DATA ENGINEERING

Name: Ramireddy Preethi

Batch: Python Batch 2

DAY4:

```
-- Drop the existing 'org' database if it exists
DROP DATABASE org;


CREATE DATABASE org;


-- Switch to the 'org' database
USE org;


-- Create the 'employees' table with specified columns
CREATE TABLE employees (
    id INT NOT NULL AUTO_INCREMENT,  -- Unique identifier for each employee
    first_name VARCHAR(255) NOT NULL,  -- Employee's first name
    last_name VARCHAR(255),  -- Employee's last name
    dept VARCHAR(255),  -- Department in which the employee works
    email VARCHAR(255) NOT NULL,  -- Employee's email address
    phone VARCHAR(50),  -- Employee's phone number
    hire_date DATE,  -- Date when the employee was hired
    salary DECIMAL(10, 2),  -- Employee's salary
    PRIMARY KEY (id)  -- Set 'id' as the primary key
);


INSERT INTO employees (first_name, last_name, dept, email, phone, hire_date, salary) VALUES
('Nancy', 'Davolio', 'Sales Representative', 'nancy.davolio1@example.com', '555-0123', '2023-01-15', 5000.00),
('Andrew', 'Fuller', 'Vice President, Sales', 'andrew.fuller@example.com', '555-0456', '2022-03-10', 8000.00),
```

('Janet', 'Leverling', 'Sales Representative', 'janet.leverling@example.com', '555-0789', '2023-05-20', 5000.00),

('Margaret', 'Peacock', 'Sales Representative', 'margaret.peacock@example.com', '555-1011', '2023-02-25', 5000.00),

('Steven', 'Buchanan', 'Sales Manager', 'steven.buchanan@example.com', '555-1213', '2022-12-01', 7000.00),

('Michael', 'Suyama', 'Sales Representative', 'michael.suyama@example.com', '555-1415', '2023-04-11', 5000.00),

('Robert', 'King', 'Sales Representative', 'robert.king@example.com', '555-1617', '2023-07-05', 5000.00),

('Laura', 'Callahan', 'Inside Sales Coordinator', 'laura.callahan@example.com', '555-1819', '2023-08-15', 6000.00),

('Anne', 'Dodsworth', 'Sales Representative', 'anne.dodsworth@example.com', '555-2021', '2023-06-30', 5000.00);

```
CREATE TABLE customers (
   id INT NOT NULL AUTO_INCREMENT,
   employee_id INT,
   company_name VARCHAR(255) NOT NULL,
   address VARCHAR(255),
   city VARCHAR(255),
   region VARCHAR(255),
   country VARCHAR(255),
   phone VARCHAR(255),
   PRIMARY KEY (id),
   FOREIGN KEY (employee_id) REFERENCES employees(id)  ON DELETE SET NULL
);
INSERT INTO customers (employee_id, company_name, address, city, region, country, phone)
VALUES(1, 'Rancho Grande', 'Av. del Libertador 900', 'Buenos Aires', '', 'Argentina', '(1) 123-5555'),
(2, 'Alfreds Futterkiste','Obere Str. 57','Berlin', '','Germany','030-0074321'),
(2, 'Bottom-Dollar Markets','23 Tsawassen Blvd.','Tsawassen','BC','Canada','(604) 555-4729'),
```

(3, 'Drachenblut Delikatessen','Walserweg 21','Aachen', '','Germany','0241-039123'),

(4, 'Du monde entier','67, rue des Cinquante Otages','Nantes', '','France','40.67.88.88'),

(1, 'Familia Arquibaldo','Rua Orós, 92','São Paulo','SP','Brazil','(11) 555-9857'),

(4, 'Frankenversand','Berliner Platz 43','München', '','Germany','089-0877310'),

(5, 'France restauration','54, rue Royale','Nantes', '','France','40.32.21.21'),

(5, 'Great Lakes Food Market','2732 Baker Blvd.','Eugene','OR','USA','(503) 555-7555'),

(5, 'Königlich Essen','Maubelstr. 90','Brandenburg', '','Germany','0555-09876'),

(6, 'La corne d''abondance','67, avenue de l''Europe','Versailles', '','France','30.59.84.10'),

(7, 'Laughing Bacchus Wine Cellars','1900 Oak St.','Vancouver','BC','Canada','(604) 555-3392'),

(6, 'Lazy K Kountry Store','12 Orchestra Terrace','Walla Walla','WA','USA','(509) 555-7969'),

(2, 'Let''s Stop N Shop','87 Polk St. Suite 5','San Francisco','CA','USA','(415) 555-5938'),

(8, 'Lonesome Pine Restaurant','89 Chiaroscuro Rd.','Portland','OR','USA','(503) 555-9573'),

(7, 'Old World Delicatessen','2743 Bering St.','Anchorage','AK','USA','(907) 555-7584'),

