |
| minit    | varchar(50)  | YES  |     | NULL    |       |
| lname    | varchar(50)  | YES  |     | NULL    |       |
| ssn      | char(10)     | NO   | PRI | NULL    |       |
| bdate    | date         | YES  |     | NULL    |       |
| address  | varchar(100) | YES  |     | NULL    |       |
| sex      | char(1)      | YES  |     | NULL    |       |
| salary   | decimal(10,2) | YES  |     | NULL    |       |
| superssn | char(10)     | YES  | MUL | NULL    |       |
| dno      | int(11)      | YES  | MUL | NULL    |       |
+----------+--------------+------+-----+---------+-------+
10 rows in set (0.016 sec)
```

- create table dept_locations (dnumber int, dlocation varchar(50), primary key (dnumber,dlocation), foreign key(dnumber) references department(dnumber));

```
MariaDB [database_six]> desc dept_locations;
+-----------+------------+------+-----+---------+-------+
| Field     | Type       | Null | Key | Default | Extra |
+-----------+------------+------+-----+---------+-------+
| dnumber   | int(11)    | NO   | PRI | NULL    |       |
| dlocation | varchar(50) | NO   | PRI | NULL    |       |
+-----------+------------+------+-----+---------+-------+
2 rows in set (0.018 sec)
```

- create table project(pname varchar(50), pnumber int primary key, plocation varchar(50), dnum int, foreign key(dnum) references department(dnumber));

```
MariaDB [database_six]> desc project;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| pname     | varchar(50) | YES  |     | NULL    |       |
| pnumber   | int(11)     | NO   | PRI | NULL    |       |
| plocation | varchar(50) | YES  |     | NULL    |       |
| dnum      | int(11)     | YES  | MUL | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
4 rows in set (0.012 sec)
```

- create table works_on(essn char(10), pno int, hours decimal(4,2), foreign key(essn) references employee(ssn), primary key(essn,pno), foreign key(pno) references project(pnumber));

```
MariaDB [database_six]> desc works_on;
+-------+--------------+------+-----+---------+-------+
| Field | Type         | Null | Key | Default | Extra |
+-------+--------------+------+-----+---------+-------+
| essn  | char(10)     | NO   | PRI | NULL    |       |
| pno   | int(11)      | NO   | PRI | NULL    |       |
| hours | decimal(4,2) | YES  |     | NULL    |       |
+-------+--------------+------+-----+---------+-------+
3 rows in set (0.016 sec)
```

- create table dependent(essn char(10), dependent_name varchar(30), sex char(1), bdate date, relationship varchar(30), primary key(essn, dependent_name), foreign key(essn) references employee(ssn));

```
MariaDB [database_six]> desc dependent;
+----------------+-------------+------+-----+---------+-------+
| Field          | Type        | Null | Key | Default | Extra |
+----------------+-------------+------+-----+---------+-------+
| essn           | char(10)    | NO   | PRI | NULL    |       |
| dependent_name | varchar(30) | NO   | PRI | NULL    |       |
| sex            | char(1)     | YES  |     | NULL    |       |
| bdate          | date        | YES  |     | NULL    |       |
| relationship   | varchar(30) | YES  |     | NULL    |       |
+----------------+-------------+------+-----+---------+-------+
5 rows in set (0.016 sec)
```

Commands: (DML)

- insert into employee values("Arjun", "A", "Smith", "123", "2002-05-15", "10 devinagar, ktm", "M", 60000, NULL, NULL);

```
MariaDB [database_six]> select * from employee;
+-------+-------+-------+-----+------------+-------------------+------+----------+----------+------+
| fname | minit | lname | ssn | bdate      | address           | sex  | salary   | superssn | dno  |
+-------+-------+-------+-----+------------+-------------------+------+----------+----------+------+
| Arjun | A     | Smith | 123 | 2002-05-15 | 10 devinagar, ktm | M    | 60000.00 | NULL     | NULL |
+-------+-------+-------+-----+------------+-------------------+------+----------+----------+------+
1 row in set (0.001 sec)
```

- insert into department values("Research", 1, "123", "2005-01-01"),("Development", 2, NULL, "2006-05-04"),("Sales", 3, NULL, "2010-03-10"), ("Support", 5, NULL, "2013-08-12");

