Mysql-7 Consider following Relation Employee(emp_id,employee_name,street,city) Works(employee_name,company_name,salary) Company(company_name,city) Manages(employee_name,manager_name) Create above tables with appropriate constraints like primary key, foreign key, not null etc.

1. Change the city of employee working with InfoSys to 'Bangalore'

2. Find the names of all employees who earn more than the average salary of all employees of their company. Assume that all people work for at most one company.

3. Find the names, street address, and cities of residence for all employees who work for 'TechM' and earn more than $10,000.

4. Change name of table Manages to Management.

5. Create Simple and Unique index on employee table.

6. Display index Information

```
-- Employee Table
CREATE TABLE Employee (
    emp_id INT PRIMARY KEY,
    employee_name VARCHAR(50) NOT NULL UNIQUE,
    street VARCHAR(50) NOT NULL,
    city VARCHAR(50) NOT NULL
);

-- Company Table
CREATE TABLE Company (
    company_name VARCHAR(50) PRIMARY KEY,
    city VARCHAR(50) NOT NULL
);

-- Works Table
CREATE TABLE Works (
    employee_name VARCHAR(50),
    company_name VARCHAR(50),
    salary DECIMAL(10,2) CHECK (salary >= 0),
    PRIMARY KEY (employee_name, company_name),
    FOREIGN KEY (employee_name) REFERENCES Employee(employee_name) ON
DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (company_name) REFERENCES Company(company_name) ON
DELETE CASCADE ON UPDATE CASCADE
);

-- 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) ON
DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (manager_name) REFERENCES Employee(employee_name) ON
DELETE CASCADE ON UPDATE CASCADE
);
```

```
-- Employee Table
INSERT INTO Employee VALUES
(101, 'Ramesh', 'FC Road', 'Pune'),
(102, 'Suresh', 'MG Road', 'Mumbai'),
(103, 'Kiran', 'JM Road', 'Bangalore'),
(104, 'Anita', 'Park Street', 'Kolkata'),
(105, 'Priya', 'Brigade Road', 'Bangalore');

-- Company Table
INSERT INTO Company VALUES
('TCS', 'Mumbai'),
('Infosys', 'Pune'),
('TechM', 'Hyderabad'),
('Wipro', 'Bangalore');

-- Works Table
INSERT INTO Works VALUES
('Ramesh', 'TCS', 12000),
('Suresh', 'Infosys', 9500),
('Kiran', 'TechM', 15000),
('Anita', 'Wipro', 8000),
('Priya', 'TechM', 11000);

-- Manages Table
INSERT INTO Manages VALUES
('Ramesh', 'Suresh'),
('Kiran', 'Priya'),
('Anita', 'Ramesh');
```

Step 7: Test the Queries

✅ **Query 1 – Change the city of employee working with ‘InfoSys’ to ‘Bangalore’**

```
UPDATE Employee e
JOIN Works w ON e.employee_name = w.employee_name
SET e.city = 'Bangalore'
WHERE w.company_name = 'Infosys';
```

```
[SELECT e.employee_name, e.city, w.company_name
FROM Employee e
JOIN Works w ON e.employee_name = w.employee_name
WHERE w.company_name = 'Infosys';]to view specific changes
```

Query 2 – Find the names of all employees who earn more than the average salary of their company

```
SELECT w.employee_name, w.company_name, w.salary
FROM Works w
JOIN (
    SELECT company_name, AVG(salary) AS avg_salary
    FROM Works
    GROUP BY company_name
) AS avg_table
ON w.company_name = avg_table.company_name
WHERE w.salary > avg_table.avg_salary;
```

Query 3 – Find names, street, and city of employees who work for 'TechM' and earn more than $10,000

```
SELECT e.employee_name, e.street, e.city
FROM Employee e
JOIN Works w ON e.employee_name = w.employee_name
WHERE w.company_name = 'TechM' AND w.salary > 10000;
```

Query 4 – Change the name of table Manages to Management

```
RENAME TABLE Manages TO Management;
```

Query 5 – Create Simple and Unique Index on Employee table

```
-- Simple Index on city
CREATE INDEX idx_city ON Employee(city);

-- Unique Index on employee_name
CREATE UNIQUE INDEX idx_empname ON Employee(employee_name);
```

Query 6-Display Index Information

```
SHOW INDEX FROM Employee;
```