**Windows Functions**

1. What is SQL Aggregate Function

-- An aggregate function performs calculation on a set of values to return a single value.

-- Aggregate functions are usually used with the GROUP BY clause of the SELECT statement.

-- HAVING clause is used to write conditions on the value that is returned by the aggregate function.

-- Aggregate functions return the same value each time that it is called with a specific set of input values on same dataset.

-- For example,

**-- Fetch total salary distribution for each department**

SELECT

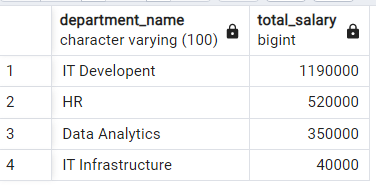
dept.department\_name, SUM(emp.salary) as total\_salary

FROM

dev\_schema.employee as emp INNER JOIN dev\_schema.department as dept

ON emp.fk\_department\_id = dept.department\_id

GROUP BY dept.department\_name ORDER BY total\_salary DESC;



-- The above example is to get sum of salaries by each department. It uses a simple aggregate function which is sum(salary) output by single column grouping.

-- We can add multiple columns in the select statement as long as those columns are included in the group by clause as shown below.

**-- Fetch total salary distribution for each department along with the department id**

SELECT

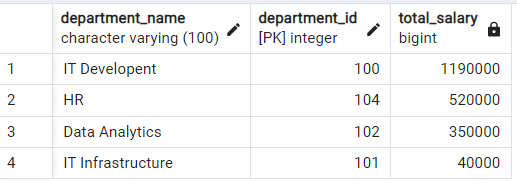
dept.department\_name, dept.department\_id, SUM(emp.salary) as total\_salary

FROM

dev\_schema.employee as emp INNER JOIN dev\_schema.department as dept

ON emp.fk\_department\_id = dept.department\_id

GROUP BY dept.department\_name, dept.department\_id ORDER BY total\_salary DESC;



-- The above query uses two field for grouping the data and for each group the query return a single row as the result.

-- If we try to select other columns that are not in the group by clause, the aggregate function query will give error.

**-- Fetch total salary distribution for each department along with the department id**

SELECT

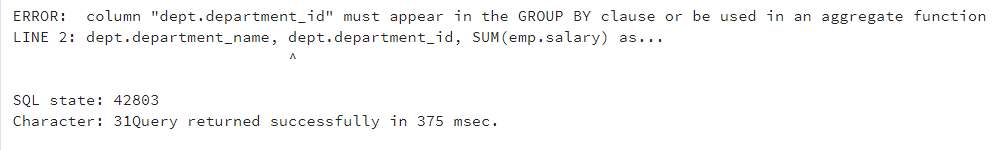
dept.department\_name, dept.department\_id, SUM(emp.salary) as total\_salary

FROM

dev\_schema.employee as emp INNER JOIN dev\_schema.department as dept

ON emp.fk\_department\_id = dept.department\_id

GROUP BY dept.department\_name ORDER BY total\_salary DESC;



-- Here the Aggregate functions worked on a set of rows to return a single result value. If we need only a high-level summary on the data the aggregate functions are useful.

1. What is SQL Window Function

-- The SQL Window Function calculates an aggregate value based on a group of table records called window frame and return multiple rows for each group.

-- FROM clause in the Query filters the data and those virtual table data are considered in a window function for operations.

-- Multiple window functions can slice up the data in different ways by using OVER clauses.

-- We can specify the window frame partition by using PARTITION BY clauses.

-- When we have the OVER clause with an empty parameter, it will operate on the whole set of rows.

-- Below example does calculations sum, average, min, max and percentage using the whole set of rows:

**-- Using empty over()**

SELECT

salary,

SUM(salary) over() as total\_salary,

AVG(salary) over() as avg\_salary,

MAX(salary) over() as max\_salary,

MIN(salary) over() as min\_salary,

(salary::numeric/sum(salary) over()) \* 100 as percentage\_salary

FROM

dev\_schema.employee ORDER BY total\_salary DESC;

