# NEPAL COLLEGE OF INFORMATION TECHNOLOGY

## BALKUMARI LALITPUR


**(Affiliated To Pokhara University)**

**SUBJECT : Database Management System**

**ASSIGNMENT # 3**

**Submitted By :**

**Name :** Pradip Dhungana

**Roll No :** 201751

**Semester :** 4th

**Submitted To :**

**Name :** Amit K. Shrivastava

**Department of Software**

**1. Consider the insurance database of Figure1 below, where the primary keys are underlined. Construct the following SQL queries for this relational database.**

*person* (*driver-id*, *name*, *address*)

*car* (*license*, *model*, *year*)

*accident* (*report-number*, *date*, *location*)

*owns* (*driver-id*, *license*)

*participated* (*driver-id*, *car*, *report-number*, *damage-amount*)

**fig1: Insurance database**

Answer :

➔ **Creating a Database :**

= create database insurance;

OUTPUT :

```
[zoro@archlinux ~]$ mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 7
Server version: 10.11.4-MariaDB Arch Linux

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
4 rows in set (0.001 sec)

MariaDB [(none)]> create database insurance;
Query OK, 1 row affected (0.000 sec)

MariaDB [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| insurance          |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
5 rows in set (0.001 sec)

MariaDB [(none)]>
```

➔ **Creating Tables and assigning Foreign Keys :**

**a) PERSON Table :**

= CREATE TABLE person(driver_id INT PRIMARY KEY,name VARCAR(50),address VARCHAR(100));

**b) Car Table :**

= CREATE TABLE car(license VARCHAR(20) PRIMARY KEY,model VARHCAR(50),year INT);

**c) Accident Table :**

= CREATE TABLE accident(report_number VARCHAR(25),date DATE,location VARCHAR(100));

**d) Owns Table :**

= CREATE TABLE owns(driver_id INT, license VARCHAR(20),FOREIGN KEY (driver_id) REFERENCES person(driver_id),FOREIGN KEY (license) REFERENCES car(license));

**e) Participated Table :**

= CREATE TABLE participated(driver_id INT, car VARCHAR(20),report_number VARCHAR(25),damage_amount DECIMAL(10, 2),FOREIGN KEY (driver_id) REFERENCES person(driver_id),FOREIGN KEY (car) REFERENCES car(license),FOREIGN KEY (report_number) REFERENCES accident(report_number));

**OUTPUT :**

```
MariaDB [(none)]> use insurance;
MariaDB [insurance]> create table person(driver_id INT PRIMARY KEY,name VARCHAR(50),address VARCHAR(100));
Query OK, 0 rows affected (0.016 sec)

MariaDB [insurance]> create table car(license VARCHAR(20) PRIMARY KEY,model VARCHAR(50),year INT);
Query OK, 0 rows affected (0.014 sec)

MariaDB [insurance]> CREATE TABLE accident ( report_number INT PRIMARY KEY, date DATE, location VARCHAR(100) );
Query OK, 0 rows affected (0.016 sec)

MariaDB [insurance]> CREATE TABLE owns ( driver_id INT, license VARCHAR(20), FOREIGN KEY (driver_id) REFERENCES person(driver_id), FOREIGN KEY (license) REFERE
NCES car(license) );
Query OK, 0 rows affected (0.018 sec)

MariaDB [insurance]> CREATE TABLE participated ( driver_id INT, car VARCHAR(20), report_number INT, damage_amount DECIMAL(10, 2), FOREIGN KEY (driver_id) REFER
ENCES person(driver_id), FOREIGN KEY (car) REFERENCES car(license), FOREIGN KEY (report_number) REFERENCES accident(report_number) );
Query OK, 0 rows affected (0.010 sec)

MariaDB [insurance]> show tables;
+---------------------+
| Tables_in_insurance |
+---------------------+
| accident            |
| car                 |
| owns                |
| participated        |
| person              |
+---------------------+
5 rows in set (0.000 sec)
```

**Inserting Data :**

➔ **Praticipated Table :**

= INSERT INTO participated VALUES (1, 'ABC123', 'BR3197', 5000.00), (2, 'DEF456', 'BR3198', 2500.00), (3, 'GHI789', 'BR3199', 10000.00), (4, 'JKL012', 'BR3200', 7500.00), (5, 'MNO345', 'BR3201', 3000.00), (6, 'PQR678', 'BR3202', 6000.00), (7, 'STU901','BR3203', 4000.00), (8, 'VWX234', 'BR3204', 8000.00), (9, 'YZA567', 'BR3205', 1500.00), (10, 'BCD890', 'BR3206', 2000.00), (11, 'AABB2001', 'BR3197', 5000.00), (12, 'CCDD2002', 'BR3208', 2500.00);

