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:

-- creating table in db

```
CREATE TABLE
    employee_data (first_name VARCHAR(50), last_name VARCHAR(50), sex
CHAR(1), doj DATE, date_current DATE, designation VARCHAR(50), age FLOAT,
salary INT, unit VARCHAR(50), leaves_used FLOAT, leaves_remaining FLOAT,
ratings FLOAT, past_exp INT);
```

-- COPY CSV TO DOCKER

```
docker cp data.csv postgres:/
```

-- copy from csv

```
\copy employee_data (first_name, last_name, sex, doj, date_current,
designation, age, salary, unit, leaves_used, leaves_remaining, ratings,
past_exp) FROM '/path/to/data.csv' DELIMITER ',' CSV HEADER;
```

-- Common Table Expressions (CTEs):
-- Question 1: Calculate the average salary by department for all Analysts.

```
WITH
    Analysts as (
        select
            *
        from
            employee_data
        where
            designation LIKE '%Analyst'
    )
SELECT
    avg(salary) as avg_salary,
    unit
from
    Analysts
```

```
group by
    unit;
```

Data Output Messages Notifications

	avg_salary numeric	unit character varying (50)
1	47305.423469387755	Operations
2	47538.666666666667	Finance
3	47424.554089709763	Web
4	47396.185792349727	Management
5	46797.497512437811	IT
6	47409.559366754617	Marketing

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

```
SELECT
    CONCAT (first_name, ' ', last_name) as Name,
    leaves_used
from
    employee_data
where
    leaves_used > 10;
```

Data Output Messages Notifications

	name text	leaves_used double precision
1	OLIVE ANCY	23
2	CHERRY AQUILAR	22
3	LEON ABOULAHOUD	27
4	VICTORIA	20
5	ELLIOT AGULAR	19
6	JACQUES AKMAL	29
7	KATHY ALSOP	20
8	LILIAN APELA	15
9	WELDON AIVAO	15
10	BOYD AFTON	23
11	BART AGUILLERA	30
12	CORINNE ANDRZEJCZYK	16

Total rows: 1000 of 2636 Query complete 00:00:00.08

-- Views:

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

```
CREATE VIEW
    Seniors as
SELECT
    *
from
    employee_data
where
    designation = 'Senior Analyst';
select * from Seniors;
```

Data Output Messages Notifications

	first_name character varying (50)	last_name character varying (50)	sex character	doj date	date_current date	designation character varying (50)	age double precision	salary integer	unit character varying (50)	leaves_used double precision	leaves_remaining double precision	ratings double precision	pas inte
1	KATHY	ALSOP	F	2014-06-29	2016-01-07	Senior Analyst	28	63478	Operations	20	10	3	
2	SEYMOUR	ALBEN	M	2014-12-21	2016-01-07	Senior Analyst	25	57488	Operations	25	5	3	
3	FOSTER	ALDERMAN	M	2014-05-22	2016-01-07	Senior Analyst	26	68295	Operations	28	2	5	
4	CARI	ARENALES	F	2014-04-10	2016-01-07	Senior Analyst	28	66338	Web	24	6	3	
5	PAULINE	ALTSHULER	F	2014-12-13	2016-01-07	Senior Analyst	28	61647	Finance	28	2	4	
6	RILEY	AIKINS	M	2013-06-16	2016-01-07	Senior Analyst	25	60712	Finance	26	4	4	
7	MARYJANE	ARES	F	2012-08-24	2016-01-07	Senior Analyst	25	65212	Management	29	1	5	
8	MARY	ALMESTICA	F	2013-10-12	2016-01-07	Senior Analyst	27	53339	Finance	26	4	2	
9	WILMER	AKIONA	M	2014-05-30	2016-01-07	Senior Analyst	25	50739	IT	25	5	3	
10	ELOISA	ARGIE	F	2013-08-07	2016-01-07	Senior Analyst	28	52690	Marketing	30	0	5	
11	KATELYN	APPENZELLER	F	2014-11-02	2016-01-07	Senior Analyst	28	56314	Finance	16	14	5	
12	ELIZA	ANSBACHER	F	2013-09-30	2016-01-07	Senior Analyst	25	50813	Marketi				

Successfully run. Total query runtime: 66 msec. 356 rows affected.

Total rows: 356 of 356 Query complete 00:00:00.066 Ln 10, Col 22

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

```
CREATE MATERIALIZED VIEW Emp_count as
SELECT
    count(*),
    unit
from
    employee_data
group by
    Unit;

select * from Emp_count;
```

Data Output Messages Notifications

	count bigint	unit character varying (50)
1	438	Operations
2	446	Finance
3	431	Web
4	425	Management
5	461	IT
6	438	Marketing

-- Procedures (Stored Procedures):
-- Question 6: Create a procedure to update an employee's salary by their first name and last name.

```
CREATE
OR REPLACE PROCEDURE update_salary (firstName varchar(50), lastName
varchar(50), newSalary INT) language plpgsql
as $$
BEGIN
Update employee_data
set
    salary = newSalary
where
```

```
    first_name = firstName
    AND last_name = lastName;
-- commit;
END;
$$;
CALL update_salary ('BELLE','ARDS', 30000);
```

-- Question 7: Create a procedure to calculate the total number of leaves used across all departments.
-- drop procedure total_leave;

```
CREATE OR REPLACE PROCEDURE total_leave(INOUT total_leave int default 0)
LANGUAGE plpgsql
AS $$
BEGIN
    SELECT sum(leaves_used) INTO total_leave
    FROM employee_data;
END;
$$;

call total_leave();
```

Data Output Messages Notificati

	total_leave integer
1	59314