# SQL Window Functions Tutorial

## 1. Introduction to Window Functions

Window functions perform a calculation across a set of table rows that are related to the current row. They do not collapse rows like aggregate functions.

Syntax:
<function_name>(<expression>) OVER (PARTITION BY <column> ORDER BY <column>)

## 2. Sample Data

```
CREATE TABLE Employees (
    emp_id INT,
    emp_name VARCHAR(50),
    department VARCHAR(50),
    salary INT
);

INSERT INTO Employees VALUES
(1, 'Alice', 'HR', 6000),
(2, 'Bob', 'HR', 5000),
(3, 'Carol', 'IT', 7000),
(4, 'Dave', 'IT', 7500),
(5, 'Eve', 'IT', 7200),
(6, 'Frank', 'Sales', 4000),
(7, 'Grace', 'Sales', 4500);
```

## 3. Using SUM()

Query:
```
SELECT emp_name, department, salary,
       SUM(salary) OVER (PARTITION BY department) AS total_dept_salary
FROM Employees;
```

Explanation: Computes total salary in each department.

## 4. Using AVG()

Query:
```
SELECT emp_name, department, salary,
       AVG(salary) OVER (PARTITION BY department) AS avg_dept_salary
FROM Employees;
```

Explanation: Computes average salary in each department.

## 5. Using RANK()

Query:
```
SELECT emp_name, department, salary,
```

# SQL Window Functions Tutorial

```
    RANK() OVER (PARTITION BY department ORDER BY salary DESC) AS rank_in_dept
FROM Employees;
```

Explanation: Ranks employees in each department by salary, ties get same rank, gaps exist.

## 6. Using ROW_NUMBER()

Query:

```
SELECT emp_name, department, salary,
    ROW_NUMBER() OVER (PARTITION BY department ORDER BY salary DESC) AS row_num
FROM Employees;
```

Explanation: Assigns a unique row number to each employee in a department based on salary.

## 7. Summary

| Function | Handles Ties | Gaps | Use Case |
|---|---|---|---|
| SUM() | N/A | N/A | Aggregates over a window |
| AVG() | N/A | N/A | Averages over a window |
| RANK() | Yes | Yes | Rankings with tied values |
| ROW_NUMBER() | No | No | Unique row number per partition |