➔ **Accident Table :**

= INSERT INTO accident VALUES ('BR3197', 2023, 'Kathmandu'), ('BR3198', 2023, 'Pokhara'), ('BR3199', 2023, 'Biratnagar'), ('BR3200', 2023, 'Lalitpur'), ('BR3201', 2023, 'Bhaktapur'), ('BR3202', 2023, 'Dharan'), ('BR3203', 2023, 'Butwal'), ('BR3204', 2023, 'Hetauda'), ('BR3205', 2023, 'Nepalgunj'), ('BR3206', 2023, 'Itahari'),('BR3197', 2020, 'Kathmandu'), ('BR3208', 2020, 'Pokhara');

➔ **Owns Table :**

= INSERT INTO owns VALUES (1, 'ABC123'), (2, 'DEF456'), (3, 'GHI789'), (4, 'JKL012'), (5, 'MNO345'), (6, 'PQR678'), (7, 'STU901'), (8, 'VWX234'), (9, 'YZA567'), (10, 'BCD890'),(11, 'AABB2001'), (12, 'CCDD2002');

➔ **Car Table :**

= INSERT INTO car VALUES ('ABC123', 'Toyota Camry', 2018), ('DEF456', 'Honda Civic', 2020), ('GHI789', 'Ford Mustang', 2019), ('JKL012', 'Chevrolet Silverado', 2021), ('MNO345', 'BMW 3 Series', 2017), ('PQR678', 'Tesla Model S', 2022), ('STU901', 'Volkswagen Jetta', 2016), ('VWX234', 'Mercedes-Benz C-Class', 2020), ('YZA567', 'Audi A4', 2019), ('BCD890', 'Hyundai Elantra', 2015),('AABB2001', 'Range Rover', 2019), ('CCDD2002', 'Toyota Corolla', 2020);

➔ **Person Table :**

= INSERT INTO person VALUES (1, 'John Doe', 'Kathmandu'), (2, 'Jane Smith', 'Pokhara'), (3, 'David Johnson', 'Biratnagar'), (4, 'Sarah Williams', 'Lalitpur'), (5, 'Michael Brown', 'Bhaktapur'), (6, 'Emily Davis', 'Dharan'), (7, 'Robert Taylor', 'Butwal'), (8, 'Olivia Wilson', 'Hetauda'), (9, 'James Anderson', 'Nepalgunj'), (10,

'Sophia Martin', 'Itahari'),(11, 'Black Smith', 'Kathmandu'), (12, 'John Black', 'Pokhara');

**OUTPUT :**

```
MariaDB [insurance]> SELECT * FROM person; SELECT * FROM car; SELECT * FROM accident; SELECT * FROM owns; SELECT * FROM participated;
+-----------+---------------+------------+
| driver_id | name          | address    |
+-----------+---------------+------------+
|         1 | John Doe      | Kathmandu  |
|         2 | Jane Smith    | Pokhara    |
|         3 | David Johnson | Biratnagar |
|         4 | Sarah Williams | Lalitpur   |
|         5 | Michael Brown | Bhaktapur  |
|         6 | Emily Davis   | Dharan     |
|         7 | Robert Taylor | Butwal     |
|         8 | Olivia Wilson | Hetauda    |
|         9 | James Anderson | Nepalgunj  |
|        10 | Sophia Martin | Itahari    |
|        11 | Black Smith   | Kathmandu  |
|        12 | John Black    | Pokhara    |
+-----------+---------------+------------+
12 rows in set (0.000 sec)

+----------+----------------------+------+
| license  | model                | year |
+----------+----------------------+------+
| AABB2001 | Range Rover          | 2019 |
| ABC123   | Toyota Camry         | 2018 |
| BCD890   | Hyundai Elantra      | 2015 |
| CCDD2002 | Toyota Corolla       | 2020 |
| DEF456   | Honda Civic          | 2020 |
| GHI789   | Ford Mustang         | 2019 |
| JKL012   | Chevrolet Silverado  | 2021 |
| MNO345   | BMW 3 Series         | 2017 |
| PQR678   | Tesla Model S        | 2022 |
| STU901   | Volkswagen Jetta     | 2016 |
| VWX234   | Mercedes-Benz C-Class | 2020 |
| YZA567   | Audi A4              | 2019 |
+----------+----------------------+------+
12 rows in set (0.000 sec)

+---------------+------+------------+
| report_number | date | location   |
+---------------+------+------------+
|             1 | 2023 | Kathmandu  |
|             2 | 2023 | Pokhara    |
|             3 | 2023 | Biratnagar |
|             4 | 2023 | Lalitpur   |
|             5 | 2023 | Bhaktapur  |
|             6 | 2023 | Dharan     |
|             7 | 2023 | Butwal     |
|             8 | 2023 | Hetauda    |
|             9 | 2023 | Nepalgunj  |
|            10 | 2023 | Itahari    |
|            11 | 2020 | Kathmandu  |
|            12 | 2020 | Pokhara    |
+---------------+------+------------+
12 rows in set (0.000 sec)

+-----------+----------+
| driver_id | license  |
+-----------+----------+
|         1 | ABC123   |
|         2 | DEF456   |
|         3 | GHI789   |
|         4 | JKL012   |
|         5 | MNO345   |
|         6 | PQR678   |
|         7 | STU901   |
|         8 | VWX234   |
|         9 | YZA567   |
|        10 | BCD890   |
|         1 | ABC123   |
|         2 | DEF456   |
|         3 | GHI789   |
|         4 | JKL012   |
|         5 | MNO345   |
|         6 | PQR678   |
|         7 | STU901   |
|         8 | VWX234   |
|         9 | YZA567   |
|        10 | BCD890   |
|        11 | AABB2001 |
|        12 | CCDD2002 |
+-----------+----------+
22 rows in set (0.000 sec)

+-----------+----------+---------------+---------------+
| driver_id | car      | report_number | damage_amount |
+-----------+----------+---------------+---------------+
|         1 | ABC123   |             1 |       5000.00 |
|         2 | DEF456   |             2 |       2500.00 |
|         3 | GHI789   |             3 |      10000.00 |
|         4 | JKL012   |             4 |       7500.00 |
|         5 | MNO345   |             5 |       3000.00 |
|         6 | PQR678   |             6 |       6000.00 |
|         7 | STU901   |             7 |       4000.00 |
|         8 | VWX234   |             8 |       8000.00 |
|         9 | YZA567   |             9 |       1500.00 |
|        10 | BCD890   |            10 |       2000.00 |
|        11 | AABB2001 |            11 |       5000.00 |
|        12 | CCDD2002 |            12 |       2500.00 |
+-----------+----------+---------------+---------------+
12 rows in set (0.000 sec)
```

