Data Engineering Batch 1 **Date: 25/01/2024**

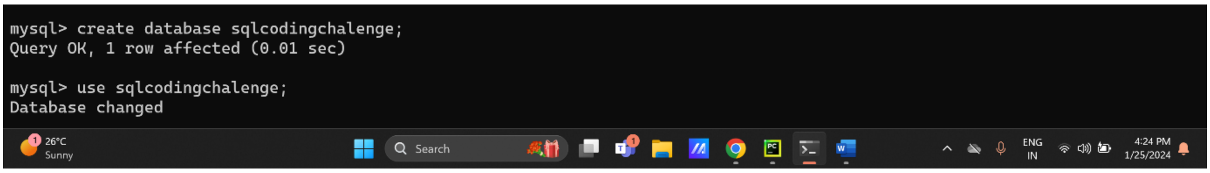
Name: Pradip Bochare

SQL Coding Challenge

❖ Execute OVER and PARTITION BY Clause in SQL Queries, creating subtotals &Total Aggregations using SQL Queries.

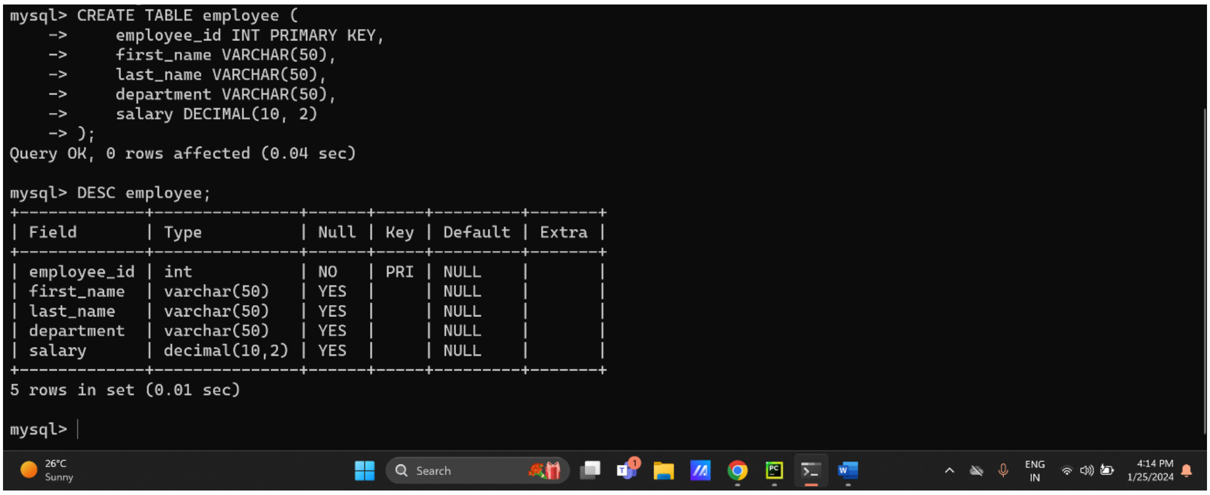
- Over and Partition By clauses in SQL are used in conjunction with window functions to perform calculations on a specific subset of data within a result set. This is useful for creating subtotals and total aggregations.

- Here we have created a database ‘sqlcodinchaleng’ to implement Over and Partition By clause



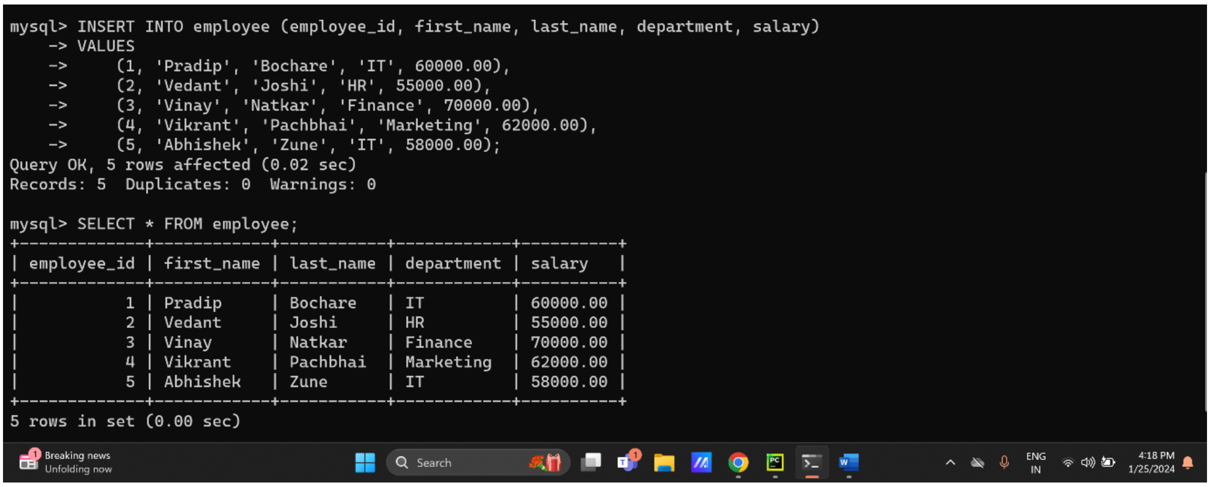
```
mysql> create database sqlcodingchalenge;
Query OK, 1 row affected (0.01 sec)

mysql> use sqlcodingchalenge;
Database changed
```

