```
CREATE SCHEMA IF NOT EXISTS assignment4;
CREATE TABLE IF NOT EXISTS assignment4.employees(
   id SERIAL PRIMARY KEY,
  first_name VARCHAR(50),
  last_name VARCHAR(50),
  sex CHAR(1),
  doj DATE,
  current_date_ DATE,
  designation VARCHAR(50),
  age INTEGER,
  salary NUMERIC(10, 2),
  unit VARCHAR(50),
  leaves_used INT,
  leaves_remaining INT,
  ratings INT,
  past_exp INT
);
-- Copy csv files to tables
COPY assignment4.employees
(first_name, last_name, sex, doj, current_date_, designation, age, salary,
unit, leaves_used, leaves_remaining, ratings, past_exp)
FROM '/Salary Prediction of Data Professions.csv'
DELIMITER ','
CSV HEADER;
```

```
Analysts.
WITH analyst as(
  SELECT id AS analyst id, unit, salary AS analyst salary
  FROM assignment4.employees
 WHERE designation LIKE '%Analyst'
)SELECT unit, ROUND(avg(analyst_salary), 2)
FROM analyst a
GROUP BY unit;
                   123 round
      Operations
                        47,305.42
       Finance
                         47,720.28
3
       Web
                         47,424.55
                         47,396.19
       Management
                         46,797.5
       Marketing
                         47,440.14
```

```
-- List all employees who have used more than 10 leaves.

WITH employees as(
    SELECT id AS analyst_id, concat(first_name,' ', last_name) AS emp_name
,leaves_used
    FROM assignment4.employees
)SELECT emp_name, leaves_used
FROM employees

WHERE leaves_used > 10;
```

Nabin Thapa - 17 – I Database Assignment 4

•	ABC emp_name	123 leaves_used 🔻	
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	BELLE ARDS	22	
10	WELDON AIVAO	15	
11	BOYD AFTON	23	
12	BART AGUILLERA	30	
13	CORINNE ANDRZEJCZYK	16	

```
-- Create a materialized view to store the count of employees by department.

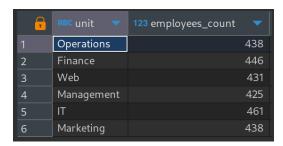
CREATE MATERIALIZED VIEW assignment4.employees_count AS

SELECT unit, COUNT(id) AS employees_count

FROM assignment4.employees

GROUP BY unit;

SELECT * FROM assignment4.employees_count;
```



```
SELECT id, first_name, last_name, salary FROM assignment4.employees
ORDER BY id
LIMIT 5;
       123 id
                   RBC first_name
                                     RBC last_name
                                                       123 salary
                  TOMASA
                                     ARMEN
                                                            100,282.5
                                                              89,207
                2 ANNIE
 2
                                                              40,955
                3 OLIVE
                                     ANCY
                4 CHERRY
                                     AQUILAR
                                                              45,550
                5 LEON
                                     ABOULAHOUD
                                                              43,161
```

```
-- Create a procedure to update an employee's salary by their
first name and last name.
CREATE OR REPLACE PROCEDURE update salary(
 firstName varchar,
 lastName varchar,
 updatePercentage decimal
LANGUAGE plpgsql
AS $$
BEGIN
 UPDATE assignment4.employees
 SET salary = salary + salary * updatePercentage
 WHERE first name = firstName
 AND last_name = lastName;
 COMMIT;
END; $$;
CALL update_salary('OLIVE', 'ANCY', 0.5);
SELECT id, first_name, last_name, salary FROM assignment4.employees
ORDER BY id
```

LIMIT 5;

	12 2 id ▼	RBC first_name	ABC last_name	123 salary 🔻
1	1	TOMASA	ARMEN	100,282.5
2	2	ANNIE		89,207
3	3	OLIVE	ANCY	61,432.5
4	4	CHERRY	AQUILAR	45,550
5	5	LEON	ABOULAHOUD	43,161

```
-- Create a procedure to calculate the total number of leaves
used across all departments.

CREATE OR REPLACE PROCEDURE used_leaves(
   INOUT _total_leaves int DEFAULT 0
)

LANGUAGE plpgsql
AS $$
BEGIN
   SELECT sum(leaves_used)
   FROM assignment4.employees
   INTO _total_leaves;
END;$$;

CALL used_leaves();

123 _total_leaves

59,314
```