## RDBMS Concepts

- Data Types
- Database, Table
- Type of Relationships
- Type of Keys
- ER diagram

## Types of SQL Commands

- DDL
- DML
- DCL
- TCL
- **DQL**

## BigQuery

(Data Warehouse)

## Order of Execution

- FROM, JOIN
- WHERE
- GROUP By
- HAVING
- **SELECT, Window Functions**
- ORDER By
- LIMIT, OFFSET

## Operator Precedence

## Order of Writing

- **SELECT, Window Functions**
- FROM, JOIN
- WHERE
- GROUP By
- HAVING
- ORDER By
- LIMIT, OFFSET

## Pattern Matching LIKE

Regular Expression (REGEX)

Syntax → Only 2 Symbols

- Like 'A%' OR Like '%M' OR Like '%Class%'
- Like 'IND_A'

Used in

- Where clause
- Conditional Expressions

## Operations with null

**null** (+, -, *, / ) Any Number = **null**

Checking Is null = null => Will give answer FALSE

## SELECT Clause

### Sub Clauses

- AS (Alias)
- DISTINCT

### Conditional Expression

- IFNULL
- IF
  SUM + IF => Get Items Count of Flexibly Merged Groups
- **CASE**

```
      WHEN  <Condition>  THEN <Value 1>
      - - - - - - - - - - - - -
      ELSE     <Value N> )
  END       as      <New Column name>
```

## WHERE Clause → Syntax Variants

- where age > 18
- where genres = 'Horror'
- where genres = ~~'Horror' OR 'Action'~~
- where genres = 'Horror' OR genres ='Action'
- where genres in ('Horror' , 'Action' )
- where genres NOT in ('Horror', 'Action' )
- where department_id in (10, 50, 80)
- where d~~epartment_id in (10, null, 50, 80)~~
- where product_id between 3 and 8 (includes 3 & 8)
- where market_date = '2019-04-03'
- where market_date = ~~2019-04-03~~
- where product_size is NULL
- where product_size is NOT NULL
- Where Column 1 !=5 OR Column 1 is NULL

## Sub-Query

- Sub-Query in WHERE Clause *(returns a Column/ field)*
- Sub-Query in FROM Clause *(returns a Table)*
- Sub-Query in HAVING Clause *(returns a Column/ field)*
- Sub-Query in SELECT Clause will be Correlated Sub-Query

- Correlated Sub-Query → Looping *(returns a field)*
  - SELECT & WHERE Clause can have Correlated Sub-Query
  - Correlated Sub-Query can do the work of Window Functions

- Nested Sub-Query vs Correlated Sub-Query
- Sub-Query vs JOIN

## JOINS (Horizontal merging of Tables)

- 4 Types of JOINs → INNER, LEFT, RIGHT and OUTER
- Tip to Display/Select all Columns of 1 Table in a Join
- How to get non-Matching rows in 2nd Table
- JOIN 'ON' Column having Duplicate value

- Venn Diagrams of different types of JOIN
- Full OUTER JOIN in MySQL (Uses UNION)

- JOIN 'ON' Column having NULL value
- Joining Multiple Tables in 1 Query
- CROSS JOIN

## Self-JOIN

- 1 row compare with Single row
  Self-Referencing Tables, Table has relationship with itself.

- 1 row compare with Multiple rows

## JOIN & Self-JOIN → More Concepts

- NOT equal Condition in JOIN ON clause
- Multiple Conditions in JOIN ON clause

- Is WHERE clause needed in JOIN/Self-JOIN queries as 'JOIN ON' conditions do the same work?

## UNION (Vertical merging of Tables)

## Aggregate Functions without Group By

Aggregation on Entire Table i.e. all the rows in Table

## GROUP By

- Aggregation on a subset of rows within the table

## HAVING By

- Filter Rows after GROUP By clause has been executed
- Need for HAVING Clause => WHERE clause cannot use Aggregation

- Query with both HAVING and WHERE clause
- HAVING clause using a SUB-Query

## Window Functions

- **SYNTAX**

  OVER ( Partition by Col1, Col2 ….. Order by Col1, Col2 …..
  Rows Between unbounded preceding and current row)
  - Partition by
  - Order by
  - Rows Between

- The commonly used Window Aggregate Functions and 9 major Window Analytical Functions have been covered in Pg 1 of this doc.

## Scenarios to Apply → Window Function Frame

- **Window Frame Syntax** => Rows between ……

- **Cumulative/Running Sum (DEFAULT)** →
  Rows between Unbounded Preceding and Current Row

- **Get Largest Value in a Frame →**
  Rows between Unbounded Preceding and Unbounded Following

