

DBMS Assignment.

Date:03 August 2025

Names:UMURERWA Nadine.

Group:F

The database i have created is called company_db. Then using the Postgre sql tool I used this query:

Create database company_db;

After applying connection to the company_db database, i created the tables and according to their relationships so i had almost 4 tables which are:

- 1.employees
- 2.departments
- 3.projects

```

Database [postgres]:
Port [5432]:
Username [postgres]:
Password for user postgres:
psql (17.5)
WARNING: Console code page (437) differs from Windows code page (1252)
        8-bit characters might not work correctly. See psql reference
        page "Notes for Windows users" for details.
Type "help" for help.

postgres=# CREATE DATABASE company_db;
CREATE DATABASE
postgres=# \c company_db;
You are now connected to database "company_db" as user "postgres".
company_db=# CREATE TABLE departments (
company_db(#   department_id INT PRIMARY KEY,
company_db(#   department_name VARCHAR(100)
company_db(# );
CREATE TABLE
company_db=# CREATE TABLE employees (
company_db(#   employee_id INT PRIMARY KEY,
company_db(#   first_name VARCHAR(50),
company_db(#   last_name VARCHAR(50),
company_db(#   email VARCHAR(100),
company_db(#   hire_date DATE,
company_db(#   salary DECIMAL(10, 2),
company_db(#   department_id INT,
company_db(#   FOREIGN KEY (department_id) REFERENCES departments(department_id)
company_db(# );
CREATE TABLE
company_db=# CREATE TABLE projects (
company_db(#   project_id INT PRIMARY KEY,
company_db(#   project_name VARCHAR(100),
company_db(#   start_date DATE,
company_db(#   end_date DATE
company_db(# );
CREATE TABLE
company_db=# CREATE TABLE employee_projects (
company_db(#   employee_id INT,
company_db(#   project_id INT,
company_db(#   assigned_date DATE,
company_db(#   PRIMARY KEY (employee_id, project_id),
company_db(#   FOREIGN KEY (employee_id) REFERENCES employees(employee_id),

```

Then insert into table departments

```
company_db=# INSERT INTO departments (department_id, department_name) VALUES
company_db-# (1, 'Human Resources'),
company_db-# (2, 'Finance'),
company_db-# (3, 'Information Technology'),
company_db-# (4, 'Marketing'),
company_db-# (5, 'Legal'),
company_db-# (6, 'Operations'),
company_db-# (7, 'Customer Service'),
company_db-# (8, 'Sales'),
company_db-# (9, 'Research and Development'),
company_db-# (10, 'Procurement');
INSERT 0 10
```

Employees table

```
company_db=# INSERT INTO employees (employee_id, first_name, last_name, email, hire_date, salary, department_id) VALUES
company_db-# (101, 'Alice', 'Johnson', 'alice.johnson@company.com', '2015-03-15', 4500.00, 1),
company_db-# (102, 'Bob', 'Smith', 'bob.smith@company.com', '2018-06-23', 5200.00, 3),
company_db-# (103, 'Carol', 'Adams', 'carol.adams@company.com', '2012-09-10', 6700.00, 2),
company_db-# (104, 'David', 'Lee', 'david.lee@company.com', '2020-01-05', 3800.00, 4),
company_db-# (105, 'Eve', 'Martins', 'eve.martins@company.com', '2019-12-11', 4000.00, 3),
company_db-# (106, 'Frank', 'Green', 'frank.green@company.com', '2017-07-08', 6000.00, 8),
company_db-# (107, 'Grace', 'Brown', 'grace.brown@company.com', '2014-11-02', 4900.00, 5),
company_db-# (108, 'Hank', 'Wilson', 'hank.wilson@company.com', '2013-02-17', 3100.00, 6),
company_db-# (109, 'Ivy', 'Clark', 'ivy.clark@company.com', '2021-08-30', 2700.00, 9),
company_db-# (110, 'Jake', 'White', 'jake.white@company.com', '2022-05-19', 3600.00, 7);
INSERT 0 10
company_db=#
```

