**WEEK 2**

**NAME:** ANISHA RAJ

**SUPERSET ID:** 6362542

**TOPIC:** Advanced SQL

**Exercise 1: Ranking and Window Functions**

**CODE:-**

CREATE TABLE employees (

id INT,

name VARCHAR(50),

department VARCHAR(50),

salary INT

);

INSERT INTO employees (id, name, department, salary) VALUES

(1, 'Anisha', 'HR', 52000),

(2, 'Rahul', 'IT', 68000),

(3, 'Priya', 'HR', 48000),

(4, 'Karan', 'IT', 72000),

(5, 'Meera', 'HR', 56000),

(6, 'Arjun', 'IT', 68000),

(7, 'Sneha', 'Finance', 64000),

(8, 'Manav', 'Finance', 60000),

(9, 'Divya', 'IT', 70000);

SELECT

id,

name,

department,

salary,

RANK() OVER (PARTITION BY department ORDER BY salary DESC) AS rank\_in\_dept,

ROW\_NUMBER() OVER (PARTITION BY department ORDER BY salary DESC) AS row\_number\_in\_dept,

LAG(salary) OVER (PARTITION BY department ORDER BY salary DESC) AS prev\_salary,

LEAD(salary) OVER (PARTITION BY department ORDER BY salary DESC) AS next\_salary,

SUM(salary) OVER (

PARTITION BY department

ORDER BY salary DESC

ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW

) AS cumulative\_salary

FROM employees;

**OUTPUT:-**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Name** | **Department** | **Salary** | **Performance Rating 1** | **Performance Rating 2** | **Previous Salary 1** | **Previous Salary 2** | **Total Compensation** |
| 7 | Sneha | Finance | 64000 | 1 | 1 | null | 60000 | 64000 |
| 8 | Manav | Finance | 60000 | 2 | 2 | 64000 | null | 124000 |
| 5 | Meera | HR | 56000 | 1 | 1 | null | 52000 | 56000 |
| 1 | Anisha | HR | 52000 | 2 | 2 | 56000 | 48000 | 108000 |
| 3 | Priya | HR | 48000 | 3 | 3 | 52000 | null | 156000 |
| 4 | Karan | IT | 72000 | 1 | 1 | null | 70000 | 72000 |
| 9 | Divya | IT | 70000 | 2 | 2 | 72000 | 68000 | 142000 |
| 2 | Rahul | IT | 68000 | 3 | 3 | 70000 | 68000 | 210000 |
| 6 | Arjun | IT | 68000 | 3 | 4 | 68000 | null | 278000 |

**Exercise 2 : Create a Stored Procedure**

**CODE:-**

CREATE TABLE employees (

id INTEGER,

name TEXT,

department TEXT,

salary INTEGER

);

INSERT INTO employees (id, name, department, salary) VALUES

(1, 'Anisha', 'HR', 52000),

(2, 'Rahul', 'IT', 68000),

(3, 'Priya', 'HR', 48000),

(4, 'Karan', 'IT', 72000),

(5, 'Meera', 'HR', 56000),

(6, 'Arjun', 'IT', 68000),

(7, 'Sneha', 'Finance', 64000),

(8, 'Manav', 'Finance', 60000),

(9, 'Divya', 'IT', 70000),

(10, 'Neha', 'Finance', 58000),

(11, 'Kritika', 'HR', 54000);

WITH ranked\_employees AS (

SELECT

id,

name,

department,

salary,

ROW\_NUMBER() OVER (

PARTITION BY department

ORDER BY salary DESC

) AS rank\_in\_dept

FROM employees

)

SELECT

id,

name,

department,

salary,

rank\_in\_dept

FROM

ranked\_employees

WHERE

rank\_in\_dept <= 3

ORDER BY

department, rank\_in\_dept;

CREATE VIEW IF NOT EXISTS top\_3\_employees\_by\_dept AS

WITH ranked\_employees AS (

SELECT

id,

name,

department,

salary,

ROW\_NUMBER() OVER (

PARTITION BY department

ORDER BY salary DESC

) AS rank\_in\_dept

FROM employees

)

SELECT

id,

name,

department,

salary,

rank\_in\_dept

FROM

ranked\_employees

WHERE

rank\_in\_dept <= 3;

**OUTPUT:-**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Name** | **Department** | **Salary** | **Rank** |
| 7 | Sneha | Finance | 64000 | 1 |
| 8 | Manav | Finance | 60000 | 2 |
| 10 | Neha | Finance | 58000 | 3 |
| 5 | Meera | HR | 56000 | 1 |
| 11 | Kritika | HR | 54000 | 2 |
| 1 | Anisha | HR | 52000 | 3 |
| 4 | Karan | IT | 72000 | 1 |
| 9 | Divya | IT | 70000 | 2 |
| 2 | Rahul | IT | 68000 | 3 |

**Exercise 3: Return Data from a Stored Procedure**

**CODE:-**

CREATE TABLE employees (

id INTEGER,

name TEXT,

department TEXT,

salary INTEGER

);

INSERT INTO employees (id, name, department, salary) VALUES

(1, 'Anisha', 'HR', 52000),

(2, 'Rahul', 'IT', 68000),

(3, 'Priya', 'HR', 48000),

(4, 'Karan', 'IT', 72000),

(5, 'Meera', 'HR', 56000),

(6, 'Arjun', 'IT', 68000),

(7, 'Sneha', 'Finance', 64000),

(8, 'Manav', 'Finance', 60000),

(9, 'Divya', 'IT', 70000),

(10, 'Neha', 'Finance', 58000),

(11, 'Kritika', 'HR', 54000);

SELECT

id,

name,

department,

salary

FROM

employees

WHERE

salary >= 60000

ORDER BY

salary DESC;

CREATE VIEW IF NOT EXISTS employees\_above\_60k AS

SELECT

id,

name,

department,

salary

FROM

employees

WHERE

salary >= 60000

ORDER BY

salary DESC;

**OUTPUT:-**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Department** | **Salary** |
| 4 | Karan | IT | 72000 |
| 9 | Divya | IT | 70000 |
| 2 | Rahul | IT | 68000 |
| 6 | Arjun | IT | 68000 |
| 7 | Sneha | Finance | 64000 |
| 8 | Manav | Finance | 60000 |