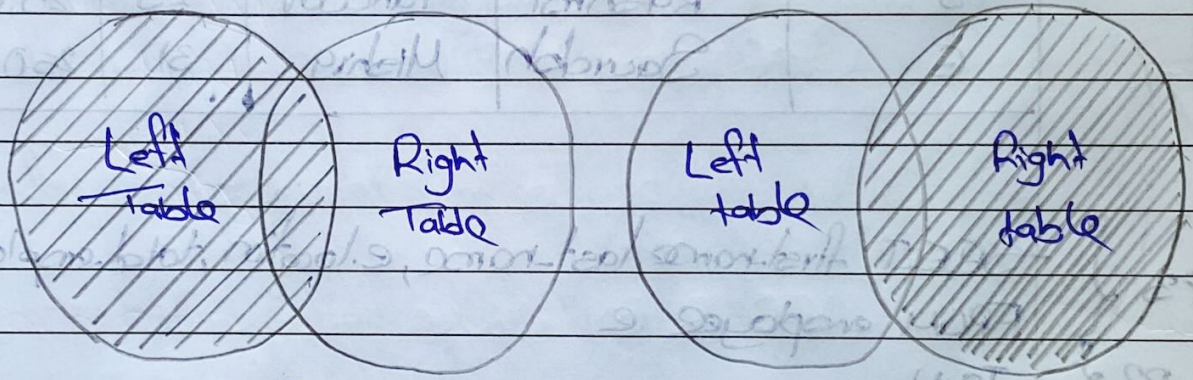
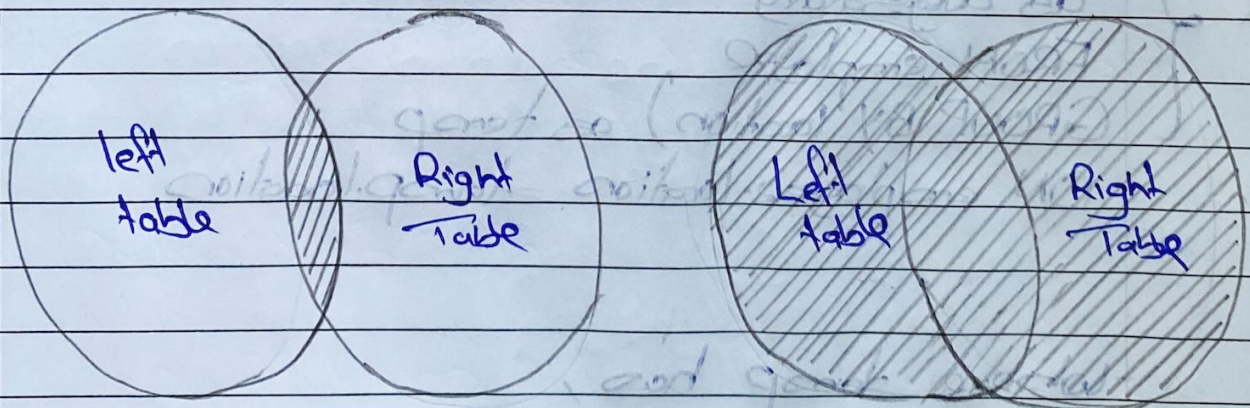
Window Function and Analytic Queries

% ⊙ JOINS



LEFT JOIN

RIGHT JOIN



INNER JOIN (Default)

FULL JOIN

Example:

Select * From employees

EmpID	first_name	last_name	Age	salary	location
1	Ram	Mehta	31	30,000	Bangalore
2	Priya	Bhatia	26	10,000	Bangalore
3	Ajay	Mishra	28	45,000	Noida
4	Harshit	Sidhwa	28	1,00,000	Pune
5	Rashmi	Tancoor	25	25,000	Bangalore
6	Saurabh	Mishra	31	250,000	Hyderabad

