-- Basic Queries

-- 1. Retrieve all employees' first\_name and their departments.

SELECT first\_name, department

FROM employees;

-- 2. Retrieve all employees' full names (first\_name and last\_name) and their email addresses.

SELECT first\_name, last\_name, email

FROM employees;

-- 3. Retrieve the details of employees who work in the 'Sales' department.

SELECT \*

FROM employees

WHERE department = 'Sales';

-- 4. Retrieve the names of employees who have a salary greater than 50,000.

SELECT first\_name, last\_name

FROM employees

WHERE salary > 50000;

-- Intermediate Queries

-- 5. Retrieve the average salary of employees in each department.

SELECT department, AVG(salary) AS average\_salary

FROM employees

GROUP BY department;

-- 6. Retrieve the details of employees who joined after January 1, 2020.

SELECT \*

FROM employees

WHERE joining\_date > '2020-01-01';

-- 7. Retrieve the number of employees in each department.

SELECT department, COUNT(\*) AS number\_of\_employees

FROM employees

GROUP BY department;

-- 8. Retrieve the highest salary in each department.

SELECT department, MAX(salary) AS highest\_salary

FROM employees

GROUP BY department;

-- 9. Retrieve the details of the youngest employee in each department.

SELECT department, first\_name, last\_name, age

FROM employees

WHERE (department, age) IN (

SELECT department, MIN(age)

FROM employees

GROUP BY department

);

-- 10. Retrieve the details of employees whose age is between 25 and 35.

SELECT \*

FROM employees

WHERE age BETWEEN 25 AND 35;

-- 11. Retrieve the total salary expenditure for each department.

SELECT department, SUM(salary) AS total\_salary\_expenditure

FROM employees

GROUP BY department;

-- Advanced Queries

-- 12. Retrieve the details of employees who do not have an email address.

SELECT \*

FROM employees

WHERE email IS NULL;

-- 13. Retrieve the details of employees whose first name starts with 'A'.

SELECT \*

FROM employees

WHERE first\_name LIKE 'A%';

-- 14. Retrieve the details of employees who are older than 30 years.

SELECT \*

FROM employees

WHERE age > 30;

-- 15. Retrieve the details of employees who joined in the year 2021.

SELECT \*

FROM employees

WHERE EXTRACT(YEAR FROM joining\_date) = 2021;

-- 16. Retrieve the details of employees who have a salary between 30,000 and 60,000.

SELECT \*

FROM employees

WHERE salary BETWEEN 30000 AND 60000;

-- 17. Retrieve the details of employees who work in either 'HR' or 'Finance' departments.

SELECT \*

FROM employees

WHERE department IN ('HR', 'Finance');

-- 18. Retrieve the details of employees who have 'gmail.com' as their email domain.

SELECT \*

FROM employees

WHERE email LIKE '%@gmail.com';

-- 19. Retrieve the details of employees who joined in the last 6 months.

SELECT \*

FROM employees

WHERE joining\_date > CURRENT\_DATE - INTERVAL '6 months';

-- 20. Retrieve the details of employees who have the same first name.

SELECT \*

FROM employees

WHERE first\_name IN (

SELECT first\_name

FROM employees

GROUP BY first\_name

HAVING COUNT(\*) > 1

);

-- 21. Retrieve the details of employees who have the highest salary in the company.

SELECT \*

FROM employees

WHERE salary = (SELECT MAX(salary) FROM employees);

-- 22. Retrieve the details of employees who have been with the company for more than 5 years.

SELECT \*

FROM employees

WHERE joining\_date < CURRENT\_DATE - INTERVAL '5 years';

-- 23. Retrieve the details of employees who have a salary that is above the average salary.

SELECT \*

FROM employees

WHERE salary > (SELECT AVG(salary) FROM employees);

-- 24. Retrieve the details of employees who have a salary that is below the average salary.

SELECT \*

FROM employees

WHERE salary < (SELECT AVG(salary) FROM employees);

-- 25. Retrieve the details of employees who have the same last name.

SELECT \*

FROM employees

WHERE last\_name IN (

SELECT last\_name

FROM employees

GROUP BY last\_name

HAVING COUNT(\*) > 1

);

-- 26. Retrieve the details of employees who have a salary that is within the top 10% of all salaries.