Projects table

```
company_db=# INSERT INTO projects (project_id, project_name, start_date, end_date) VALUES
company_db-# (201, 'HR Revamp', '2023-01-01', '2023-12-31'),
company_db-# (202, 'Finance Automation', '2022-05-15', '2023-04-30'),
company_db-# (203, 'IT Infrastructure Upgrade', '2024-01-01', NULL),
company_db-# (204, 'Marketing Blitz 2025', '2025-02-01', '2025-06-30'),
company_db-# (205, 'Legal Compliance', '2023-07-10', '2024-01-10'),
company_db-# (206, 'Customer Portal', '2021-11-01', '2022-10-31'),
company_db-# (207, 'Sales Booster', '2022-04-01', '2023-03-31'),
company_db-# (208, 'R&D Pilot', '2025-01-01', NULL),
company_db-# (209, 'Procurement Tracker', '2024-03-15', '2024-11-15'),
company_db-# (210, 'Operations Streamline', '2022-09-01', '2023-09-01');
INSERT 0 10
```

Employees_projects

```
company_db=# INSERT INTO employee_projects (employee_id, project_id, assigned_date) VALUES
company_db-# (101, 201, '2023-01-10'),
company_db-# (102, 203, '2024-01-05'),
company_db-# (103, 202, '2022-05-20'),
company_db-# (104, 204, '2025-02-10'),
company_db-# (105, 203, '2024-01-07'),
company_db-# (106, 207, '2022-04-15'),
company_db-# (107, 205, '2023-07-15'),
company_db-# (108, 210, '2022-09-10'),
company_db-# (109, 208, '2025-01-10'),
company_db-# (110, 206, '2021-11-05');
INSERT 0 10
company_db=#
```

Now the questions to perform

1. Concatenate first and last name as full_name.

```
company_db=# select first_name||' '||last_name as full_name from employees;
 full_name
-----
AliceJohnson
BobSmith
CarolAdams
DavidLee
EveMartins
FrankGreen
GraceBrown
HankWilson
IvyClark
JakeWhite
(10 rows)
```

2. Convert all employee names to lowercase

```
company_db=# SELECT LOWER(first_name), LOWER(last_name) FROM employees;
 lower |  lower
-----+-----
alice  | johnson
bob    | smith
carol  | adams
david  | lee
eve    | martins
frank  | green
grace  | brown
hank   | wilson
ivy    | clark
jake   | white
(10 rows)
```

. 3.Extract first 3 letters of the employee's first name

```
company_db=# SELECT SUBSTRING(first_name FROM 1 FOR 3) FROM employees;
 substring
-----
 Ali
 Bob
 Car
 Dav
 Eve
 Fra
 Gra
 Han
 Ivy
 Jak
(10 rows)
```

.4 Replace '@company.com' in email with '@org.com'

```
company_db=# SELECT REPLACE(email, '@company.com', '@org.com')as address FROM employees;
 address
-----
alice.johnson@org.com
bob.smith@org.com
carol.adams@org.com
david.lee@org.com
eve.martins@org.com
frank.green@org.com
grace.brown@org.com
hank.wilson@org.com
ivy.clark@org.com
jake.white@org.com
(10 rows)
```

5. Trim spaces from a padded string.

```
company_db=# SELECT TRIM('  example text  ') AS trimmed;
 trimmed
-----
example text
(1 row)
```

6. Count characters in an employee's full name.

```
company_db=# SELECT LENGTH(first_name || ' ' || last_name) AS full_name_length FROM employees;
full_name_length
-----
13
9
11
9
11
11
11
11
9
10
(10 rows)
```

7. Find position of '@' in email using INSTR()/CHARINDEX().

```
company_db=# SELECT POSITION('@' IN email) AS position FROM employees;
position
-----
14
10
12
10
12
12
12
12
12
10
11
(10 rows)
```

