ASSIGNMENT 4:

1. Download dataset from https://www.kaggle.com/datasets/krishujeniya/salary-prediction-of-data-professions?resource=download
2. Ingest the dataset from your local machine storage into postgresQL database
   Hint: use **copy command** in sql editor which will copy your csv file to postgres DB
   For ingesting csv you might also need to create table according to the column structure of your CSV file ahead of executing copy command

3. Once the table is populated please complete following queries:

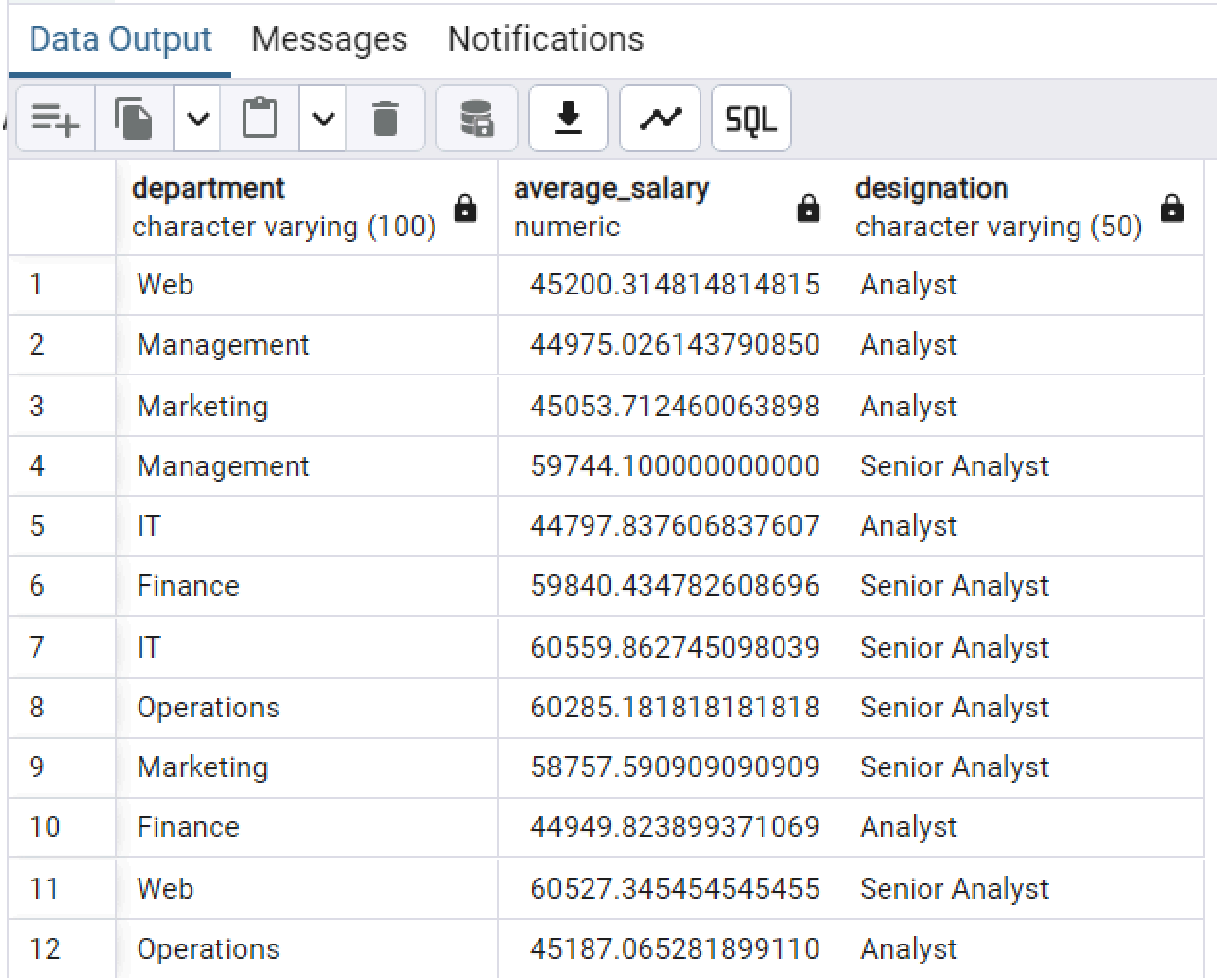
Common Table Expressions (CTEs):

Question 1: Calculate the average salary by department for all Analysts.