**a.** Find the total number of people who owned cars that were involved in accidents in 2020.

= SELECT COUNT(DISTINCT driver_id) AS total_owners FROM owns WHERE license IN (SELECT car FROM participated WHERE report_number IN(SELECT report_number FROM accident WHERE YEAR(date) = 2020));

**OUTPUT :**

```
MariaDB [insurance]> SELECT COUNT(DISTINCT driver_id) AS total_owners FROM owns WHERE license IN ( SELECT car FROM participated WHERE report_number IN ( SELECT report_number
 FROM accident WHERE YEAR(date) = 2020 ) );
+--------------+
| total_owners |
+--------------+
|            2 |
+--------------+
1 row in set (0.011 sec)

MariaDB [insurance]>
```

**b.** Find the number of accidents in which the cars belonging to “Black Smith” were involved.

= SELECT COUNT(*) AS total_accidents FROM participated WHERE car IN (SELECT license FROM owns WHERE driver_id = (SELECT driver_id FROM person WHERE name = ‘Black Smith’));

**OUTPUT :**

```
MariaDB [insurance]> SELECT COUNT(*) AS total_accidents FROM participated WHERE car IN ( SELECT license FROM owns WHERE driver_id = ( SELECT driver_id FROM person WHERE name
 = 'Black Smith' ) );
+-----------------+
| total_accidents |
+-----------------+
|               1 |
+-----------------+
1 row in set (0.001 sec)

MariaDB [insurance]>
```

**c.** Delete the Range Rover belonging to “Black Smith”.

= DELETE FROM owns WHERE license IN ( SELECT license FROM owns WHERE driver_id = ( SELECT driver_id FROM person WHERE name = 'Black Smith' ) ) AND license IN ( SELECT license FROM car WHERE model = 'Range Rover' );

**OUTPUT :**

```
MariaDB [insurance]> DELETE FROM owns WHERE license IN ( SELECT license FROM owns WHERE driver_id = ( SELECT driver_id FROM person WHERE name = 'Black Smith' ) ) AND license IN ( SELECT license FROM car WHERE model = 'Range Rover' );
Query OK, 0 rows affected (0.001 sec)

MariaDB [insurance]> SELECT * FROM owns WHERE driver_id = ( SELECT driver_id FROM person WHERE name = 'Black Smith' ) AND license IN ( SELECT license FROM car WHERE model = 'Range Rover' );
Empty set (0.001 sec)

MariaDB [insurance]> select * from owns;
+-----------+----------+
| driver_id | license  |
+-----------+----------+
|         1 | ABC123   |
|        10 | BCD890   |
|        12 | CCDD2002 |
|         2 | DEF456   |
|         3 | GHI789   |
|         4 | JKL012   |
|         5 | MNO345   |
|         6 | PQR678   |
|         7 | STU901   |
|         8 | VWX234   |
|         9 | YZA567   |
+-----------+----------+
11 rows in set (0.000 sec)
```

**d.** Update the damage amount for the car with license number “AABB2001” in the accident with report number “BR3197” to $4000.