8. Add 'Mr.' or 'Ms.' before names based on gender (assume gender exists)

9. Format project names to uppercase.

```
company_db=# SELECT UPPER(project_name)as project_name FROM projects;
project_name
-----
HR REVAMP
FINANCE AUTOMATION
IT INFRASTRUCTURE UPGRADE
MARKETING BLITZ 2025
LEGAL COMPLIANCE
CUSTOMER PORTAL
SALES BOOSTER
R&D PILOT
PROCUREMENT TRACKER
OPERATIONS STREAMLINE
(10 rows)

company_db=#
```

10. Remove any dashes from project names.

```
company_db=# SELECT REPLACE(project_name, '-', '') AS project_name_dash_removed FROM projects;
project_name_dash_removed
-----
HR Revamp
Finance Automation
IT Infrastructure Upgrade
Marketing Blitz 2025
Legal Compliance
Customer Portal
Sales Booster
R&D Pilot
Procurement Tracker
Operations Streamline
(10 rows)

company_db=#
```

11. Create a label like “Emp: John Doe (HR)”.

```
company_db=# SELECT
company_db=#   'Emp: ' || first_name || ' ' || last_name || ' (' || department_name || ')' AS label
company_db=# FROM employees e
company_db=# JOIN departments d ON e.department_id = d.department_id;
label
-----
Emp: Alice Johnson (Human Resources)
Emp: Bob Smith (Information Technology)
Emp: Carol Adams (Finance)
Emp: David Lee (Marketing)
Emp: Eve Martins (Information Technology)
Emp: Frank Green (Sales)
Emp: Grace Brown (Legal)
Emp: Hank Wilson (Operations)
Emp: Ivy Clark (Research and Development)
Emp: Jake White (Customer Service)
(10 rows)
```

12. Check email length for each employee.

```
company_db=# SELECT email, LENGTH(email) AS email_length FROM employees;
      email      | email_length
-----+-----
alice.johnson@company.com |      25
bob.smith@company.com    |      21
carol.adams@company.com  |      23
david.lee@company.com    |      21
eve.martins@company.com  |      23
frank.green@company.com  |      23
grace.brown@company.com  |      23
hank.wilson@company.com  |      23
ivy.clark@company.com    |      21
jake.white@company.com   |      22
(10 rows)
```

13. Extract last name only from email (before @).

```
company_db=# SELECT SPLIT_PART(email, '@', 1) AS email_user FROM employees;
email_user
-----
alice.johnson
bob.smith
carol.adams
david.lee
eve.martins
frank.green
grace.brown
hank.wilson
ivy.clark
jake.white
(10 rows)
```

14. Format: "LASTNAME, Firstname" using UPPER and CONCAT.

```
company_db=# SELECT UPPER(last_name) || ', ' || INITCAP(first_name) AS formatted_name_to_upper FROM employees;
formatted_name_to_upper
-----
JOHNSON, Alice
SMITH, Bob
ADAMS, Carol
LEE, David
MARTINS, Eve
GREEN, Frank
BROWN, Grace
WILSON, Hank
CLARK, Ivy
WHITE, Jake
(10 rows)
```

15. Add "(Active)" next to employee names who have current projects.

```
company_db=# SELECT
company_db=#   first_name || ' ' || last_name ||
company_db=#   CASE
company_db=#     WHEN ep.project_id IS NOT NULL AND p.end_date IS NULL THEN ' (Active)'
company_db=#     ELSE ''
company_db=#   END AS status
company_db=# FROM employees e
company_db=# LEFT JOIN employee_projects ep ON e.employee_id = ep.employee_id
company_db=# LEFT JOIN projects p ON ep.project_id = p.project_id;
status
-----
Alice Johnson
Bob Smith (Active)
Carol Adams
David Lee
Eve Martins (Active)
Frank Green
Grace Brown
Hank Wilson
Ivy Clark (Active)
Jake White
(10 rows)
```


Numeric Function

Exercises (10)

16. Round salary to the nearest whole number.

```
company_db=# SELECT ROUND(salary) AS rounded_salary FROM employees;
 rounded_salary
-----
      4500
      5200
      6700
      3800
      4000
      6000
      4900
      3100
      2700
      3600
(10 rows)
```

