ROW_NUMBER()
DENSE_RANK()

```
DROP TABLE IF EXISTS employees;

CREATE TABLE employees (
    employee_id INTEGER NOT NULL PRIMARY KEY,
    name VARCHAR(100),
    department VARCHAR(50),
    salary INTEGER,
    joining_date DATE
);

INSERT INTO employees (employee_id, name, department, salary, joining_date) VALUES
(1, 'Judy', 'HR', 60000, '2020-01-15'),
(2, 'Khan', 'IT', 75000, '2019-11-23'),
(3, 'Sameer', 'IT', 75000, '2021-03-01'),   -- duplicate salary
(4, 'Carlos', 'Finance', 68000, '2018-07-12'),
(5, 'Eve', 'Finance', 70000, '2020-06-30'),
(6, 'Happy', 'HR', 62000, '2019-05-17'),
(7, 'Grace', 'IT', 77000, '2020-10-10'),
(8, 'Heidi', 'Finance', 70000, '2022-02-14'), -- duplicate salary
(9, 'Ivan', 'HR', 62000, '2021-08-19'),       -- duplicate salary
(10, 'Alice', 'IT', 73000, '2022-01-01');
```

	employee_id	name	department	salary	joining_date
1	1	Judy	HR	60000	2020-01-15
2	2	Khan	IT	75000	2019-11-23
3	3	Sameer	IT	75000	2021-03-01
4	4	Carlos	Finance	68000	2018-07-12
5	5	Eve	Finance	70000	2020-06-30
6	6	Happy	HR	62000	2019-05-17
7	7	Grace	IT	77000	2020-10-10
8	8	Heidi	Finance	70000	2022-02-14
9	9	Ivan	HR	62000	2021-08-19
10	10	Alice	IT	73000	2022-01-01

ROW_NUMBER()

- Assigns **unique sequential number** to each row
- No ties allowed (even if salary is same)

```
SELECT name,
    department,
    salary,
    ROW_NUMBER() OVER (ORDER BY salary DESC) AS row_num
FROM employees;
```

Even if two employees have same salary,
every row gets a **unique** number.
Example: Khan and Sameer have same salary and they get **unique** number in row_number

	name	department	salary	row_num
1	Grace	IT	77000	1
2	Khan	IT	75000	2
3	Sameer	IT	75000	3
4	Alice	IT	73000	4
5	Eve	Finance	70000	5
6	Heidi	Finance	70000	6
7	Carlos	Finance	68000	7
8	Happy	HR	62000	8
9	Ivan	HR	62000	9
10	Judy	HR	60000	10

Department-wise Ranking Example:

ROW_NUMBER Per Department

```
SELECT name,
       department,
       salary,
       ROW_NUMBER() OVER (
          PARTITION BY department
          ORDER BY salary DESC
       ) AS dept_row_num

FROM employees;
```

	name	department	salary	dept_row_num
1	Eve	Finance	70000	1
2	Heidi	Finance	70000	2
3	Carlos	Finance	68000	3
4	Happy	HR	62000	1
5	Ivan	HR	62000	2
6	Judy	HR	60000	3
7	Grace	IT	77000	1
8	Khan	IT	75000	2
9	Sameer	IT	75000	3
10	Alice	IT	73000	4

DENSE_RANK()

- Gives same rank to equal values
- Does NOT skip ranking numbers

```
SELECT name,
       department,
       salary,
       DENSE_RANK() OVER (ORDER BY salary DESC) AS dense_rank
FROM employees;
```

If two employees have same salary,
those rows gets a **same** number.
Example: Khan and Sameer have same salary and they get **same** number in dense_rank.

	name	department	salary	dense_rank
1	Grace	IT	77000	1
2	Khan	IT	75000	2
3	Sameer	IT	75000	2
4	Alice	IT	73000	3
5	Eve	Finance	70000	4
6	Heidi	Finance	70000	4
7	Carlos	Finance	68000	5
8	Happy	HR	62000	6
9	Ivan	HR	62000	6
10	Judy	HR	60000	7

DENSE_RANK Per Department

```
SELECT name,
       department,
       salary,
       DENSE_RANK() OVER (
          PARTITION BY department
          ORDER BY salary DESC
       ) AS dept_dense_rank
FROM employees;
```

	name	department	salary	dept_dense_rank
1	Eve	Finance	70000	1
2	Heidi	Finance	70000	1
3	Carlos	Finance	68000	2
4	Happy	HR	62000	1
5	Ivan	HR	62000	1
6	Judy	HR	60000	2
7	Grace	IT	77000	1
8	Khan	IT	75000	2
9	Sameer	IT	75000	2
10	Alice	IT	73000	3

Tect box