# 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: 4<sup>th</sup>

**Submitted To:** 

**Name :** Amit K. Shrivastava

**Department of Software** 

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

```
person (<u>driver-id</u>, name, address)
car (<u>license</u>, model, year)
accident (<u>report-number</u>, date, location)
owns (<u>driver-id</u>, license)
participated (<u>driver-id</u>, <u>car</u>, <u>report-number</u>, damage-amount)
fig1: Insurance database
```

#### Answer:

## → Creating a Database :

= create database insurance:

```
[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
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));

## **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');

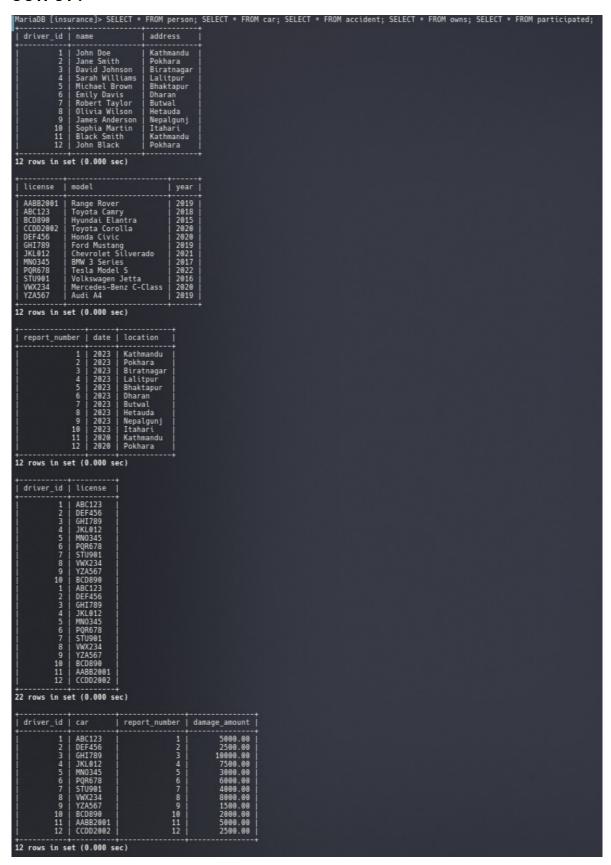
## → Car Table :

= 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');



- **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));

- **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:**

- 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' );

```
MariaDB [Insurance]> DELETE FROM owns WHERE license IN ( SELECT license FROM owns WHERE model = 'Range Rover' );

(Query ON, O 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' );

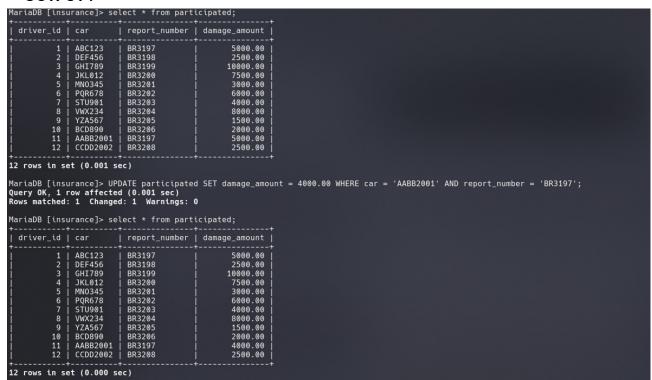
Empty set (0.001 sec)

MariaDB [Insurance]> SELECT * FROM owns 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)

| driver_id | license | |
| 1 | ABC123 |
| 1 | BEC1999 |
| 1 | 2 | CC002692 |
| 2 | 2 | CM1789 |
| 4 | JXL4012 |
| 5 | NNO.345 |
| 6 | PORGOTS |
| 7 | STUMEL |
| 8 | WAC234 |
| 9 | YAX667 |
| 1 | Tows 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';



# 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 (<u>employee-name</u>, street, city)
works (<u>employee-name</u>, company-name, salary)
company (<u>company-name</u>, city)
manages (<u>employee-name</u>, manager-name)
```

Figure 2. Employee database.

#### Answer:

# → Creating a Database :

= create database employee;

#### **OUTPUT:**

# → 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, O 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, O rows affected (0.005 sec)

MariaDB [employee]> CREATE TABLE company ( company_name VARCHAR(50) PRIMARY KEY, city VARCHAR(50));
Query OK, O 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), FOREIGN KEY (empl
```