(6, 'Que Delícia','Rua da Panificadora, 12','Rio de Janeiro','RJ','Brazil','(21) 555-4252'),

(5, 'Queen Cozinha','Alameda dos Canàrios, 891','São Paulo','SP','Brazil','(11) 555-1189'),

(4, 'QUICK-Stop','Taucherstraße 10','Cunewalde', '','Germany','0372-035188'),

(4, 'Spécialités du monde','25, rue Lauriston','Paris', '','France','(1) 47.55.60.10'),

(3, 'Split Rail Beer & Ale','P.O. Box 555','Lander','WY','USA','(307) 555-4680'),

(7, 'The Cracker Box','55 Grizzly Peak Rd.','Butte','MT','USA','(406) 555-5834'),

(7, 'Die Wandernde Kuh','Adenauerallee 900','Stuttgart', '','Germany','0711-020361'),

(3, 'White Clover Markets','305 - 14th Ave. S. Suite 3B','Seattle','WA','USA','(206) 555-4112'),

(1, 'Wilman Kala','Keskuskatu 45','Helsinki', '','Finland','90-224 8858'),

(2, 'Wolski Zajazd','ul. Filtrowa 68','Warszawa', '','Poland','(26) 642-7012'),

(8, 'Tortuga Restaurante','Avda. Azteca 123','México D.F.', '','Mexico','(5) 555-2933'),

(2, 'Tradição Hipermercados','Av. Inês de Castro, 414','São Paulo','SP','Brazil','(11) 555-2167'),

(1, 'Trail''s Head Gourmet Provisioners','722 DaVinci Blvd.','Kirkland','WA','USA','(206) 555-8257'),

(1, 'Vaffeljernet','Smagsløget 45','Århus', '','Denmark','86 21 32 43');