17. Show only even salaries using MOD.

```
company_db=# SELECT * FROM employees WHERE MOD(salary, 2) = 0;
 employee_id | first_name | last_name | email | hire_date | salary | department_id
-----
      101 | Alice | Johnson | alice.johnson@company.com | 2015-03-15 | 4500.00 | 1
      102 | Bob | Smith | bob.smith@company.com | 2018-06-23 | 5200.00 | 3
      103 | Carol | Adams | carol.adams@company.com | 2012-09-10 | 6700.00 | 2
      104 | David | Lee | david.lee@company.com | 2020-01-05 | 3800.00 | 4
      105 | Eve | Martins | eve.martins@company.com | 2019-12-11 | 4000.00 | 3
      106 | Frank | Green | frank.green@company.com | 2017-07-08 | 6000.00 | 8
      107 | Grace | Brown | grace.brown@company.com | 2014-11-02 | 4900.00 | 5
      108 | Hank | Wilson | hank.wilson@company.com | 2013-02-17 | 3100.00 | 6
      109 | Ivy | Clark | ivy.clark@company.com | 2021-08-30 | 2700.00 | 9
      110 | Jake | White | jake.white@company.com | 2022-05-19 | 3600.00 | 7
(10 rows)
```

18. Show difference between two project end/start dates using DATEDIFF.

```
company_db=# SELECT project_name, (end_date - start_date) AS duration_days FROM projects WHERE end_date IS NOT NULL;
 project_name | duration_days
-----
HR Revamp | 364
Finance Automation | 350
Marketing Blitz 2025 | 149
Legal Compliance | 184
Customer Portal | 364
Sales Booster | 364
Procurement Tracker | 245
Operations Streamline | 365
(8 rows)
```

19. Show absolute difference in salaries between two employees.

```
company_db=# SELECT ABS(e1.salary - e2.salary) AS salary_diff
company_db=# FROM employees e1, employees e2
company_db=# WHERE e1.employee_id = 101 AND e2.employee_id = 102;
 salary_diff
-----
      700.00
(1 row)
```

20. Raise salary by 10% using POWER.

```
company_db=# SELECT salary, salary * POWER(1.10, 1) AS raised_salary FROM employees
 salary | raised_salary
-----+-----
 4500.00 | 4950.0000000000000000000000000000
 5200.00 | 5720.0000000000000000000000000000
 6700.00 | 7370.0000000000000000000000000000
 3800.00 | 4180.0000000000000000000000000000
 4000.00 | 4400.0000000000000000000000000000
 6000.00 | 6600.0000000000000000000000000000
 4900.00 | 5390.0000000000000000000000000000
 3100.00 | 3410.0000000000000000000000000000
 2700.00 | 2970.0000000000000000000000000000
 3600.00 | 3960.0000000000000000000000000000
(10 rows)
```

21. Generate a random number for testing IDs.

```
company_db=# SELECT FLOOR(RANDOM() * 9000 + 1000)::INT AS test_id;
 test_id
-----
      8557
(1 row)
```

22. Use CEIL and FLOOR on a floating salary.

```
company_db=# SELECT salary, CEIL(salary), FLOOR(salary) FROM employees;
 salary | ceil | floor
-----+-----+-----
 4500.00 | 4500 | 4500
 5200.00 | 5200 | 5200
 6700.00 | 6700 | 6700
 3800.00 | 3800 | 3800
 4000.00 | 4000 | 4000
 6000.00 | 6000 | 6000
 4900.00 | 4900 | 4900
 3100.00 | 3100 | 3100
 2700.00 | 2700 | 2700
 3600.00 | 3600 | 3600
(10 rows)
```

23. Use LENGTH() on phone numbers (assume column exists).

24. Categorize salary: High/Medium/Low using CASE.

```
company_db=# SELECT salary,
company_db=# CASE
company_db=#     WHEN salary >= 6000 THEN 'High'
company_db=#     WHEN salary >= 4000 THEN 'Medium'
company_db=#     ELSE 'Low'
company_db=# END AS category
company_db=# FROM employees;
 salary | category
-----+-----
 4500.00 | Medium
 5200.00 | Medium
 6700.00 | High
 3800.00 | Low
 4000.00 | Medium
 6000.00 | High
 4900.00 | Medium
 3100.00 | Low
 2700.00 | Low
 3600.00 | Low
(10 rows)
```