```
MariaDB [database_six]> select * from department;
+-------------+---------+--------+---------------+
| dname       | dnumber | mgrssn | mgrstartdate  |
+-------------+---------+--------+---------------+
| Research    |       1 | 123    | 2005-01-01    |
| Development |       2 | NULL   | 2006-05-04    |
| Sales       |       3 | NULL   | 2010-03-10    |
| Support     |       5 | NULL   | 2013-08-12    |
+-------------+---------+--------+---------------+
4 rows in set (0.001 sec)
```

- insert into employee values("Lokesh", "B", "Yogi", "456", "1999-01-31", "32 Narayaneshwor, lalitpur", "M", 67900, "123", 2),("Pabitra", "C", "Smith", "789", "2003-10-11", "15 Bhimnagar, bhaktapur", "F", 64000, "123", 1),("Sangita", "D", "Rimal", "011", "2005-04-06", "28 tirpura, gorkha", "F", 55000, "789", 5),("Aayush", "E", "Gautam", "022", "1998-09-08", "26 thamel, ktm", "M", 57000, "011", 5);

```
MariaDB [database_six]> select * from employee;
+---------+-------+--------+-----+------------+---------------------------+------+----------+----------+------+
| fname   | minit | lname  | ssn | bdate      | address                   | sex  | salary   | superssn | dno  |
+---------+-------+--------+-----+------------+---------------------------+------+----------+----------+------+
| Sangita | D     | Rimal  | 011 | 2005-04-06 | 28 tirpura, gorkha        | F    | 55000.00 | 789      |    5 |
| Aayush  | E     | Gautam | 022 | 1998-09-08 | 26 thamel, ktm            | M    | 57000.00 | 011      |    5 |
| Arjun   | A     | Smith  | 123 | 2002-05-15 | 10 devinagar, ktm         | M    | 60000.00 | NULL     | NULL |
| Lokesh  | B     | Yogi   | 456 | 1999-01-31 | 32 Narayaneshwor, lalitpur | M    | 67900.00 | 123      |    2 |
| Pabitra | C     | Smith  | 789 | 2003-10-11 | 15 Bhimnagar, bhaktapur   | F    | 64000.00 | 123      |    1 |
+---------+-------+--------+-----+------------+---------------------------+------+----------+----------+------+
5 rows in set (0.000 sec)
```

- update employee set superssn="789", dno=1 where ssn="123";

```
MariaDB [database_six]> select * from employee;
+---------+-------+--------+-----+------------+---------------------------+------+----------+----------+------+
| fname   | minit | lname  | ssn | bdate      | address                   | sex  | salary   | superssn | dno  |
+---------+-------+--------+-----+------------+---------------------------+------+----------+----------+------+
| Sangita | D     | Rimal  | 011 | 2005-04-06 | 28 tirpura, gorkha        | F    | 55000.00 | 789      |    5 |
| Aayush  | E     | Gautam | 022 | 1998-09-08 | 26 thamel, ktm            | M    | 57000.00 | 011      |    5 |
| Arjun   | A     | Smith  | 123 | 2002-05-15 | 10 devinagar, ktm         | M    | 60000.00 | 789      |    1 |
| Lokesh  | B     | Yogi   | 456 | 1999-01-31 | 32 Narayaneshwor, lalitpur | M    | 67900.00 | 123      |    2 |
| Pabitra | C     | Smith  | 789 | 2003-10-11 | 15 Bhimnagar, bhaktapur   | F    | 64000.00 | 123      |    1 |
+---------+-------+--------+-----+------------+---------------------------+------+----------+----------+------+
5 rows in set (0.001 sec)
```

- update department set mgrssn="789" where dname="Development";
- update department set mgrssn="011" where dname="Support";
- update department set mgrssn="022" where dname="Sales";