Query

```
WITH average_salary AS (

SELECT unit AS department, AVG(salary) AS average_salary, designation

FROM employees

WHERE designation LIKE '%Analyst'

        GROUP BY unit, designation

)

SELECT *

        FROM average_salary;
```

Output



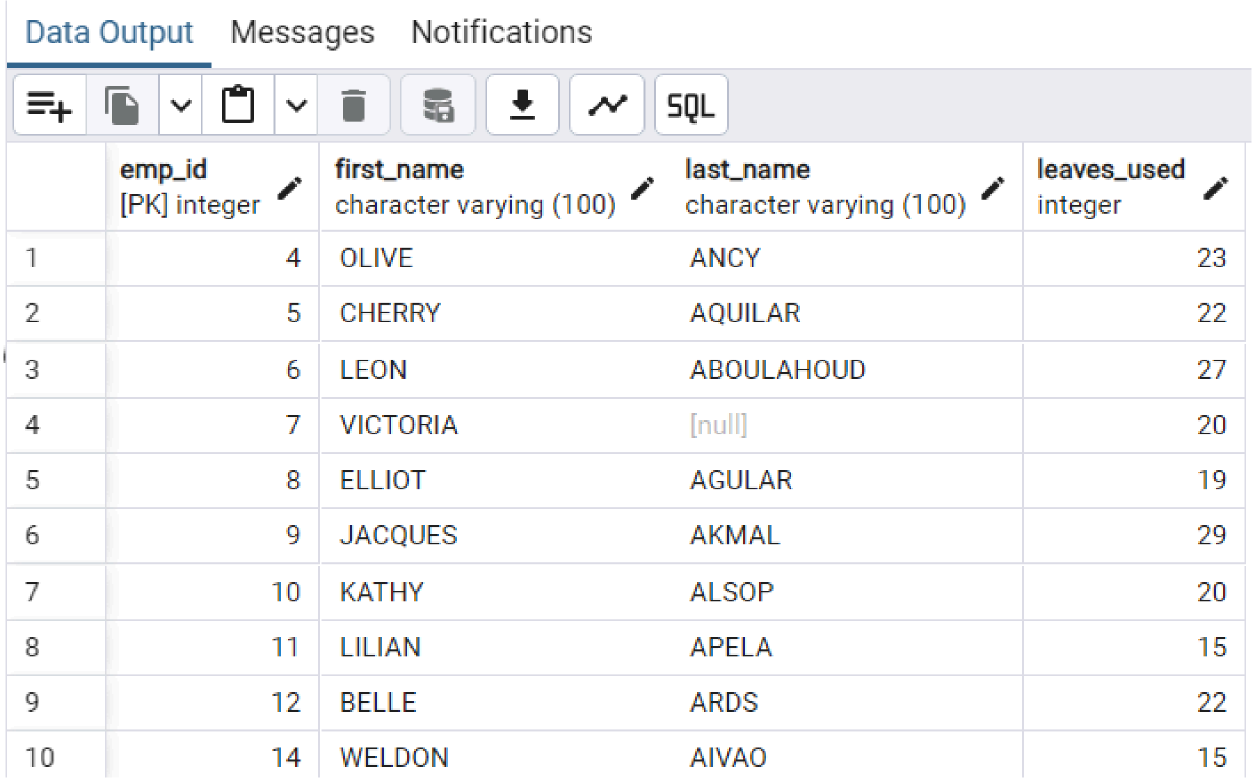
	department character varying (100)	average_salary numeric	designation character varying (50)
1	Web	45200.314814814815	Analyst
2	Management	44975.026143790850	Analyst
3	Marketing	45053.712460063898	Analyst
4	Management	59744.100000000000	Senior Analyst
5	IT	44797.837606837607	Analyst
6	Finance	59840.434782608696	Senior Analyst
7	IT	60559.862745098039	Senior Analyst
8	Operations	60285.181818181818	Senior Analyst
9	Marketing	58757.590909090909	Senior Analyst
10	Finance	44949.823899371069	Analyst
11	Web	60527.345454545455	Senior Analyst
12	Operations	45187.065281899110	Analyst