- Created Table ‘employee’ in database



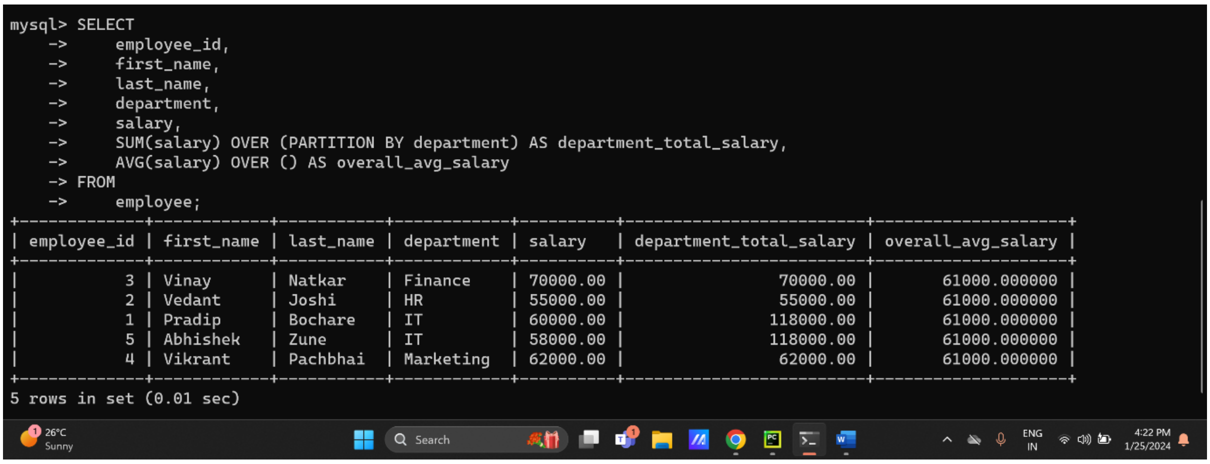
```
mysql> CREATE TABLE employee (
    ->     employee_id INT PRIMARY KEY,
    ->     first_name VARCHAR(50),
    ->     last_name VARCHAR(50),
    ->     department VARCHAR(50),
    ->     salary DECIMAL(10, 2)
    -> );
Query OK, 0 rows affected (0.04 sec)

mysql> DESC employee;
+-------------+---------------+------+-----+---------+-------+
| Field       | Type          | Null | Key | Default | Extra |
+-------------+---------------+------+-----+---------+-------+
| employee_id | int           | NO   | PRI | NULL    |       |
| first_name  | varchar(50)   | YES  |     | NULL    |       |
| last_name   | varchar(50)   | YES  |     | NULL    |       |
| department  | varchar(50)   | YES  |     | NULL    |       |
| salary      | decimal(10,2) | YES  |     | NULL    |       |
+-------------+---------------+------+-----+---------+-------+
5 rows in set (0.01 sec)

mysql>
```