```
MariaDB [database_six]> select * from department;
+-------------+---------+--------+--------------+
| dname       | dnumber | mgrssn | mgrstartdate |
+-------------+---------+--------+--------------+
| Research    |       1 | 123    | 2005-01-01   |
| Development |       2 | 789    | 2006-05-04   |
| Sales       |       3 | 022    | 2010-03-10   |
| Support     |       5 | 011    | 2013-08-12   |
+-------------+---------+--------+--------------+
4 rows in set (0.000 sec)
```

- insert into dept_locations values(1, "Kathmandu"),(2, "Lalitpur"),(5,"Stafford");

```
MariaDB [database_six]> select * from dept_locations;
+---------+-----------+
| dnumber | dlocation |
+---------+-----------+
|       1 | Kathmandu |
|       2 | Lalitpur  |
|       5 | Stafford  |
+---------+-----------+
3 rows in set (0.000 sec)
```

- insert into project values("Project Alpha", 1, "Kathmandu", 1),("Project Beta", 2, "Lalitpur", 2), ("Project Gamma", 3, "Bhaktapur", 5), ("Project Delta", 4, "Bhaktapur", 5);

```
MariaDB [database_six]> select * from project;
+---------------+---------+-----------+------+
| pname         | pnumber | plocation | dnum |
+---------------+---------+-----------+------+
| Project Alpha |       1 | Kathmandu |    1 |
| Project Beta  |       2 | Lalitpur  |    2 |
| Project Gamma |       3 | Stafford  |    5 |
| Project Delta |       4 | Stafford  |    5 |
+---------------+---------+-----------+------+
4 rows in set (0.000 sec)
```

- insert into works_on values("123", 1, 40),("456", 2, 20),("789", 1, 30),("011", 3, 25), ("022", 4, 25);

```
MariaDB [database_six]> select * from works_on;
+------+-----+-------+
| essn | pno | hours |
+------+-----+-------+
| 011  |   3 | 25.00 |
| 022  |   4 | 25.00 |
| 123  |   1 | 40.00 |
| 456  |   2 | 20.00 |
| 789  |   1 | 30.00 |
+------+-----+-------+
5 rows in set (0.000 sec)
```

- insert into dependent values("123","Sarina", "F", "2003-04-05", "Friend"),(123, "Sitaram", "M", "2006-06-15", "Brother"), ("789", "Simson", "M", "2002-04-08", "Brother"),("011", "Sarita", "F", "2010-10-10", "Sister"), ("022", "Rohan", "M", "2011-01-05", "Friend");

```
MariaDB [database_six]> select * from dependent;
+------+----------------+------+------------+--------------+
| essn | dependent_name | sex  | bdate      | relationship |
+------+----------------+------+------------+--------------+
| 011  | Sarita         | F    | 2010-10-10 | Sister       |
| 022  | Rohan          | M    | 2011-01-05 | Friend       |
| 123  | Sarina         | F    | 2003-04-05 | Friend       |
| 123  | Sitaram        | M    | 2006-06-15 | Brother      |
| 789  | Simson         | M    | 2002-04-08 | Brother      |
+------+----------------+------+------------+--------------+
5 rows in set (0.001 sec)
```

Write the SQL queries for the following:

a. Retrieve the name and address of all employees who work for the 'Research' department.

Commands:

- select E.fname, E.lname, E.address from employee E join department D on E.dno=D.dnumber where D.dname="Research";

```
MariaDB [database_six]> select E.fname, E.lname, E.address from employee E j
oin department D on E.dno=D.dnumber where D.dname="Research";
+---------+-------+------------------------+
| fname   | lname | address                |
+---------+-------+------------------------+
| Arjun   | Smith | 10 devinagar, ktm      |
| Pabitra | Smith | 15 Bhimnagar, bhaktapur |
+---------+-------+------------------------+
2 rows in set (0.001 sec)
```

b. For every project located in 'Stafford', list the project number, the controlling department number, and the department manager's last name, address, and birth date.

- select P.pnumber, P.dnum, E.lname, E.address, E.bdate from project P join department D on P.dnum = D.dnumber join employee E on D.mgrssn = E.ssn where P.plocation = "Stafford";