SELECT \*

FROM employees

WHERE salary >= (SELECT PERCENTILE\_CONT(0.9) WITHIN GROUP (ORDER BY salary) FROM employees);

-- 27. Retrieve the details of employees who have a salary that is within the bottom 10% of all salaries.

SELECT \*

FROM employees

WHERE salary <= (SELECT PERCENTILE\_CONT(0.1) WITHIN GROUP (ORDER BY salary) FROM employees);

-- 28. Retrieve the details of employees who have a salary that is within the middle 50% of all salaries.

SELECT \*

FROM employees

WHERE salary BETWEEN (SELECT PERCENTILE\_CONT(0.25) WITHIN GROUP (ORDER BY salary) FROM employees)

AND (SELECT PERCENTILE\_CONT(0.75) WITHIN GROUP (ORDER BY salary) FROM employees);

-- 29. Retrieve the details of employees who have a salary that is within the top 5% of all salaries.

SELECT \*

FROM employees

WHERE salary >= (SELECT PERCENTILE\_CONT(0.95) WITHIN GROUP (ORDER BY salary) FROM employees);

-- 30. Retrieve the details of employees who have a salary that is within the bottom 5% of all salaries.

SELECT \*

FROM employees

WHERE salary <= (SELECT PERCENTILE\_CONT(0.05) WITHIN GROUP (ORDER BY salary) FROM employees);

-- Personal Account Queries

-- 31. Retrieve the details of all transactions made in the last month.

SELECT \*

FROM transactions

WHERE transaction\_date >= CURRENT\_DATE - INTERVAL '1 month';

-- 32. Retrieve the total amount spent on groceries in the last year.

SELECT SUM(amount) AS total\_groceries\_spent

FROM transactions

WHERE category = 'Groceries' AND transaction\_date >= CURRENT\_DATE - INTERVAL '1 year';

-- 33. Retrieve the details of all transactions made with a specific vendor (e.g., Amazon).

SELECT \*

FROM transactions

WHERE vendor = 'Amazon';

-- 34. Retrieve the average monthly expenditure for the current year.

SELECT EXTRACT(MONTH FROM transaction\_date) AS month, AVG(amount) AS average\_expenditure

FROM transactions

WHERE EXTRACT(YEAR FROM transaction\_date) = EXTRACT(YEAR FROM CURRENT\_DATE)

GROUP BY EXTRACT(MONTH FROM transaction\_date)

ORDER BY month;

-- 35. Retrieve the highest single transaction amount for each category.

SELECT category, MAX(amount) AS highest\_transaction

FROM transactions

GROUP BY category;

-- 36. Retrieve the details of all transactions above a certain amount (e.g., 10,000 INR).

SELECT \*

FROM transactions

WHERE amount > 10000;

-- 37. Retrieve the total amount spent in each category for the current year.

SELECT category, SUM(amount) AS total\_spent

FROM transactions

WHERE EXTRACT(YEAR FROM transaction\_date) = EXTRACT(YEAR FROM CURRENT\_DATE)

GROUP BY category;

-- 38. Retrieve the number of transactions made in each category for the current year.

SELECT category, COUNT(\*) AS number\_of\_transactions

FROM transactions

WHERE EXTRACT(YEAR FROM transaction\_date) = EXTRACT(YEAR FROM CURRENT\_DATE)

GROUP BY category;

-- 39. Retrieve the details of all transactions made on weekends.

SELECT \*

FROM transactions

WHERE EXTRACT(DOW FROM transaction\_date) IN (0, 6);

-- 40. Retrieve the details of the transactions with the highest amount spent in each month for the current year.

SELECT \*

FROM transactions

WHERE (EXTRACT(MONTH FROM transaction\_date), amount) IN (

SELECT EXTRACT(MONTH FROM transaction\_date), MAX(amount)

FROM transactions

WHERE EXTRACT(YEAR FROM transaction\_date) = EXTRACT(YEAR FROM CURRENT\_DATE)

GROUP BY EXTRACT(MONTH FROM transaction\_date)

);