Question 2: List all employees who have used more than 10 leaves.

Query

```
WITH employees_with_leaves AS (

        SELECT emp_id, first_name, last_name, leaves_used

        FROM employees

        WHERE leaves_used>10

)

SELECT *

        FROM employees_with_leaves;
```

Output



	emp_id [PK] integer	first_name character varying (100)	last_name character varying (100)	leaves_used integer
1	4	OLIVE	ANCY	23
2	5	CHERRY	AQUILAR	22
3	6	LEON	ABOULAHOUD	27
4	7	VICTORIA	[null]	20
5	8	ELLIOT	AGULAR	19
6	9	JACQUES	AKMAL	29
7	10	KATHY	ALSOP	20
8	11	LILIAN	APELA	15
9	12	BELLE	ARDS	22
10	14	WELDON	AIVAO	15

more…

Views:

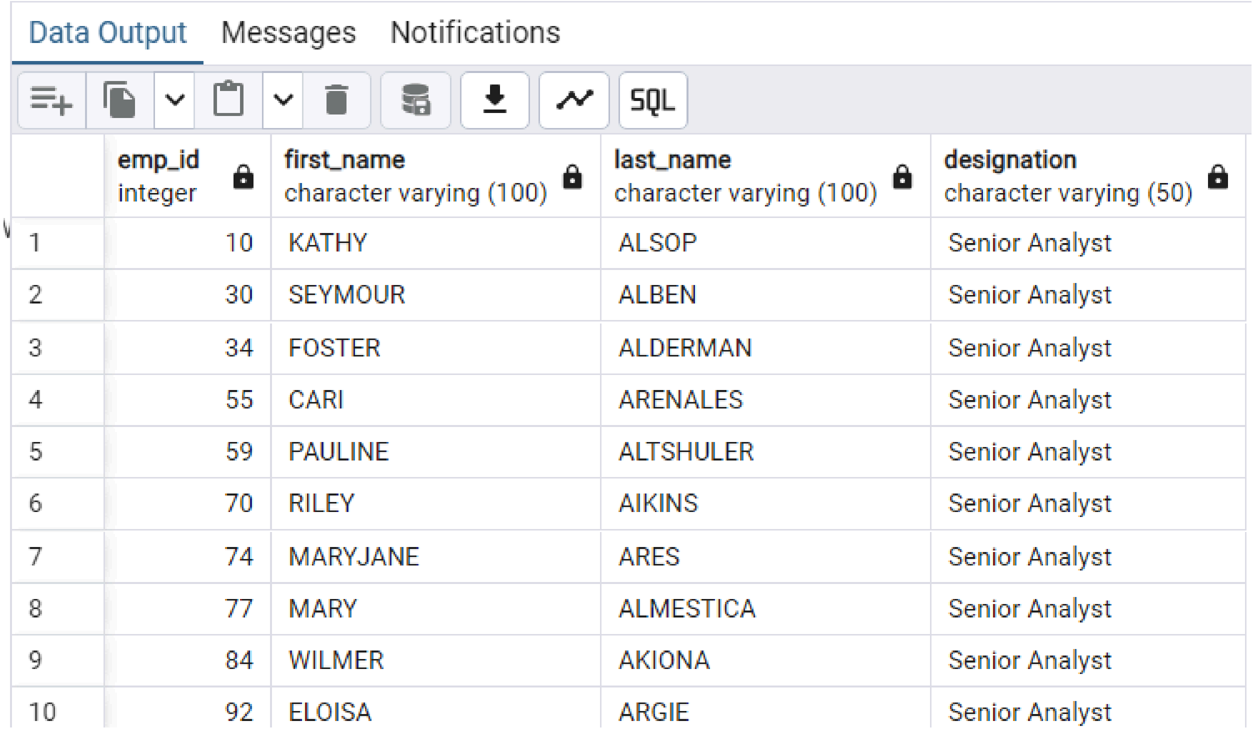
Question 3: Create a view to show the details of all Senior Analysts.