```
MariaDB [database_six]> select P.pnumber, P.dnum, E.lname, E.address, E.bdat
e from project P join department D on P.dnum = D.dnumber join employee E on
D.mgrssn = E.ssn where P.plocation = "Stafford";
+---------+------+-------+--------------------+------------+
| pnumber | dnum | lname | address            | bdate      |
+---------+------+-------+--------------------+------------+
|       3 |    5 | Rimal | 28 tirpura, gorkha | 2005-04-06 |
|       4 |    5 | Rimal | 28 tirpura, gorkha | 2005-04-06 |
+---------+------+-------+--------------------+------------+
2 rows in set (0.001 sec)
```

c. Find the names of employees who work on all the projects controlled by department number 5.

- select E.fname, E.lname from employee E where not exists (select P.pnumber from project P where P.dnum=5 and not exists(select W.essn from works_on W where W.pno=P.pnumber and W.essn=E.ssn));

```
MariaDB [database_six]> select E.fname, E.lname from employee E where not ex
ists (select P.pnumber from project P where P.dnum=5 and not exists(select W
.essn from works_on W where W.pno=P.pnumber and W.essn=E.ssn));
Empty set (0.001 sec)
```

d. Make a list of project numbers for projects that involve an employee whose last name is 'Smith', either as a worker or as a manager of the department that controls the project.

- select distinct P.pnumber from project P join works_on W on P.pnumber = W.pno join employee E on W.essn = E.ssn where E.lname = "Smith" or P.dnum in(select D.dnumber from department D join employee MGR on D.mgrssn = MGR.ssn where MGR.lname = "Smith");

```
MariaDB [database_six]> select distinct P.pnumber from project P join works_
on W on P.pnumber = W.pno join employee E on W.essn = E.ssn where E.lname =
"Smith" or P.dnum in( select D.dnumber from department D join employee MGR o
n D.mgrssn = MGR.ssn where MGR.lname = "Smith");
+---------+
| pnumber |
+---------+
|       1 |
|       2 |
+---------+
2 rows in set (0.002 sec)
```

e. List the names of all employees with two or more dependents

- select E.fname, E.lname from employee E join dependent D on E.ssn = D.essn group by E.ssn, E.fname, E.lname having count(D.dependent_name) >=2;

```
MariaDB [database_six]> select E.fname, E.lname from employee E join depende
nt D on E.ssn = D.essn group by E.ssn, E.fname, E.lname having count(D.depen
dent_name) >=2;
+-------+-------+
| fname | lname |
+-------+-------+
| Arjun | Smith |
+-------+-------+
1 row in set (0.001 sec)
```

f. List the names of managers who have at least one dependent

- select distinct E.fname, E.lname from employee E join department D on E.ssn = D.mgrssn join dependent DEP on E.ssn = DEP.essn;

```
MariaDB [database_six]> select distinct E.fname, E.lname from employee E joi
n department D on E.ssn = D.mgrssn join dependent DEP on E.ssn = DEP.essn;
+---------+--------+
| fname   | lname  |
+---------+--------+
| Sangita | Rimal  |
| Aayush  | Gautam |
| Arjun   | Smith  |
| Pabitra | Smith  |
+---------+--------+
4 rows in set (0.001 sec)
```

g. Retrieve the names of employees who have no dependents.

- select E.fname, E.lname from employee E left join dependent D on E.ssn = D.essn where D.essn IS NULL;

