1. Introduction to SQL 3

Types of SQL 3

Database client tool 3

RDBMS Database Concepts 3

2. Basic SQL Queries 4

3. Multi-row Functions and `GROUP BY` Clause 4

4. Subqueries 4

What is subqueries 4

Types of Subqueries 4

5. Data Manipulation Language (DML) 5

6. Advanced SQL Concepts 5

- Joins 5

- Set Operations 5

- Indexes and Performance Optimization 6

7. Views 6

8. Combining advanced SQL concepts in queries 6

# Introduction to SQL

SQL (Structured Query Language) is a standardized programming language designed for managing and manipulating relational databases.

## Types of SQL

- Data Definition Language (DDL)

- Used for defining and modifying database structures.

- Examples: `CREATE`, `ALTER`, `DROP`, `TRUNCATE`

- Data Manipulation Language (DML)

- Used for managing data within database objects.

- Examples: `SELECT`, `INSERT`, `UPDATE`, `DELETE`

- Data Control Language (DCL)

- Used for controlling access to data.

- Examples: `GRANT`, `REVOKE`

- Transaction Control Language (TCL)

- Used for managing transactions in the database.

- Examples: `COMMIT`, `ROLLBACK`, `SAVEPOINT`

## Database client tool

Database server acts as core infrastructure software handling data storage, management an d processing. Client tools are means for users to interact with the database servers. They are used to connect to the database server over a network and used for querying, managing and analyzing data. SQL developer is one of such tools for Oracle. There are other command line tools and web-based tools available.

- Interaction Between SQL Client and Database Server:

- SQL clients (tools or applications) send SQL queries to the database server.

- The database server processes the queries and sends back the results to the clients.

## RDBMS Database Concepts

Relational Database Management System (RDBMS) stores data in tables and allows for relationships between tables.

- Characteristics include ACID properties (Atomicity, Consistency, Isolation, Durability).

- Primary Keys, Foreign Keys, and Indexes:

- Primary Key: A unique identifier for a table record.

- Foreign Key: A field in one table that uniquely identifies a row in another table.

- Constraints: SQL enforces data integrity rules within the database by using constraints such as `PRIMARY KEY`, `FOREIGN KEY`, `UNIQUE`, `CHECK`, and `NOT NULL`.

- Indexes: Structures that improve the speed of data retrieval operations.

- Normalization and Denormalization:

- Normalization: The process of organizing data to reduce redundancy.

- Denormalization: The process of combining normalized tables to improve query

performance.

# 2. Basic SQL Queries

- `SELECT` Statement

- Selecting specific columns

- Using `DISTINCT`

- Filtering Data

- `WHERE` clause

- Logical operators (`AND`, `OR`, `NOT`)

- Sorting Data

- `ORDER BY` clause

- Single row functions

# 3. Multi-row Functions and `GROUP BY` Clause

- What is multi row functions and most frequently used ones

- `SUM()`, `AVG()`, `COUNT()`, `MAX()`, `MIN()`

- Grouping Data with `GROUP BY`

- Syntax and usage

- Filtering Grouped Data

- `HAVING` clause

# 4. Subqueries

## What is subqueries

*Subqueries are SQL queries within another SQL query.* Subqueries help in making SQL queries more flexible and easier to manage by

- Simplifying Complex Queries: Breaks complex queries into simpler steps.

- Filtering Data: Allows filtering based on results from another query.

- Improving Readability: Organizes complex logic for better understanding.

## Types of Subqueries

#### - Single-Row Subqueries

A subquery that returns a single row and is used with comparison operators like `=`, `<`, `>`.

Example:

SELECT first\_name, last\_name, salary

FROM employees

WHERE salary = (SELECT MAX(salary) FROM employees);

Explanation: This query retrieves the first name, last name, and salary of the employee who has the highest salary in the `employees` table.

#### - Multi-Row Subqueries

- A subquery that returns multiple rows and is used with operators like `IN`, `ANY`, `ALL`.

- Example:

SELECT first\_name, last\_name

FROM employees

WHERE department\_id IN (SELECT department\_id FROM departments WHERE location\_id = 1700);

- Explanation: This query retrieves the first name and last name of employees who work in departments located at location ID 1700.

#### - Correlated Subqueries:

- A subquery that references a column from the outer query and is executed once for each row of the outer query.

- Example:

SELECT employee\_id, first\_name, last\_name

FROM employees e

WHERE salary > (SELECT AVG(salary) FROM employees WHERE department\_id = e.department\_id);

- Explanation: This query retrieves the employee ID, first name, and last name of employees whose salary is greater than the average salary of their respective departments. The subquery calculates the average salary for each department and compares it with each employee's salary.

# 5. Data Manipulation Language (DML)

- Overview of DML

- `INSERT` Statement

- Inserting single and multiple rows

- `UPDATE` Statement

- Updating specific records

- `DELETE` Statement

- Deleting specific records

- `MERGE` Statement

- Combining `INSERT` and `UPDATE`

# 6. Advanced SQL Concepts

## - Joins

- Inner Join

- Left Join

- join is often used with equality but it can be used with inequality comparisons. e.g.:

select b.title, b.retail, p.gift from books b

inner join promotion p

on b.retail between minretail and maxretail;

## - Set Operations

- `UNION` s

## - Indexes and Performance Optimization

- Importance of indexes

# 7. Views

A view is a virtual table created by a query that presents data from one or more tables. It does not store data itself but shows a result set based on the query used to define it.

Purpose:

- Simplify: Makes complex queries easier to use and understand.

- Secure: Restricts access to specific data, hiding sensitive information.

- Consistent: Provides a uniform way to access data, regardless of the underlying table structure.

- Abstract: Hides the complexity of the underlying data and schema from users.

Creating a View:

- Use the `CREATE VIEW` statement with a query to define which data the view should display.

Querying a View:

- Query a view just like a regular table to retrieve data.

Updating a View:

- Some views are updatable, allowing you to insert, update, or delete data, but this depends on the view's structure and the database system.

Dropping a View:

- Use the `DROP VIEW` statement to remove a view from the database.

Types of Views:

- Simple Views: Based on a single table.

- Complex Views: Involve multiple tables and may include joins or aggregations.

Limitations:

- Can impact performance, especially for complex queries.

- Not all views are updatable.

- Changes to underlying tables can affect the view.

# 8. Combining advanced SQL concepts in queries

Try read the query and see if you can explain what it does. Try understand what each clause does and what it generates eventually.

SELECT e.first\_name, e.last\_name, d.department\_name, MAX(e.salary) AS highest\_salary

FROM employees e

INNER JOIN departments d ON e.department\_id = d.department\_id

WHERE e.job\_id in (SELECT JOB\_ID FROM JOBS WHERE UPPER(JOB\_TITLE) LIKE '%A%')

GROUP BY e.first\_name, e.last\_name, d.department\_name

HAVING MAX(e.salary) > 8000

ORDER BY e.last\_name || ' ' || e.first\_name;