- Inserted Sample data in ‘employee’ table



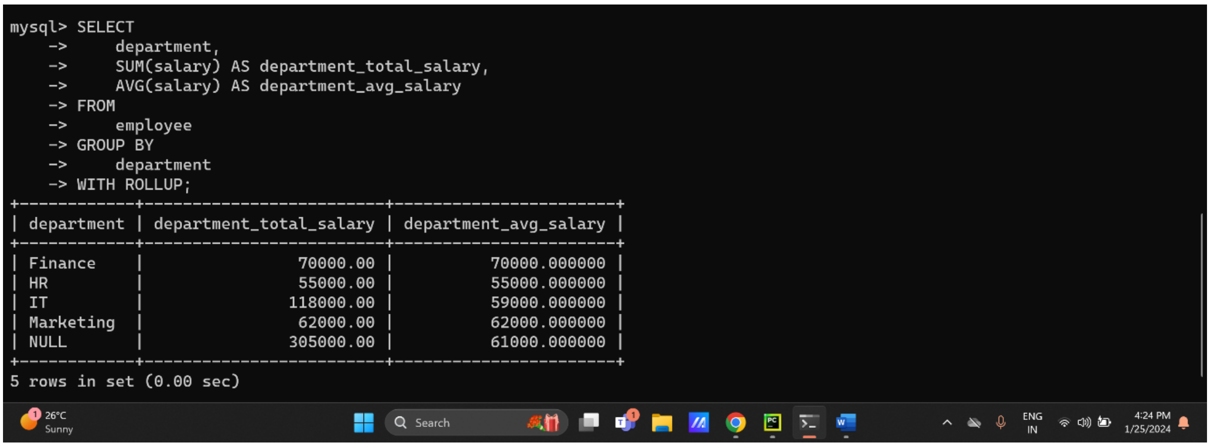
```
mysql> INSERT INTO employee (employee_id, first_name, last_name, department, salary)
    -> VALUES
    ->     (1, 'Pradip', 'Bochare', 'IT', 60000.00),
    ->     (2, 'Vedant', 'Joshi', 'HR', 55000.00),
    ->     (3, 'Vinay', 'Natkar', 'Finance', 70000.00),
    ->     (4, 'Vikrant', 'Pachbhai', 'Marketing', 62000.00),
    ->     (5, 'Abhishek', 'Zune', 'IT', 58000.00);
Query OK, 5 rows affected (0.02 sec)
Records: 5  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM employee;
+-------------+------------+-----------+------------+----------+
| employee_id | first_name | last_name | department | salary   |
+-------------+------------+-----------+------------+----------+
|           1 | Pradip     | Bochare   | IT         | 60000.00 |
|           2 | Vedant     | Joshi     | HR         | 55000.00 |
|           3 | Vinay      | Natkar    | Finance    | 70000.00 |
|           4 | Vikrant    | Pachbhai  | Marketing  | 62000.00 |
|           5 | Abhishek   | Zune      | IT         | 58000.00 |
+-------------+------------+-----------+------------+----------+
5 rows in set (0.00 sec)
```

- Calculate total salary for each department and overall average salary



```
mysql> SELECT
    ->     employee_id,
    ->     first_name,
    ->     last_name,
    ->     department,
    ->     salary,
    ->     SUM(salary) OVER (PARTITION BY department) AS department_total_salary,
    ->     AVG(salary) OVER () AS overall_avg_salary
    -> FROM
    ->     employee;
+-------------+------------+-----------+------------+----------+-------------------------+--------------------+
| employee_id | first_name | last_name | department | salary   | department_total_salary | overall_avg_salary |
+-------------+------------+-----------+------------+----------+-------------------------+--------------------+
|           3 | Vinay      | Natkar    | Finance    | 70000.00 |                70000.00 |       61000.000000 |
|           2 | Vedant     | Joshi     | HR         | 55000.00 |                55000.00 |       61000.000000 |
|           1 | Pradip     | Bochare   | IT         | 60000.00 |               118000.00 |       61000.000000 |
|           5 | Abhishek   | Zune      | IT         | 58000.00 |               118000.00 |       61000.000000 |
|           4 | Vikrant    | Pachbhai  | Marketing  | 62000.00 |                62000.00 |       61000.000000 |
+-------------+------------+-----------+------------+----------+-------------------------+--------------------+
5 rows in set (0.01 sec)
```

- Calculate Subtotals and total aggregations



```
mysql> SELECT
    ->     department,
    ->     SUM(salary) AS department_total_salary,
    ->     AVG(salary) AS department_avg_salary
    -> FROM
    ->     employee
    -> GROUP BY
    ->     department
    -> WITH ROLLUP;
+------------+-------------------------+-----------------------+
| department | department_total_salary | department_avg_salary |
+------------+-------------------------+-----------------------+
| Finance    |                70000.00 |          70000.000000 |
| HR         |                55000.00 |          55000.000000 |
| IT         |               118000.00 |          59000.000000 |
| Marketing  |                62000.00 |          62000.000000 |
| NULL       |               305000.00 |          61000.000000 |
+------------+-------------------------+-----------------------+
5 rows in set (0.00 sec)
```

- Queries used in Coding Challenge Q1

- Calculate total salary for each department and overall average salary

```
SELECT
    employee_id,
    first_name,
    last_name,
    department,
    salary,
    SUM(salary) OVER (PARTITION BY department) AS department_total_salary,
    AVG(salary) OVER () AS overall_avg_salary
FROM
    employee;
```

- Calculate Subtotals and total aggregations

```
SELECT
    department,
    SUM(salary) AS department_total_salary,
    AVG(salary) AS department_avg_salary
FROM
    employee
GROUP BY
    department
WITH ROLLUP;
```