SQL in 8 Days: Complete Roadmap with Practice Problems (Easy, Medium, Hard)

✅Day 1: Basics & SELECT Statements

Topics Covered:

- SELECT
- FROM
- WHERE
- Logical Operators (AND, OR, NOT)

Practice Problems

Easy: 1. Select all columns from the "customers" table. 2. Select names and emails of customers from "India".

Medium: 3. Get customers whose age is between 25 and 40. 4. Get customers who are not from "USA" and are older than 30.

Hard: 5. Get customers who either live in 'Germany' or are younger than 20 but not both.

✅Day 2: Sorting, Limiting & Aliases

Topics Covered:

- ORDER BY
- LIMIT
- DISTINCT
- Aliasing columns with AS

Practice Problems

Easy: 1. Get the top 5 highest-paid employees. 2. Select distinct departments from the "employees" table.

Medium: 3. Show the 3 youngest employees' names and their ages. 4. Create a column showing annual salary (monthly_salary * 12).

Hard: 5. Find employees with the same salary (use ORDER BY and filtering logic).

✅Day 3: Aggregate Functions & GROUP BY

Topics Covered:

- COUNT, SUM, AVG, MIN, MAX
- GROUP BY
- HAVING

Practice Problems

Easy: 1. Count number of customers in "orders". 2. Get total revenue from "sales" table.

Medium: 3. Find average salary per department. 4. Get departments with more than 10 employees.

Hard: 5. Find customers who placed more than 5 orders and average order value > 1000.

✅Day 4: Joins

Topics Covered:

- INNER JOIN
- LEFT JOIN
- RIGHT JOIN
- FULL OUTER JOIN

Practice Problems

Easy: 1. Join "employees" with "departments" and show names with department names. 2. Get all products and their supplier names.

Medium: 3. Show customers who didn’t place any orders (LEFT JOIN + NULL). 4. Get products with no reviews (RIGHT JOIN).

Hard: 5. Join three tables: orders, customers, and payments to get full order history.

✅Day 5: Subqueries

Topics Covered:

- Subqueries in SELECT, FROM, WHERE
- Correlated Subqueries

Practice Problems

Easy: 1. Find employees with salary > average salary. 2. Get products with price more than the average price.

Medium: 3. Find customers who made more than 3 orders (subquery in WHERE). 4. Show products that were never ordered.

Hard: 5. Get top 3 departments with the highest average salary (nested subquery).

✅Day 6: DML & Transactions

Topics Covered:

- INSERT, UPDATE, DELETE
- Transactions: COMMIT, ROLLBACK, SAVEPOINT

Practice Problems

Easy: 1. Insert a new customer into "customers". 2. Update salary of employee with id = 5.

Medium: 3. Delete all employees in the "Intern" role. 4. Begin a transaction, update 2 tables, and rollback.

Hard: 5. Use SAVEPOINT to rollback part of a multi-step salary update process.

✅Day 7: DDL & Schema Design

Topics Covered:

- CREATE TABLE
- ALTER TABLE
- DROP, TRUNCATE
- Constraints (PRIMARY KEY, FOREIGN KEY, UNIQUE, CHECK)

Practice Problems

Easy: 1. Create a table called "students" with id, name, grade. 2. Add a column "email" to "students" table.

Medium: 3. Add a UNIQUE constraint to "email" column. 4. Add a foreign key from "orders.customer_id" to "customers.id".

Hard: 5. Drop and recreate a table with compound primary key and constraint checks.

✅Day 8: Advanced SQL (DCL, TCL, Window Functions, CTEs)

Topics Covered:

- GRANT, REVOKE (DCL)
- Window Functions: RANK(), ROW_NUMBER(), LAG(), LEAD()
- CTEs (Common Table Expressions)
- UNION, INTERSECT, EXCEPT
- CASE WHEN THEN ELSE

Practice Problems

Easy: 1. Use ROW_NUMBER to rank employees by salary. 2. Use CASE to assign grade labels based on marks.

Medium: 3. Use CTE to get employees with salary > department average. 4. Use UNION to combine two SELECTs from archived and current orders.

Hard: 5. Get 2nd highest salary using RANK() inside a CTE. 6. Grant SELECT permission on "sales" to role "analyst".

Let me know if you want this exported as a downloadable PDF or if you'd like a version with sample datasets!