25. Count digits in salary amount. Date/Time Function Exercises (10)

```
company_db=# SELECT salary, LENGTH(TRIM(TO_CHAR(salary, '999999.99'))) - 1 AS digit_count FROM employees;
 salary | digit_count
-----+-----
 4500.00 | 6
 5200.00 | 6
 6700.00 | 6
 3800.00 | 6
 4000.00 | 6
 6000.00 | 6
 4900.00 | 6
 3100.00 | 6
 2700.00 | 6
 3600.00 | 6
(10 rows)
```

26. Show today's date using CURRENT_DATE.

```
company_db=# SELECT CURRENT_DATE;
 current_date
-----
 2025-08-04
(1 row)
```

27. Calculate how many days an employee has worked.

```
company_db=# SELECT first_name, CURRENT_DATE - hire_date AS days_worked FROM employees;
first_name | days_worked
-----+-----
Alice      |          3795
Bob        |          2599
Carol      |          4711
David      |          2038
Eve        |          2063
Frank      |          2949
Grace      |          3928
Hank       |          4551
Ivy        |          1435
Jake       |          1173
```

28. Show employees hired in the current year.

```
company_db=# SELECT * FROM employees WHERE EXTRACT(YEAR FROM hire_date) = EXTRACT(YEAR FROM CURRENT_DATE);
employee_id | first_name | last_name | email | hire_date | salary | department_id
-----+-----+-----+-----+-----+-----+-----
(0 rows)
```

29. Display current date and time using NOW().

```
company_db=# SELECT NOW();
now
-----
2025-08-04 11:26:14.841405-07
(1 row)
```

30. Extract the year, month, and day from hire_date.

```
company_db=# SELECT hire_date, EXTRACT(YEAR FROM hire_date) AS year,
company_db=#           EXTRACT(MONTH FROM hire_date) AS month,
company_db=#           EXTRACT(DAY FROM hire_date) AS day
company_db=# FROM employees;
hire_date | year | month | day
-----+-----+-----+-----
2015-03-15 | 2015 |      3 | 15
2018-06-23 | 2018 |      6 | 23
2012-09-10 | 2012 |      9 | 10
2020-01-05 | 2020 |      1 |  5
2019-12-11 | 2019 |     12 | 11
2017-07-08 | 2017 |      7 |  8
2014-11-02 | 2014 |     11 |  2
2013-02-17 | 2013 |      2 | 17
2021-08-30 | 2021 |      8 | 30
2022-05-19 | 2022 |      5 | 19
(10 rows)
```

31. Show employees hired before 2020.

```
company_db=# SELECT * FROM employees WHERE hire_date < '2020-01-01';
```

employee_id	first_name	last_name	email	hire_date	salary	department_id
101	Alice	Johnson	alice.johnson@company.com	2015-03-15	4500.00	1
102	Bob	Smith	bob.smith@company.com	2018-06-23	5200.00	3
103	Carol	Adams	carol.adams@company.com	2012-09-10	6700.00	2
105	Eve	Martins	eve.martins@company.com	2019-12-11	4000.00	3
106	Frank	Green	frank.green@company.com	2017-07-08	6000.00	8
107	Grace	Brown	grace.brown@company.com	2014-11-02	4900.00	5
108	Hank	Wilson	hank.wilson@company.com	2013-02-17	3100.00	6

(7 rows)

32. List projects that ended in the last 30 days.

```
company_db=# SELECT * FROM projects
company_db=# WHERE end_date IS NOT NULL
company_db=# AND end_date >= CURRENT_DATE - INTERVAL '30 days';
```

project_id	project_name	start_date	end_date
------------	--------------	------------	----------

(0 rows)