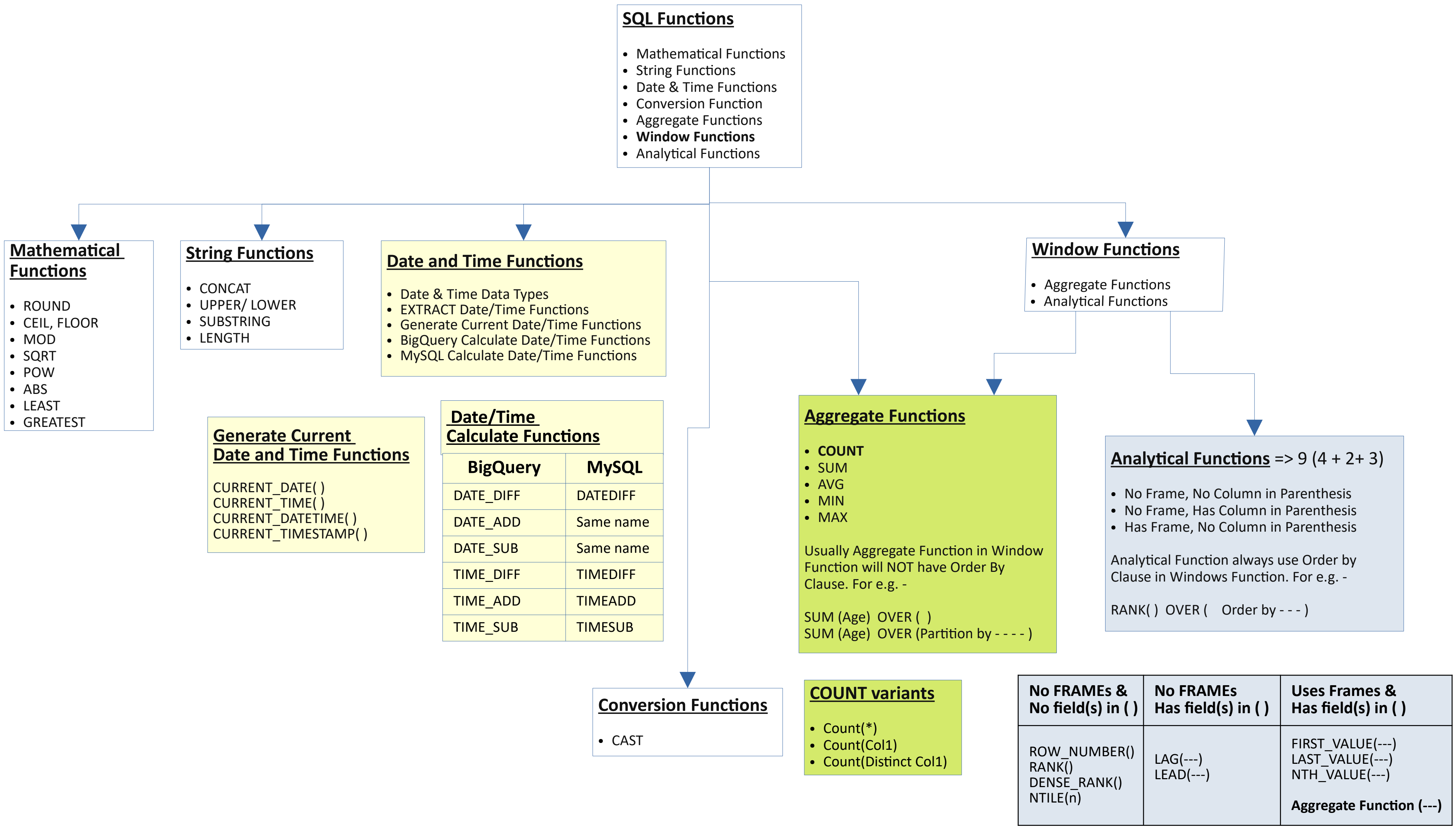
- **Moving Window Frame of 3** →
  Rows between 1 Preceding and 1 Following

## Commands to write Simpler Queries

- **CTE** (Common Table Expression)
- **VIEWs** (Virtual Table)

## Features to help write Efficient Queries

- Indexing
- Partitioning



## SQL Functions

- Mathematical Functions
- String Functions
- Date & Time Functions
- Conversion Function
- Aggregate Functions
- **Window Functions**
- Analytical Functions

## Mathematical Functions

- ROUND
- CEIL, FLOOR
- MOD
- SQRT
- POW
- ABS
- LEAST
- GREATEST

## String Functions

- CONCAT
- UPPER/ LOWER
- SUBSTRING
- LENGTH

## Date and Time Functions

- Date & Time Data Types
- EXTRACT Date/Time Functions
- Generate Current Date/Time Functions
- BigQuery Calculate Date/Time Functions
- MySQL Calculate Date/Time Functions

## Generate Current Date and Time Functions

CURRENT_DATE( )
CURRENT_TIME( )
CURRENT_DATETIME( )
CURRENT_TIMESTAMP( )

## Date/Time Calculate Functions

| BigQuery | MySQL |
|---|---|
| DATE_DIFF | DATEDIFF |
| DATE_ADD | Same name |
| DATE_SUB | Same name |
| TIME_DIFF | TIMEDIFF |
| TIME_ADD | TIMEADD |
| TIME_SUB | TIMESUB |

## Conversion Functions

- CAST

## Window Functions

- Aggregate Functions
- Analytical Functions

## Aggregate Functions

- **COUNT**
- SUM
- AVG
- MIN
- MAX

Usually Aggregate Function in Window Function will NOT have Order By Clause. For e.g. -

SUM (Age) OVER ( )
SUM (Age) OVER (Partition by - - - - )

## COUNT variants

- Count(*)
- Count(Col1)
- Count(Distinct Col1)

## Analytical Functions => 9 (4 + 2+ 3)

- No Frame, No Column in Parenthesis
- No Frame, Has Column in Parenthesis
- Has Frame, No Column in Parenthesis

Analytical Function always use Order by Clause in Windows Function. For e.g. -

RANK( ) OVER ( Order by - - - )

| No FRAMEs & No field(s) in ( ) | No FRAMEs Has field(s) in ( ) | Uses Frames & Has field(s) in ( ) |
|---|---|---|
| ROW_NUMBER()<br>RANK()<br>DENSE_RANK()<br>NTILE(n) | LAG(---)<br>LEAD(---) | FIRST_VALUE(---)<br>LAST_VALUE(---)<br>NTH_VALUE(---)<br>**Aggregate Function (---)** |