## **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');

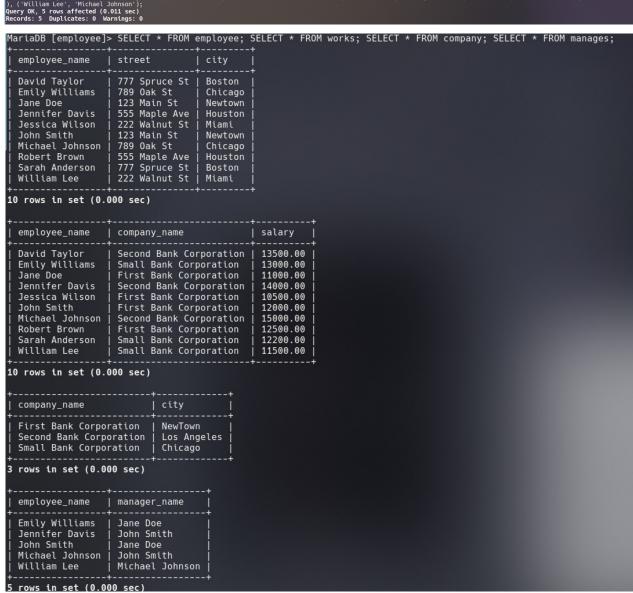
## → Manages Table :

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

```
MariaDB [employee]> INSERT INTO employee VALUES ('John Smith', '123 Main St', 'NewTown'), ('Jane Doe', '123 Main St', 'Newtown'), ('Michael Johnson', '789 Oak St', 'Chicago'), ('Robert Brown', '555 Maple Ave', 'Houston'), ('Jennifer Davis', '555 Maple Ave', 'Houston'), ('William Lee', '222 Walnut St', 'Miami'), ('David Taylor', '777 Spruce St', 'Boston'), ('Sarah Anderson', '777 Spruce St', 'Boston');

Query OK, 10 rows affected (0.017 sec)

MariaDB [employee]> INSERT INTO works VALUES ('John Smith', 'First Bank Corporation', 12000), ('Jene 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', 12500), ('Jessica Wilson', 'First Bank Corporation', 10500), ('David Taylor', 'Second Bank Corporation', 13500), ('Sarah Anderson', 'Small Bank Corporation', 12500), ('Sarah Anderson', 'Sarah Anderson', 'Small Bank Corporation', 'Small Bank Corporatio
```



- a. Find the names of all employees who work for First Bank Corporation.
  - = SELECT employee\_name FROM works WHERE company\_name = 'First Bank Corporation';

- **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:**

- **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;

- **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 ) );

- **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:**



- 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' );



- **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:**

- **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:**

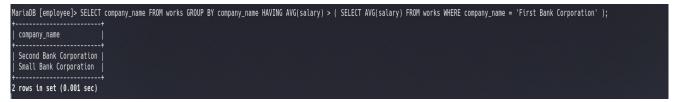
```
MariaDB [employee]> SELECT company_name FROM works GROUP BY company_name HAVING SUM(salary) = ( SELECT MIN(total_salary) FROM ( SELECT SUM(salary) AS total_salary FROM works GROUP BY company_name | com
```

- **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:**



- **j.** Modify the database so that Jones now lives in Newtown.
  - = UPDATE employee SET city = 'Newtown' WHERE employee\_name = 'Jones';

- 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';

- **I.** 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:**



- m. Delete all tuples in the works relation for employees of Small Bank Corporation.
  - = DELETE FROM works WHERE company\_name = 'Small Bank Corporation';

```
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 |
| Jennifer Davis | Second Bank Corporation | 15500.00 |
| John Smith | First Bank Corporation | 14520.00 |
| Michael Johnson | Second Bank Corporation | 14520.00 |
| Robert Brown | First Bank Corporation | 13750.00 |
| Robert Brown | First Bank Corporation | 13750.00 |
```