33. Calculate total days between project start and end dates.

```
company_db=# SELECT project_name, end_date - start_date AS total_days
company_db=# FROM projects WHERE end_date IS NOT NULL;
```

project_name	total_days
HR Revamp	364
Finance Automation	350
Marketing Blitz 2025	149
Legal Compliance	184
Customer Portal	364
Sales Booster	364
Procurement Tracker	245
Operations Streamline	365

(8 rows)

34. Format date: '2025-07-23' to 'July 23, 2025' (use CONCAT).

```
company_db=# SELECT TO_CHAR(DATE '2025-07-23', 'Month DD, YYYY') AS formatted;
formatted
```

formatted
July 23, 2025

(1 row)

35. Add a CASE: if project still active (end_date IS NULL), show 'Ongoing'.

```
company_db=# SELECT project_name,
company_db=# CASE
company_db=#     WHEN end_date IS NULL THEN 'Ongoing'
company_db=#     ELSE 'Completed'
company_db=# END AS status
company_db=# FROM projects;
```

project_name	status
HR Revamp	Completed
Finance Automation	Completed
IT Infrastructure Upgrade	Ongoing
Marketing Blitz 2025	Completed
Legal Compliance	Completed
Customer Portal	Completed
Sales Booster	Completed
R&D Pilot	Ongoing
Procurement Tracker	Completed
Operations Streamline	Completed

(10 rows)

Conditional Function Exercises (15)

36. Use CASE to label salaries.

```
company_db=# SELECT salary,
company_db=# CASE
company_db=#     WHEN salary >= 6000 THEN 'Top Tier'
company_db=#     WHEN salary >= 4000 THEN 'Mid Tier'
company_db=#     ELSE 'Entry Level'
company_db=# END AS label
company_db=# FROM employees;
```

salary	label
4500.00	Mid Tier
5200.00	Mid Tier
6700.00	Top Tier
3800.00	Entry Level
4000.00	Mid Tier
6000.00	Top Tier
4900.00	Mid Tier
3100.00	Entry Level
2700.00	Entry Level
3600.00	Entry Level

(10 rows)

37. Use COALESCE to show 'No Email' if email is NULL.

```
company_db=# SELECT COALESCE(email, 'No Email') FROM employees;
      coalesce
-----
alice.johnson@company.com
bob.smith@company.com
carol.adams@company.com
david.lee@company.com
eve.martins@company.com
frank.green@company.com
grace.brown@company.com
hank.wilson@company.com
ivy.clark@company.com
jake.white@company.com
(10 rows)
```

38. CASE: If hire_date < 2015, mark as 'Veteran'.

```
company_db=# SELECT first_name, hire_date,
company_db=# CASE
company_db=#     WHEN hire_date < '2015-01-01' THEN 'Veteran'
company_db=#     ELSE 'Newcomer'
company_db=#     END AS status
company_db=# FROM employees;
 first_name | hire_date | status
-----+-----+-----
Alice      | 2015-03-15 | Newcomer
Bob        | 2018-06-23 | Newcomer
Carol      | 2012-09-10 | Veteran
David      | 2020-01-05 | Newcomer
Eve        | 2019-12-11 | Newcomer
Frank      | 2017-07-08 | Newcomer
Grace      | 2014-11-02 | Veteran
Hank       | 2013-02-17 | Veteran
Ivy        | 2021-08-30 | Newcomer
Jake       | 2022-05-19 | Newcomer
(10 rows)
```

39. If salary is NULL, default it to 3000 using COALESCE.

```
company_db=# SELECT COALESCE(salary, 3000) FROM employees;
coalesce
-----
 4500.00
 5200.00
 6700.00
 3800.00
 4000.00
 6000.00
 4900.00
 3100.00
 2700.00
 3600.00
(10 rows)
```

40. CASE: Categorize departments (IT, HR, Other).

```
company_db=# SELECT d.department_name,
company_db=# CASE
company_db=#   WHEN d.department_name = 'Information Technology' THEN 'IT'
company_db=#   WHEN d.department_name = 'Human Resources' THEN 'HR'
company_db=#   ELSE 'Other'
company_db=# END AS category
company_db=# FROM departments d;
 department_name | category
-----+-----
 Human Resources | HR
      Finance   | Other
Information Technology | IT
      Marketing | Other
        Legal   | Other
      Operations | Other
Customer Service | Other
        Sales   | Other
Research and Development | Other
      Procurement | Other
(10 rows)
```