= UPDATE participated SET damage_amount = 4000.00 WHERE car = 'AABB2001' AND report_number = 'BR3197';

**OUTPUT :**

```
MariaDB [insurance]> select * from participated;
+-----------+----------+---------------+---------------+
| driver_id | car      | report_number | damage_amount |
+-----------+----------+---------------+---------------+
|         1 | ABC123   | BR3197        |       5000.00 |
|         2 | DEF456   | BR3198        |       2500.00 |
|         3 | GHI789   | BR3199        |      10000.00 |
|         4 | JKL012   | BR3200        |       7500.00 |
|         5 | MNO345   | BR3201        |       3000.00 |
|         6 | PQR678   | BR3202        |       6000.00 |
|         7 | STU901   | BR3203        |       4000.00 |
|         8 | VWX234   | BR3204        |       8000.00 |
|         9 | YZA567   | BR3205        |       1500.00 |
|        10 | BCD890   | BR3206        |       2000.00 |
|        11 | AABB2001 | BR3197        |       5000.00 |
|        12 | CCDD2002 | BR3208        |       2500.00 |
+-----------+----------+---------------+---------------+
12 rows in set (0.001 sec)

MariaDB [insurance]> UPDATE participated SET damage_amount = 4000.00 WHERE car = 'AABB2001' AND report_number = 'BR3197';
Query OK, 1 row affected (0.001 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [insurance]> select * from participated;
+-----------+----------+---------------+---------------+
| driver_id | car      | report_number | damage_amount |
+-----------+----------+---------------+---------------+
|         1 | ABC123   | BR3197        |       5000.00 |
|         2 | DEF456   | BR3198        |       2500.00 |
|         3 | GHI789   | BR3199        |      10000.00 |
|         4 | JKL012   | BR3200        |       7500.00 |
|         5 | MNO345   | BR3201        |       3000.00 |
|         6 | PQR678   | BR3202        |       6000.00 |
|         7 | STU901   | BR3203        |       4000.00 |
|         8 | VWX234   | BR3204        |       8000.00 |
|         9 | YZA567   | BR3205        |       1500.00 |
|        10 | BCD890   | BR3206        |       2000.00 |
|        11 | AABB2001 | BR3197        |       4000.00 |
|        12 | CCDD2002 | BR3208        |       2500.00 |
+-----------+----------+---------------+---------------+
12 rows in set (0.000 sec)
```

**2. Consider the employee database of Figure 2, where the primary keys are underlined. Give an expression in SQL for each of the following queries.**

*employee* (*employee-name*, *street*, *city*)

*works* (*employee-name*, *company-name*, *salary*)

*company* (*company-name*, *city*)

*manages* (*employee-name*, *manager-name*)

**Figure 2**. Employee database.

Answer :

➔ **Creating a Database :**

= create database employee;

OUTPUT :

```
MariaDB [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| insurance          |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
5 rows in set (0.044 sec)

MariaDB [(none)]> create database employee;
Query OK, 1 row affected (0.012 sec)

MariaDB [(none)]> use employee;
Database changed
```

➔ **Creating Tables and assigning Foreign Keys :**

**a) Employee Table :**

= CREATE TABLE employee ( employee_name VARCHAR(50) PRIMARY KEY, street VARCHAR(50), city VARCHAR(50) );

**b) Works Table :**

= CREATE TABLE works ( employee_name VARCHAR(50), company_name VARCHAR(50), salary DECIMAL(10, 2), PRIMARY KEY (employee_name, company_name), FOREIGN KEY (employee_name) REFERENCES employee(employee_name) );

**c) Company Table :**

= CREATE TABLE company ( company_name VARCHAR(50) PRIMARY KEY, city VARCHAR(50) );

**d) Manages Table :**

= CREATE TABLE manages ( employee_name VARCHAR(50), manager_name VARCHAR(50), PRIMARY KEY (employee_name, manager_name), FOREIGN KEY (employee_name) REFERENCES employee(employee_name), FOREIGN KEY (manager_name) REFERENCES employee(employee_name) );

**OUTPUT :**

```
MariaDB [(none)]> use employee;
Database changed
MariaDB [employee]> CREATE TABLE employee ( employee_name VARCHAR(50) PRIMARY KEY, street VARCHAR(50), city VARCHAR(50) );
Query OK, 0 rows affected (0.024 sec)

MariaDB [employee]> CREATE TABLE works ( employee_name VARCHAR(50), company_name VARCHAR(50), salary DECIMAL(10, 2), PRIMARY KEY (employee_name, company_name), FOREIGN KEY (
employee_name) REFERENCES employee(employee_name) );
Query OK, 0 rows affected (0.005 sec)

MariaDB [employee]> CREATE TABLE company ( company_name VARCHAR(50) PRIMARY KEY, city VARCHAR(50) );
Query OK, 0 rows affected (0.018 sec)

MariaDB [employee]> CREATE TABLE manages ( employee_name VARCHAR(50), manager_name VARCHAR(50), PRIMARY KEY (employee_name, manager_name), FOREIGN KEY (employee_name) REFERE
NCES employee(employee_name), FOREIGN KEY (manager_name) REFERENCES employee(employee_name) );
Query OK, 0 rows affected (0.018 sec)
```