Query

```
CREATE VIEW senior_analysts AS (

        SELECT emp_id, first_name, last_name, designation

        FROM employees

        WHERE designation='Senior Analyst'

);


SELECT *
```

FROM senior_analysts;

Output



	emp_id integer	first_name character varying (100)	last_name character varying (100)	designation character varying (50)
1	10	KATHY	ALSOP	Senior Analyst
2	30	SEYMOUR	ALBEN	Senior Analyst
3	34	FOSTER	ALDERMAN	Senior Analyst
4	55	CARI	ARENALES	Senior Analyst
5	59	PAULINE	ALTSHULER	Senior Analyst
6	70	RILEY	AIKINS	Senior Analyst
7	74	MARYJANE	ARES	Senior Analyst
8	77	MARY	ALMESTICA	Senior Analyst
9	84	WILMER	AKIONA	Senior Analyst
10	92	ELOISA	ARGIE	Senior Analyst

more...

Materialized Views:

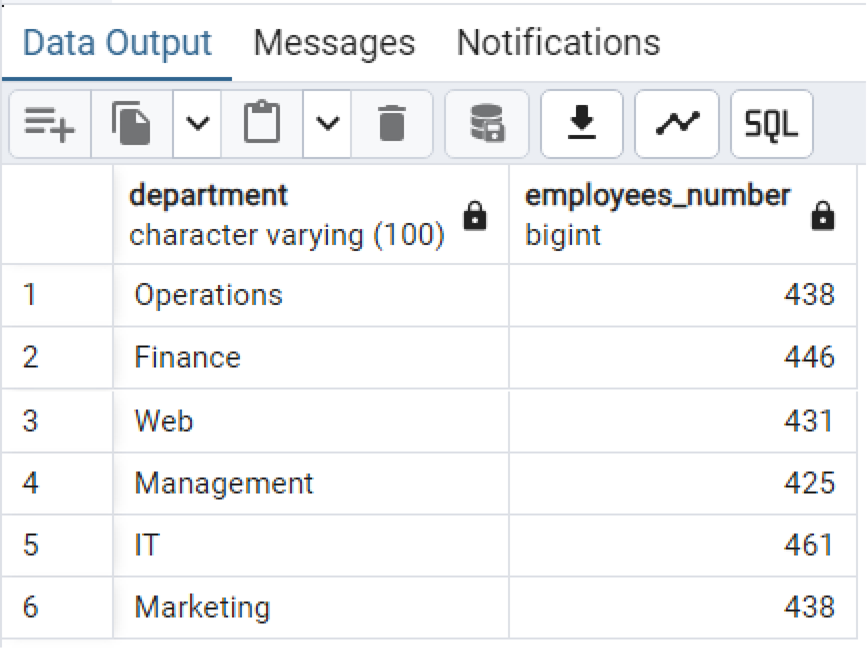
Question 4: Create a materialized view to store the count of employees by department.

Query

```
CREATE MATERIALIZED VIEW employees_in_department AS (

        SELECT unit as department, COUNT(unit) as employees_number

        FROM employees

        GROUP BY unit

);


SELECT *
```

```
FROM employees_in_department;
```

Output



	department character varying (100)	employees_number bigint
1	Operations	438
2	Finance	446
3	Web	431
4	Management	425
5	IT	461
6	Marketing	438

Procedures (Stored Procedures):

Question 5: Create a procedure to update an employee's salary by their first name and last name.