-- Use the 'org' database for subsequent operations

USE org;

```
-- Equi join: Selecting employees and their associated customers
SELECT e.first_name, e.last_name, c.company_name
FROM employees e
JOIN customers c ON e.id = c.employee_id;

-- Self-join: Finding pairs of employees within the same department
SELECT e1.first_name AS employee1, e2.first_name AS employee2, e1.dept
FROM employees e1
JOIN employees e2 ON e1.dept = e2.dept AND e1.id < e2.id;

-- Counting the number of customers for each employee,
-- filtering to show only employees with more than 2 customers
SELECT e.first_name, e.last_name, COUNT(c.id) AS customer_count
FROM employees e
JOIN customers c ON e.id = c.employee_id
GROUP BY e.id
HAVING customer_count > 2;

-- Grouping customers by department, returning NULL for first_name
SELECT e.dept, NULL AS first_name, COUNT(c.id) AS customer_count
FROM employees e
LEFT JOIN customers c ON e.id = c.employee_id
GROUP BY e.dept

UNION ALL

-- Grouping customers by both department and first_name
SELECT e.dept, e.first_name, COUNT(c.id) AS customer_count
FROM employees e
LEFT JOIN customers c ON e.id = c.employee_id
GROUP BY e.dept, e.first_name;
```

```
-- Selecting employees with the highest salary in their respective departments
SELECT first_name, last_name, dept, salary
FROM employees e1
WHERE salary = (SELECT MAX(salary) FROM employees AS e2 WHERE e2.dept = e1.dept);

-- Checking if there are customers associated with each employee
SELECT first_name, last_name
FROM employees e
WHERE EXISTS (SELECT 1 FROM customers c WHERE c.employee_id = e.id);

-- Selecting employees whose salary is greater than any salary in the Sales Representative department
SELECT first_name, last_name, salary, dept
FROM employees
WHERE salary > ANY (SELECT salary FROM employees WHERE dept = 'Sales Representative');

-- Updating the salary of an employee named Nancy to 6000
UPDATE employees SET salary = 6000 WHERE first_name = 'Nancy';

-- Selecting employees whose salary is greater than all salaries in the Sales Representative department
SELECT first_name, last_name, salary, dept
FROM employees
WHERE salary > ALL (SELECT salary FROM employees WHERE dept = 'Sales Representative');

-- Selecting employees whose salary is greater than the average salary in their department
SELECT first_name, last_name, salary, dept
FROM employees e
WHERE salary > (SELECT AVG(salary) FROM employees WHERE dept = e.dept);

-- Combining results of employees and customers using LEFT and RIGHT joins to include all records
```

```
SELECT e.first_name, e.last_name, c.company_name
FROM employees e
LEFT JOIN customers c ON e.id = c.employee_id

UNION

SELECT e.first_name, e.last_name, c.company_name
FROM employees e
RIGHT JOIN customers c ON e.id = c.employee_id;

-- Show all databases in the MySQL server
SHOW DATABASES;

-- Use the 'org' database again
USE org;

-- Show all tables in the 'org' database
SHOW TABLES;

-- Describe the structure of the 'employees' table
DESCRIBE employees;

-- Add a new column 'commission' to the 'employees' table
ALTER TABLE employees ADD col1 DECIMAL(5, 2);

-- Modify the 'phone' column to increase its length
ALTER TABLE employees MODIFY phone VARCHAR(100);

-- Define a function to get average salary by department
DELIMITER //
CREATE FUNCTION GetAvgSalaryByDept(department VARCHAR(40))
```

```
RETURNS DECIMAL(10, 2)
DETERMINISTIC
BEGIN
    DECLARE avg_salary DECIMAL(10, 2);

    -- Calculate the average salary for the specified department
    SELECT AVG(salary) INTO avg_salary
    FROM employees
    WHERE dept = department;

    RETURN avg_salary;
END //
DELIMITER ;

-- Call the function to get average salary for 'Sales Representative' department
SELECT GetAvgSalaryByDept('Sales Representative') AS avg_salary;

-- Define a stored procedure to get employees by department
DELIMITER $$
CREATE PROCEDURE GetEmployeesByDept(IN dept_name VARCHAR(50))
BEGIN
    -- Select employees from the specified department
    SELECT first_name, last_name, salary, dept
    FROM employees
    WHERE dept = dept_name;
END$$
DELIMITER ;

-- Call the procedure to get employees from the 'Sales Representative' department
CALL GetEmployeesByDept('Sales Representative');
```

```
-- Inserting multiple records into the employees table
INSERT INTO employees (first_name, last_name, salary, hire_date, dept, email, phone) VALUES
('John', 'Doe ', 55000, '2021-05-01', 'Sales Representative', 'johndoe@@example.com', '12345678'),
('Mary', ' Smith', 62000, '2020-02-15', 'Sales Representative', 'mary.smith@example..com', '12345678'),
(' Williams', NULL, 70000, '2021-12-10', 'Marketing', 'williams@@example.com', '12345678'),
('James', ' Brown', 58000, '2022-04-23', 'Sales', 'james.brown@e.example.com', '12345678'),
('Anna', 'Taylor', 45000, NULL, 'Marketing', 'anna.taylor@example.com', NULL);

-- Select all records from the employees table
SELECT * FROM employees;

-- Trim whitespace from first and last names of employees
UPDATE employees SET first_name = TRIM(first_name), last_name = TRIM(last_name);

-- Fix email format by replacing double '@' with a single '@'
UPDATE employees SET email = REPLACE(email, '@@', '@') WHERE email LIKE '%@@%';

-- Update hire date to the current date for any records where it's null
UPDATE employees SET hire_date = CURRENT_DATE WHERE hire_date IS NULL;

-- Delete records where last name or phone is null
DELETE FROM employees WHERE last_name IS NULL OR phone IS NULL;

-- Select all employees again to check updates
SELECT * FROM employees;

-- Select employees whose first name starts with 'm'
SELECT * FROM employees WHERE first_name REGEXP "^m";

-- Select employees whose first name starts with 'm' or 'a'
```

```
SELECT * FROM employees WHERE first_name REGEXP "^[ma]";

-- Select employees whose first name ends with 't'
SELECT * FROM employees WHERE first_name REGEXP "t$";

-- Select employees whose first name starts with 'm' and ends with 't'
SELECT * FROM employees WHERE first_name REGEXP "^m.*t$";

-- Inserting a record into the employees table
INSERT INTO employees (first_name, last_name, salary, hire_date, dept, email, phone) VALUES
('John', 'Doe ', 4000, '2021-05-01', 'Sales Representative', 'johndoe@@example.com', '12345678');

-- Select employees with a row number assigned based on department and salary
SELECT id, first_name, last_name, dept, salary,
    ROW_NUMBER() OVER (PARTITION BY dept ORDER BY salary DESC) AS rowNumber
FROM employees;

-- Calculate the rank of employees based on salary within each department
SELECT id, first_name, last_name, dept, salary,
    RANK() OVER (PARTITION BY dept ORDER BY salary DESC) AS salary_rank
FROM employees;

-- Calculate the dense rank of employees based on salary within each department
SELECT id, first_name, last_name, dept, salary,
    DENSE_RANK() OVER (PARTITION BY dept ORDER BY salary DESC) AS salary_rank
FROM employees;

-- Common Table Expression (CTE) to find employees hired after 2023
WITH recent_hires AS (
  SELECT id, first_name, last_name, hire_date FROM employees WHERE hire_date > '2023-01-01'
)
```

```
SELECT * FROM recent_hires;


-- Summarizing total salary by department with ROLLUP
SELECT dept, SUM(salary) AS total_salary
FROM employees
GROUP BY dept WITH ROLLUP;


-- Drop the 'employees' table (be cautious with this command!)
DROP TABLE employees;
```