**Inserting Data :**

➔ **Employee Table :**

= INSERT INTO employee VALUES ('John Smith', '123 Main St', 'Newtown'), ('Jane Doe', '123 Main St', 'Newtown'), ('Michael Johnson', '789 Oak St', 'Chicago'), ('Emily Williams', '789 Oak St', 'Chicago'), ('Robert Brown', '555 Maple Ave', 'Houston'), ('Jennifer Davis', '555 Maple Ave', 'Houston'), ('William Lee', '222 Walnut St', 'Miami'), ('Jessica Wilson', '222 Walnut St', 'Miami'), ('David Taylor', '777 Spruce St', 'Boston'), ('Sarah Anderson', '777 Spruce St', 'Boston');

➔ **Works Table :**

= INSERT INTO works VALUES ('John Smith', 'First Bank Corporation', 12000), ('Jane Doe', 'First Bank Corporation', 11000), ('Michael Johnson', 'Second Bank Corporation', 15000), ('Emily Williams', 'Small Bank Corporation', 13000), ('Robert Brown', 'First Bank Corporation', 12500), ('Jennifer Davis', 'Second Bank Corporation', 14000), ('William Lee', 'Small Bank Corporation', 11500), ('Jessica Wilson', 'First Bank Corporation', 10500), ('David Taylor', 'Second Bank Corporation', 13500), ('Sarah Anderson', 'Small Bank Corporation', 12200);

➔ **Company Table :**

**=** INSERT INTO company VALUES ('First Bank Corporation', 'Newtown'), ('Second Bank Corporation', 'Los Angeles'), ('Small Bank Corporation', 'Chicago');

➔ **Manages Table :**

= INSERT INTO manages VALUES ('John Smith', 'Jane Doe'), ('Michael Johnson', 'John Smith'), ('Emily Williams', 'Jane Doe'), ('Jennifer Davis', 'John Smith'), ('William Lee', 'Michael Johnson');

**OUTPUT :**

```
MariaDB [employee]> INSERT INTO employee VALUES ('John Smith', '123 Main St', 'Newtown'), ('Jane Doe', '123 Main St', 'Newtown'), ('Michael Johnson', '789 Oak St', 'Chicago'), ('Emily Williams', '789 Oak St', 'Chicago'), ('Robert Brown', '555 Maple Ave', 'Houston'), ('Jennifer Davis', '555 Maple Ave', 'Houston'), ('William Lee', '222 Walnut St', 'Miami'), ('Jessica Wilson', '222 Walnut St', 'Miami'), ('David Taylor', '777 Spruce St', 'Boston'), ('Sarah Anderson', '777 Spruce St', 'Boston');
Query OK, 10 rows affected (0.017 sec)
Records: 10  Duplicates: 0  Warnings: 0

MariaDB [employee]> INSERT INTO works VALUES ('John Smith', 'First Bank Corporation', 12000), ('Jane Doe', 'First Bank Corporation', 11000), ('Michael Johnson', 'Second Bank Corporation', 15000), ('Emily Williams', 'Small Bank Corporation', 13000), ('Robert Brown', 'First Bank Corporation', 12500), ('Jennifer Davis', 'Second Bank Corporation', 14000), ('William Lee', 'Small Bank Corporation', 11500), ('Jessica Wilson', 'First Bank Corporation', 10500), ('David Taylor', 'Second Bank Corporation', 13500), ('Sarah Anderson', 'Small Bank Corporation', 12200);
Query OK, 10 rows affected (0.011 sec)
Records: 10  Duplicates: 0  Warnings: 0

MariaDB [employee]> INSERT INTO company VALUES ('First Bank Corporation', 'NewTown'), ('Second Bank Corporation', 'Los Angeles'), ('Small Bank Corporation', 'Chicago');
Query OK, 3 rows affected (0.037 sec)
Records: 3  Duplicates: 0  Warnings: 0

Database changed
MariaDB [employee]> INSERT INTO manages VALUES ('John Smith', 'Jane Doe'), ('Michael Johnson', 'John Smith'), ('Emily Williams', 'Jane Doe'), ('Jennifer Davis', 'John Smith'), ('William Lee', 'Michael Johnson');
Query OK, 5 rows affected (0.011 sec)
Records: 5  Duplicates: 0  Warnings: 0
```

```
MariaDB [employee]> SELECT * FROM employee; SELECT * FROM works; SELECT * FROM company; SELECT * FROM manages;
+-----------------+---------------+---------+
| employee_name   | street        | city    |
+-----------------+---------------+---------+
| David Taylor    | 777 Spruce St | Boston  |
| Emily Williams  | 789 Oak St    | Chicago |
| Jane Doe        | 123 Main St   | Newtown |
| Jennifer Davis  | 555 Maple Ave | Houston |
| Jessica Wilson  | 222 Walnut St | Miami   |
| John Smith      | 123 Main St   | Newtown |
| Michael Johnson | 789 Oak St    | Chicago |
| Robert Brown    | 555 Maple Ave | Houston |
| Sarah Anderson  | 777 Spruce St | Boston  |
| William Lee     | 222 Walnut St | Miami   |
+-----------------+---------------+---------+
10 rows in set (0.000 sec)

+-----------------+------------------------+----------+
| employee_name   | company_name           | salary   |
+-----------------+------------------------+----------+
| David Taylor    | Second Bank Corporation | 13500.00 |
| Emily Williams  | Small Bank Corporation  | 13000.00 |
| Jane Doe        | First Bank Corporation  | 11000.00 |
| Jennifer Davis  | Second Bank Corporation | 14000.00 |
| Jessica Wilson  | First Bank Corporation  | 10500.00 |
| John Smith      | First Bank Corporation  | 12000.00 |
| Michael Johnson | Second Bank Corporation | 15000.00 |
| Robert Brown    | First Bank Corporation  | 12500.00 |
| Sarah Anderson  | Small Bank Corporation  | 12200.00 |
| William Lee     | Small Bank Corporation  | 11500.00 |
+-----------------+------------------------+----------+
10 rows in set (0.000 sec)

+------------------------+-------------+
| company_name           | city        |
+------------------------+-------------+
| First Bank Corporation | NewTown     |
| Second Bank Corporation | Los Angeles |
| Small Bank Corporation | Chicago     |
+------------------------+-------------+
3 rows in set (0.000 sec)

+-----------------+-----------------+
| employee_name   | manager_name    |
+-----------------+-----------------+
| Emily Williams  | Jane Doe        |
| Jennifer Davis  | John Smith      |
| John Smith      | Jane Doe        |
| Michael Johnson | John Smith      |
| William Lee     | Michael Johnson |
+-----------------+-----------------+
5 rows in set (0.000 sec)
```