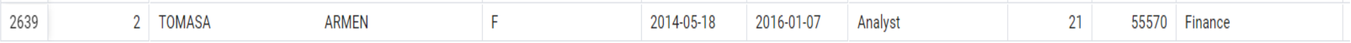
Query

```
CREATE OR REPLACE PROCEDURE update_salary (
        e_first_name varchar,
        e_last_name varchar,
        amount int
)
LANGUAGE plpgsql
AS $$
```

```
BEGIN

    UPDATE employees

    SET salary = salary + amount

    WHERE first_name = e_first_name AND last_name = e_last_name;

          COMMIT;

END;$$;


CALL update_salary('TOMASA', 'ARMEN', 10000);

SELECT * FROM employees;
```

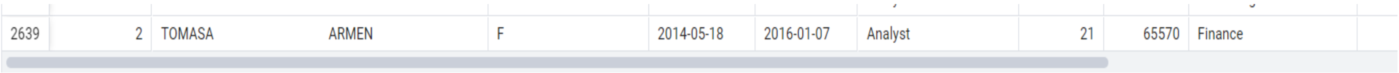
<u>Output</u>

<u>Before</u>



2639	2	TOMASA	ARMEN	F	2014-05-18	2016-01-07	Analyst	21	55570	Finance

<u>After</u>



2639	2	TOMASA	ARMEN	F	2014-05-18	2016-01-07	Analyst	21	65570	Finance

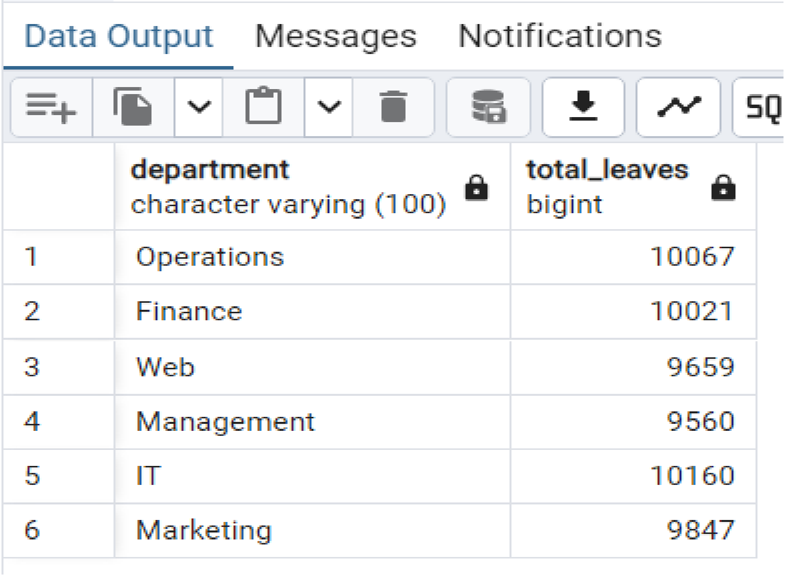
Question 6: Create a procedure to calculate the total number of leaves used across all departments.

<u>Query</u>

```
CREATE OR REPLACE PROCEDURE total_leaves()

LANGUAGE plpgsql

AS $$

BEGIN

          CREATE VIEW total_leaves AS (

                    --total leaves of each departments
```

```
        SELECT unit as department, SUM(leaves_used) as total_leaves

        FROM employees

        GROUP BY unit


        -- total leaves all across departments

        /*SELECT SUM(leaves_used) as total_leaves

        FROM employees */

    );

    COMMIT;

END;$$;

CALL total_leaves();

SELECT * FROM total_leaves;
```

Output



	department character varying (100)	total_leaves bigint
1	Operations	10067
2	Finance	10021
3	Web	9659
4	Management	9560
5	IT	10160
6	Marketing	9847