41. CASE: If employee has no project, mark as 'Unassigned'.


```

company_db=# SELECT e.employee_id, first_name,
company_db=# CASE
company_db=#     WHEN ep.employee_id IS NULL THEN 'Unassigned'
company_db=#     ELSE 'Assigned'
company_db=# END AS assignment_status
company_db=# FROM employees e
company_db=# LEFT JOIN employee_projects ep ON e.employee_id = ep.employee_id;
 employee_id | first_name | assignment_status
-----+-----+-----
          101 | Alice      | Assigned
          102 | Bob        | Assigned
          103 | Carol      | Assigned
          104 | David      | Assigned
          105 | Eve        | Assigned
          106 | Frank      | Assigned
          107 | Grace      | Assigned
          108 | Hank      | Assigned
          109 | Ivy        | Assigned
          110 | Jake       | Assigned
(10 rows)

```

42. CASE: Show tax band based on salary.

```

company_db=# SELECT salary,
company_db=# CASE
company_db=#     WHEN salary >= 6000 THEN 'High Tax'
company_db=#     WHEN salary >= 4000 THEN 'Medium Tax'
company_db=#     ELSE 'Low Tax'
company_db=# END AS tax_band
company_db=# FROM employees;
 salary | tax_band
-----+-----
 4500.00 | Medium Tax
 5200.00 | Medium Tax
 6700.00 | High Tax
 3800.00 | Low Tax
 4000.00 | Medium Tax
 6000.00 | High Tax
 4900.00 | Medium Tax
 3100.00 | Low Tax
 2700.00 | Low Tax
 3600.00 | Low Tax
(10 rows)

```

43. Use nested CASE to label project duration.

```

company_db=# SELECT project_name,
company_db=# CASE
company_db=#     WHEN end_date IS NULL THEN 'Unknown'
company_db=#     WHEN end_date - start_date < 100 THEN 'Short-Term'
company_db=#     WHEN end_date - start_date < 365 THEN 'Medium-Term'
company_db=#     ELSE 'Long-Term'
company_db=# END AS duration_label
company_db=# FROM projects;

```

project_name	duration_label
HR Revamp	Medium-Term
Finance Automation	Medium-Term
IT Infrastructure Upgrade	Unknown
Marketing Blitz 2025	Medium-Term
Legal Compliance	Medium-Term
Customer Portal	Medium-Term
Sales Booster	Medium-Term
R&D Pilot	Unknown
Procurement Tracker	Medium-Term
Operations Streamline	Long-Term

(10 rows)

44. Use CASE with MOD to show even/odd salary IDs.

```

company_db=# SELECT employee_id,
company_db=# CASE
company_db=#     WHEN MOD(employee_id, 2) = 0 THEN 'Even'
company_db=#     ELSE 'Odd'
company_db=# END AS id_type
company_db=# FROM employees;

```

employee_id	id_type
101	Odd
102	Even
103	Odd
104	Even
105	Odd
106	Even
107	Odd
108	Even
109	Odd
110	Even

(10 rows)

45. Combine COALESCE + CONCAT for fallback names.

```
company_db=# SELECT COALESCE(first_name, 'Unknown') || ' ' || COALESCE(last_name, 'User') AS full_name
company_db=# FROM employees;
full_name
-----
Alice Johnson
Bob Smith
Carol Adams
David Lee
Eve Martins
Frank Green
Grace Brown
Hank Wilson
Ivy Clark
Jake White
```

46. CASE with LENGTH(): if name length > 10, label "Long Name".

```
company_db=# SELECT first_name || ' ' || last_name AS name,
company_db=# CASE
company_db=# WHEN LENGTH(first_name || last_name) > 10 THEN 'Long Name'
company_db=# ELSE 'Short Name'
company_db=# END AS label
company_db=# FROM employees;
name | label
-----+-----
Alice Johnson | Long Name
Bob Smith | Short Name
Carol Adams | Short Name
David Lee | Short Name
Eve Martins | Short Name
Frank Green | Short Name
Grace Brown | Short Name
Hank Wilson | Short Name
Ivy Clark | Short Name
Jake White | Short Name
(10 rows)
```