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

= SELECT employee_name FROM works WHERE company_name = 'First Bank Corporation';

**OUTPUT :**

```
MariaDB [employee]> SELECT employee_name FROM works WHERE company_name = 'First Bank Corporation';
+-----------------+
| employee_name   |
+-----------------+
| Jane Doe        |
| Jessica Wilson  |
| John Smith      |
| Robert Brown    |
+-----------------+
4 rows in set (0.000 sec)
```

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

= SELECT employee.employee_name, employee.city FROM employee, works WHERE employee.employee_name = works.employee_name AND works.company_name = 'First Bank Corporation';

**OUTPUT :**

```
MariaDB [employee]> SELECT employee.employee_name, employee.city FROM employee, works WHERE employee.employee_name = works.employee_name AND works.company_name = 'First Bank Corporation';
+---------------+---------+
| employee_name | city    |
+---------------+---------+
| Jane Doe      | Newtown |
| Jessica Wilson | Miami   |
| John Smith    | Newtown |
| Robert Brown  | Houston |
+---------------+---------+
4 rows 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 $10,000.

= SELECT employee.employee_name, employee.street, employee.city FROM employee, works WHERE employee.employee_name = works.employee_name AND works.company_name = 'First Bank Corporation' AND works.salary > 10000;

**OUTPUT :**

```
MariaDB [employee]> SELECT employee.employee_name, employee.street, employee.city FROM employee, works WHERE employee.employee_name = works.employee_name AND works.company_name = 'First Bank
Corporation' AND works.salary > 10000;
+---------------+--------------+---------+
| employee_name | street       | city    |
+---------------+--------------+---------+
| Jane Doe      | 123 Main St  | Newtown |
| Jessica Wilson | 222 Walnut St | Miami   |
| John Smith    | 123 Main St  | Newtown |
| Robert Brown  | 555 Maple Ave | Houston |
+---------------+--------------+---------+
4 rows in set (0.011 sec)
```

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

= SELECT * FROM employee WHERE city IN ( SELECT city FROM company WHERE company_name IN ( SELECT company_name FROM works WHERE employee.employee_name = works.employee_name ) );

**OUTPUT :**

```
MariaDB [employee]> SELECT * FROM employee WHERE city IN ( SELECT city FROM company WHERE company_name IN ( SELECT company_name FROM works WHERE employee.employee_name = works.employee_name
 );
+---------------+-------------+---------+
| employee_name | street      | city    |
+---------------+-------------+---------+
| Jane Doe      | 123 Main St | Newtown |
| John Smith    | 123 Main St | Newtown |
| Emily Williams | 789 Oak St  | Chicago |
+---------------+-------------+---------+
```

**e.** Find all employees in the database who live in the same cities and on the same streets as do their managers.

= SELECT e.* FROM employee e, manages m, employee mng WHERE e.city = mng.city AND e.street = mng.street AND e.employee_name = m.employee_name AND m.manager_name = mng.employee_name;

**OUTPUT :**

