

SQL Notes

Date: _____

SQL Execution Order

1. From
2. Join
3. ON
4. where
5. Group by
6. Having by
7. Select
8. Distinct
9. Order by
10. Limit / offset

Common Mistakes:

applying where before aggregation

or Having after Group By

9. check output

→ Always validate result step by step

Date: / /

Query Building Approach (how to think in SQL)

1. understand the Question

→ what is being asked? "find total", "filter", "Compare", "rank" etc.

2. Identify Source Tables

→ which table contains the data you need?

3. Start small

→ Begin with `Select * from table`; to explore columns.

4. Filter Data

→ Apply where to narrow down rows

5. Aggregate or Group

→ use `group by` + aggregate functions
(`Sum`, `Avg`, `Count`).

6. Add Joins

→ Bring data from related tables only when required

7. Rank or Compare (if needed)

→ use window functions (like `Row_number`, `Rank` etc)

8. Sort and Format

→ `order by`, `Round`, `Date_format` etc

Date: / /

Aggregation

Most Common SQL Functions

	<u>Function</u>	<u>Purpose</u>
1.	Count()	counts rows
2.	Sum()	Adds up numeric values
3.	Avg()	Finds average
4.	MIN() , MAX()	find smallest/largest values

Conditional Logic

	<u>Function</u>	<u>use case</u>
1.	Case when <u>Condition</u> Then <u>Value</u> ELSE <u>other</u> END	conditional mapping
2.	COALESCE (Expr1, 2, ...)	Returns first Non-null value
3.	NULLIF (a, b)	Returns null if a=b
4.	If (Condition , true_val , false_val)	shortcut for simple cases (MySQL specific)

ToP

We will often use CASE in select for labelling and in WHERE for filtering logic.

Date: / /

String functions

	<u>function</u>	<u>Purpose</u>
1.	LENGTH()	Character Count
2.	UPPER(), LOWER()	Change Case
3.	SUBSTRING(str, start, len)	Extract Substring
4.	CONCAT()	Combine strings
5.	TRIM(), LTRIM(), RTRIM()	Remove spaces
6.	REPLACE(str, from, to)	Replace text

Filtering and Null handling

	<u>Function</u>	<u>Purpose</u>
1.	IS NULL, IS NOT NULL	Checks NULL
2.	IN, NOT IN	Match list
3.	BETWEEN a AND b	Range check
4.	LIKE , ILIKE	Pattern Match
5.	DISTINCT()	Remove Duplicates

Date: / /

Date & Time Functions

<u>Function</u>	<u>Purpose</u>
1. DATE()	Extract date from date time
2. YEAR(), MONTH(), DAY()	Extract Parts
3. DATEDIFF (date1, date2)	Days difference
4. Date_ADD (date, Interval n Day)	Add days
5. Date_format (date, '%y - %m - %d')	format date
6. Extract (Quarter from date)	useful for quarterly problems

use Cases :

1. find top N per group
2. find growth or decline using LAG()
3. Compare current and previous records
4. Rank Employees, Sales, Transactions

Date: / /

Window Functions

- Window functions perform calculations without collapsing rows, unlike Group By.

Syntax:

$<\text{function}>()$ over (Partition By $\langle\text{col}\rangle$ Order By $\langle\text{col}\rangle$)

Function

Row_Number()

Sequential rank
(No ties)

Rank()

Gives same rank
for ties, skips numbers

Dense_Rank()

Gives same rank
for ties, no skips

LAG(expr, n)

Value from previous row

LEAD(expr, n)

Value from next row

SUM() OVER (...),

Running totals/average

AVG() OVER (...)

Joins Summary

Type

Description

1. Inner JOIN only matching rows
in both tables

2. LEFT JOIN All from left,
matching from right

3. RIGHT JOIN All from right,
matching from left

4. Full JOIN Everything from both
(if supported)

5. CROSS JOIN Cartesian Product

6. SELF JOIN Join Same
Tables

Tip:

- Always Join on keys like id, customer_id etc
- After Joining check duplicates by counting
rows before and after Join.

Date: / /

GROUP BY vs WINDOW Function

	<u>GROUP BY</u>	<u>WINDOW FUNCTION</u>
1. aspect output	Collapses rows	keeps all rows
2. use case	Aggregation per group	calculations per row with group

Example

Total sales per city

Rank or running totals per city

Problem Solving Patterns:

Pattern

Example Problem

1. Filtering Invalid tweets, Employees without managers

2. Aggregation Department Salary, Sales per city

3. Ranking Nth Highest Salary, Top 3 contestants

4. Comparison Salary Growth (using LAG / LEAD)

5. Join Logic Replace Employee ID, find Duplicate Emails

Date: / /

6)

Date Logic

Transactions by Quarks,
Active users

7)

String formatting

Reformat Names,
Mask Emails

Summary

1) \Rightarrow

filter \rightarrow where, Case

2)

Group \rightarrow Group By, sum, Count

3)

Compare \rightarrow Rank, LAG, LEAD

4)

Format \rightarrow Round, Date format, Concat

5)

orders \rightarrow Order By

Date: / /

6)

Date Logic

Transactions by Quarks,
Active users

7)

String formatting

Reformat Names,
Mask Emails

Summary

1) →

Filter → where, Case

2)

Group → Group By, sum, Count

3)

Compare → Rank, LAG, LEAD

4)

Format → Round, Date format, Concat

5)

orders → Order By