47. CASE + UPPER(): if email has 'TEST', mark as dummy account.

```
company_db=# SELECT email,
company_db=# CASE
company_db=# WHEN UPPER(email) LIKE '%TEST%' THEN 'Dummy Account'
company_db=# ELSE 'Real Account'
company_db=# END AS email_type
company_db=# FROM employees;
email | email_type
-----+-----
alice.johnson@company.com | Real Account
bob.smith@company.com | Real Account
carol.adams@company.com | Real Account
david.lee@company.com | Real Account
eve.martins@company.com | Real Account
frank.green@company.com | Real Account
grace.brown@company.com | Real Account
hank.wilson@company.com | Real Account
ivy.clark@company.com | Real Account
jake.white@company.com | Real Account
(10 rows)
```

48. CASE: Show seniority based on hire year (e.g., Junior/Senior).

```

company_db=# SELECT first_name, hire_date,
company_db=# CASE
company_db=#     WHEN EXTRACT(YEAR FROM hire_date) <= 2015 THEN 'Senior'
company_db=#     WHEN EXTRACT(YEAR FROM hire_date) <= 2020 THEN 'Mid-Level'
company_db=#     ELSE 'Junior'
company_db=# END AS level
company_db=# FROM employees;
 first_name | hire_date | level
-----+-----+-----
Alice      | 2015-03-15 | Senior
Bob        | 2018-06-23 | Mid-Level
Carol      | 2012-09-10 | Senior
David      | 2020-01-05 | Mid-Level
Eve        | 2019-12-11 | Mid-Level
Frank      | 2017-07-08 | Mid-Level
Grace      | 2014-11-02 | Senior
Hank       | 2013-02-17 | Senior
Ivy        | 2021-08-30 | Junior
Jake       | 2022-05-19 | Junior
(10 rows)

```

49. Use CASE to determine salary increment range.

```

company_db=# SELECT salary,
company_db=# CASE
company_db=#     WHEN salary < 3000 THEN 'Raise by 15%'
company_db=#     WHEN salary < 5000 THEN 'Raise by 10%'
company_db=#     ELSE 'Raise by 5%'
company_db=# END AS suggestion
company_db=# FROM employees;
 salary | suggestion
-----+-----
4500.00 | Raise by 10%
5200.00 | Raise by 5%
6700.00 | Raise by 5%
3800.00 | Raise by 10%
4000.00 | Raise by 10%
6000.00 | Raise by 5%
4900.00 | Raise by 10%
3100.00 | Raise by 10%
2700.00 | Raise by 15%
3600.00 | Raise by 10%
(10 rows)

```

50. Use CASE with CURDATE() to determine anniversary month.

```

company_db=# SELECT first_name, hire_date,
company_db=# CASE
company_db=#     WHEN EXTRACT(MONTH FROM hire_date) = EXTRACT(MONTH FROM CURRENT_DATE) THEN 'Anniversary Month'
company_db=#     ELSE 'Not Anniversary Month'
company_db=# END AS anniversary_check
company_db=# FROM employees;
 first_name | hire_date | anniversary_check
-----+-----+-----
Alice      | 2015-03-15 | Not Anniversary Month
Bob        | 2018-06-23 | Not Anniversary Month
Carol      | 2012-09-10 | Not Anniversary Month
David      | 2020-01-05 | Not Anniversary Month
Eve        | 2019-12-11 | Not Anniversary Month
Frank      | 2017-07-08 | Not Anniversary Month
Grace      | 2014-11-02 | Not Anniversary Month
Hank       | 2013-02-17 | Not Anniversary Month
Ivy        | 2021-08-30 | Anniversary Month
Jake       | 2022-05-19 | Not Anniversary Month
(10 rows)

```