```
MariaDB [employee]> SELECT e.* FROM employee e, manages m, employee mng WHERE e.city = mng.city AND e.street = mng.street AND e.employee_name = m.employee_name AND m.manager_name = mng.employ
ee_name;
+---------------+-------------+---------+
| employee_name | street      | city    |
+---------------+-------------+---------+
| John Smith    | 123 Main St | Newtown |
+---------------+-------------+---------+
1 row in set (0.001 sec)
```

**f.** Find all employees in the database who do not work for First Bank Corporation.

= SELECT * FROM employee WHERE employee_name NOT IN ( SELECT employee_name FROM works WHERE company_name = 'First Bank Corporation' );

**OUTPUT :**

```
MariaDB [employee]> SELECT * FROM employee WHERE employee_name NOT IN ( SELECT employee_name FROM works WHERE company_name = 'First Bank Corporation' );
+-----------------+---------------+---------+
| employee_name   | street        | city    |
+-----------------+---------------+---------+
| David Taylor    | 777 Spruce St | Boston  |
| Emily Williams  | 789 Oak St    | Chicago |
| Jennifer Davis  | 555 Maple Ave | Houston |
| Michael Johnson | 789 Oak St    | Chicago |
| Sarah Anderson  | 777 Spruce St | Boston  |
| William Lee     | 222 Walnut St | Miami   |
+-----------------+---------------+---------+
6 rows in set (0.002 sec)
```

**g.** Find all employees in the database who earn more than each employee of Small Bank Corporation.

= SELECT * FROM employee WHERE EXISTS ( SELECT * FROM works w1 WHERE employee.employee_name = w1.employee_name AND w1.salary > ALL ( SELECT

salary FROM works w2 WHERE w2.company_name = 'Small Bank Corporation' ) );

**OUTPUT :**

```
MariaDB [employee]> SELECT * FROM employee WHERE EXISTS ( SELECT * FROM works w1 WHERE employee.employee_name = w1.employee_name AND w1.salary > ALL ( SELECT salary FROM works w2 WHERE w2.com
pany_name = 'Small Bank Corporation' ) );
+-----------------+--------------+---------+
| employee_name   | street       | city    |
+-----------------+--------------+---------+
| David Taylor    | 777 Spruce St | Boston  |
| Jennifer Davis  | 555 Maple Ave | Houston |
| Michael Johnson | 789 Oak St    | Chicago |
+-----------------+--------------+---------+
3 rows in set (0.001 sec)
```

**h.** Find the company that has the smallest payroll.

= SELECT company_name FROM works GROUP BY company_name HAVING SUM(salary) = ( SELECT MIN(total_salary) FROM ( SELECT SUM(salary) AS total_salary FROM works GROUP BY company_name ) AS subquery );

**OUTPUT :**

```
MariaDB [employee]> SELECT company_name FROM works GROUP BY company_name HAVING SUM(salary) = ( SELECT MIN(total_salary) FROM ( SELECT SUM(salary) AS total_salary FROM works GROUP BY company_
name ) AS subquery );
+----------------------+
| company_name         |
+----------------------+
| Small Bank Corporation |
+----------------------+
1 row in set (0.001 sec)
```

**i.** Find those companies whose employees earn a higher salary, on average, than the average salary at First Bank Corporation.

= SELECT company_name FROM works GROUP BY company_name HAVING AVG(salary) > ( SELECT AVG(salary) FROM works WHERE company_name = 'First Bank Corporation' );

**OUTPUT :**

```
MariaDB [employee]> SELECT company_name FROM works GROUP BY company_name HAVING AVG(salary) > ( SELECT AVG(salary) FROM works WHERE company_name = 'First Bank Corporation' );
+----------------------+
| company_name         |
+----------------------+
| Second Bank Corporation |
| Small Bank Corporation  |
+----------------------+
2 rows in set (0.001 sec)
```

**j.** Modify the database so that Jones now lives in Newtown.

= UPDATE employee SET city = 'Newtown' WHERE employee_name = 'Jones';

**OUTPUT :**

```
MariaDB [employee]> UPDATE employee SET city = 'Newtown' WHERE employee_name = 'Jones';
Query OK, 0 rows affected (0.001 sec)
Rows matched: 0  Changed: 0  Warnings: 0