P3 { -SELECT first_name, last_name, e.location, total_employee, avg_salary
FROM employee e
P2 { JOIN
P1 { (SELECT location, count(location) as total_employee, avg(salary)
as avgsalary
FROM employee
GROUP BY location) as temp
ON employee.location = temp.location

-where temp has,

Location	total_employee	avg_salary
Bangalore	3	21666.6667
Noida	1	45000.0000
Pune	1	100000.0000
Hyderabad	1	250000.0000

first_name	last_name	location	total_employee	avg_salary
Ram	Mehta	Bangalurd	3	21666.6667
Priya	Bhatia	Bangalore	3	21666.6667
Ajay	Mishra	Noida	1	45000.000
Harshit	Sidhwa	Pune	1	100000
Rashmi	Tanwar	Bangalore	3	21666.6667
Saurabh	Mishra	Hydrabad	1	260000

% Order of Execution

P1 { Internal bracket (Sub query)

P2 { JOIN (INNER JOIN)

P3 { SELECT ... display

%

Window Function

Window functions applies aggregate and ranking functions over a particular window

- OVER clause is used to define that window. OVER clause does two things

1. Partitions rows into form set of rows. (PARTITION BY clause is used)

2. Orders rows within those partitions into a particular order (ORDER BY clause is used)

Example:

```
SELECT first_name, last_name, location,
COUNT(location) OVER (PARTITION BY location) as total_employee,
AVG(salary) OVER (PARTITION BY location) as
avg_salary
FROM employees
```

- Input employees table

EmpID	first_name	last_name	age	Salary	location
1	Ram	Mehta	31	30,000	Bengaluru
2	Priya	Bhatia	26	10,000	Bengaluru
3	Ajay	Mishra	28	45,000	Noida
4	Harshit	Sidhaa	28	1,00,000	Pune
5	Rashmi	Tancon	25	25,000	Bengaluru
6	Saurabh	Mishra	31	2,50,000	Hyderbad

- Ouput:

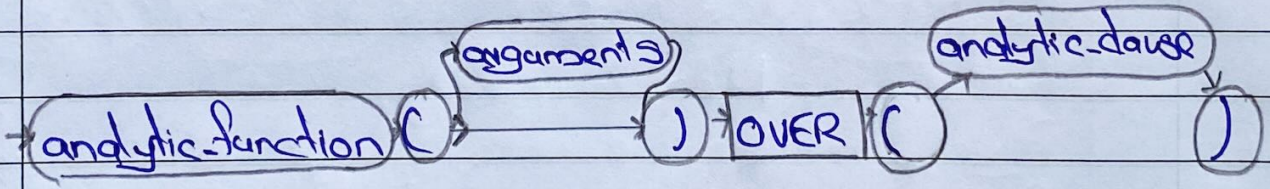
first_name	last_name	location	total_employee	avg_salary
Ram	Mehta	Bengaluru	3	21666.6667
Priya	Bhatia	Bengaluru	3	21666.6667
Rashmi	Tancon	Bengaluru	3	21666.6667
Saurabh	Mishra	Hyderbad	1	250000.0000
Ajay	Mishra	Noida	1	45000.0000
Harshit	Sidhaa	Pune	1	100000.0000

1. FROM
2. SELECT
3. Window function
(partition done then aggregation)

% Analytic Function

Analytic functions are the last set of operation performed in a query except for the final ORDER BY clause.
All JOINS and all WHERE, GROUP BY and HAVING clauses are completed before the analytic functions are processed

- ROW_NUMBER()
- RANK() ⟶ [skipping numeric data for which the rank are same]
- DENSE_RANK()



% ROW_NUMBER() vs RANK() vs DENSE_RANK

-- Give me the details of those employee who are having 2nd highest salary

```
SELECT * FROM
(SELECT first_name, last_name, salary,
DENSE_RANK() OVER (ORDER BY salary DESC) as Priority_Salary
FROM employee ) as temp
WHERE Priority_Salary =2
```

first_name	last_name	Salary	Priority_Salary
Hardik	Sadhwa	100,000	2