```
MariaDB [database_six]> select E.fname, E.lname from employee E left join
dependent D on E.ssn = D.essn where D.essn IS NULL;
+--------+-------+
| fname  | lname |
+--------+-------+
| Lokesh | Yogi  |
+--------+-------+
1 row in set (0.001 sec)
```

7) What is a view? And explain the advantages of the view. Write a command to create a view of employees working in the IT department.

- A **view** in SQL is a virtual table that is created by a SELECT query on one or more tables. It doesn't store the data itself but instead shows the result of a predefined query every time the view is accessed. The view can be used like a regular table for querying but does not physically store data.

Advantages of Views:

1. **Simplicity**: Views simplify complex queries by encapsulating them within a single object. Users can query the view without needing to know the underlying tables or the complexity of the original query.
2. **Security**: Views can be used to limit access to specific columns or rows in a table. For example, sensitive information can be hidden from certain users by creating a view that only displays non-sensitive data.
3. **Data Abstraction**: Views provide a layer of abstraction over the physical schema. If the underlying schema changes (e.g., columns are added or renamed), the view can remain the same, offering stability for applications that depend on it.
4. **Reusability**: Views allow you to reuse complex SQL logic across multiple queries. Once a view is defined, it can be accessed in multiple queries without repeating the complex logic.
5. **Maintenance**: Views can simplify database maintenance by reducing redundancy in query logic, ensuring that changes to business rules need to be updated only once (within the view) instead of in multiple queries.

Creating a View for Employees in the IT Department

Employee (id, name, department, salary)

- create table employee(id int primary key, name varchar(50), department varchar(50), salary decimal(10,2));

```
MariaDB [database_seven]> select * from employee;
+----+---------+------------+----------+
| id | name    | department | salary   |
+----+---------+------------+----------+
|  1 | Ram     | IT         | 60000.00 |
|  2 | Shyam   | HR         | 45000.00 |
|  3 | Hari    | IT         | 75000.00 |
|  4 | Krishna | Finance    | 55000.00 |
+----+---------+------------+----------+
4 rows in set (0.001 sec)
```

Insert Data:

- insert into employee values(1, "Ram", "IT", 60000),(2, "Shyam", "HR", 45000),(3, "Hari", "IT", 75000),(4, "Krishna", "Finance", 55000);

```
MariaDB [database_seven]> select * from employee;
+----+---------+------------+----------+
| id | name    | department | salary   |
+----+---------+------------+----------+
|  1 | Ram     | IT         | 60000.00 |
|  2 | Shyam   | HR         | 45000.00 |
|  3 | Hari    | IT         | 75000.00 |
|  4 | Krishna | Finance    | 55000.00 |
+----+---------+------------+----------+
4 rows in set (0.001 sec)
```

Creating View:

- create view IT_Employees AS select id, name, department, salary from Employee where department = "IT";

```
MariaDB [database_seven]> desc IT_Employees;
+------------+---------------+------+-----+---------+-------+
| Field      | Type          | Null | Key | Default | Extra |
+------------+---------------+------+-----+---------+-------+
| id         | int(11)       | NO   |     | NULL    |       |
| name       | varchar(50)   | YES  |     | NULL    |       |
| department | varchar(50)   | YES  |     | NULL    |       |
| salary     | decimal(10,2) | YES  |     | NULL    |       |
+------------+---------------+------+-----+---------+-------+
4 rows in set (0.021 sec)
```

- select * from IT_Employees;

```
MariaDB [database_seven]> select * from IT_Employees;
+----+------+------------+----------+
| id | name | department | salary   |
+----+------+------------+----------+
|  1 | Ram  | IT         | 60000.00 |
|  3 | Hari | IT         | 75000.00 |
+----+------+------------+----------+
2 rows in set (0.001 sec)
```

- update IT_Employees set salary=salary+5000 where id=1;

```
MariaDB [database_seven]> select * from IT_Employees;
+----+------+------------+----------+
| id | name | department | salary   |
+----+------+------------+----------+
|  1 | Ram  | IT         | 65000.00 |
|  3 | Hari | IT         | 75000.00 |
+----+------+------------+----------+
2 rows in set (0.001 sec)
```