MariaDB [employee]> SELECT * FROM employee;
+-----------------+---------------+---------+
| employee_name   | street        | city    |
+-----------------+---------------+---------+
| David Taylor    | 777 Spruce St | Boston  |
| Emily Williams  | 789 Oak St    | Chicago |
| Jane Doe        | 123 Main St   | Newtown |
| Jennifer Davis  | 555 Maple Ave | Houston |
| Jessica Wilson  | 222 Walnut St | Miami   |
| John Smith      | 123 Main St   | Newtown |
| Michael Johnson | 789 Oak St    | Chicago |
| Robert Brown    | 555 Maple Ave | Houston |
| Sarah Anderson  | 777 Spruce St | Boston  |
| William Lee     | 222 Walnut St | Miami   |
+-----------------+---------------+---------+
10 rows in set (0.001 sec)
```

**k.** Give all employees of First Bank Corporation a 10 percent raise.

= UPDATE works SET salary = salary * 1.1 WHERE company_name = 'First Bank Corporation';

**OUTPUT :**

```
MariaDB [employee]> UPDATE works SET salary = salary * 1.1 WHERE company_name = 'First Bank Corporation';
Query OK, 4 rows affected (0.011 sec)
Rows matched: 4  Changed: 4  Warnings: 0

MariaDB [employee]> SLECT * FROM works;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near 'SLECT * FROM works' at line 1
MariaDB [employee]> SELECT * FROM works;
+-----------------+------------------------+----------+
| employee_name   | company_name           | salary   |
+-----------------+------------------------+----------+
| David Taylor    | Second Bank Corporation | 13500.00 |
| Emily Williams  | Small Bank Corporation  | 13000.00 |
| Jane Doe        | First Bank Corporation  | 12100.00 |
| Jennifer Davis  | Second Bank Corporation | 14000.00 |
| Jessica Wilson  | First Bank Corporation  | 11550.00 |
| John Smith      | First Bank Corporation  | 13200.00 |
| Michael Johnson | Second Bank Corporation | 15000.00 |
| Robert Brown    | First Bank Corporation  | 13750.00 |
| Sarah Anderson  | Small Bank Corporation  | 12200.00 |
| William Lee     | Small Bank Corporation  | 11500.00 |
+-----------------+------------------------+----------+
10 rows in set (0.001 sec)
```

**l.** Give all managers of First Bank Corporation a 10 percent raise.

= UPDATE works SET salary = salary * 1.1 WHERE employee_name IN ( SELECT employee_name FROM manages WHERE manager_name IN ( SELECT employee_name FROM works WHERE company_name = 'First Bank Corporation' ) );

**OUTPUT :**

```
MariaDB [employee]> UPDATE works SET salary = salary * 1.1 WHERE employee_name IN ( SELECT employee_name FROM manages WHERE manager_name IN ( SELECT employee_name FROM works WHERE company_nam
e = 'First Bank Corporation' ) );
Query OK, 4 rows affected (0.012 sec)
Rows matched: 4  Changed: 4  Warnings: 0

MariaDB [employee]> SELECT * FROM works;
+-----------------+------------------------+----------+
| employee_name   | company_name           | salary   |
+-----------------+------------------------+----------+
| David Taylor    | Second Bank Corporation | 13500.00 |
| Emily Williams  | Small Bank Corporation  | 14300.00 |
| Jane Doe        | First Bank Corporation  | 12100.00 |
| Jennifer Davis  | Second Bank Corporation | 15400.00 |
| Jessica Wilson  | First Bank Corporation  | 11550.00 |
| John Smith      | First Bank Corporation  | 14520.00 |
| Michael Johnson | Second Bank Corporation | 16500.00 |
| Robert Brown    | First Bank Corporation  | 13750.00 |
| Sarah Anderson  | Small Bank Corporation  | 12200.00 |
| William Lee     | Small Bank Corporation  | 11500.00 |
+-----------------+------------------------+----------+
10 rows in set (0.000 sec)
```

**m.** Delete all tuples in the works relation for employees of Small Bank Corporation.

= DELETE FROM works WHERE company_name = 'Small Bank Corporation';

**OUTPUT :**

```
MariaDB [employee]> DELETE FROM works WHERE company_name = 'Small Bank Corporation';
Query OK, 3 rows affected (0.023 sec)

MariaDB [employee]> SELECT * FROM works;
+-----------------+------------------------+----------+
| employee_name   | company_name           | salary   |
+-----------------+------------------------+----------+
| David Taylor    | Second Bank Corporation | 13500.00 |
| Jane Doe        | First Bank Corporation  | 12100.00 |
| Jennifer Davis  | Second Bank Corporation | 15400.00 |
| Jessica Wilson  | First Bank Corporation  | 11550.00 |
| John Smith      | First Bank Corporation  | 14520.00 |
| Michael Johnson | Second Bank Corporation | 16500.00 |
| Robert Brown    | First Bank Corporation  | 13750.00 |
+-----------------+------------------------+----------+
7 rows in set (0.000 sec)
```