

Day-by-Day SQL Coding Challenges (Days 28–35)

Day 28: DDL Commands

SQL Question 1: CREATE Table

Scenario:

You are a data analyst at City Hospital. Management wants to create a new table to store patient details.

Task:

Write a SQL command to create a table named Patients with fields (PatientID, PatientName, Age, Gender, AdmissionDate).

Expected Output:

A new table Patients is created successfully in the database.

SQL Question 2: ALTER – Add Column

Scenario:

Later, the hospital decides to store the doctor assigned to each patient.

Task:

Write a SQL command to add a new column DoctorAssigned VARCHAR(50) to the Patients table.

Expected Output:

The Patients table now has an additional column DoctorAssigned.

SQL Question 3: ALTER – Modify Column

Scenario:

The hospital realizes that some patient names are longer than 50 characters.

Task:

Write a SQL command to modify the column PatientName from VARCHAR(50) to VARCHAR(100).

Expected Output:

The column PatientName now allows up to 100 characters.

SQL Question 4: RENAME Table

Scenario:

For better clarity, the hospital wants to rename Patients to Patient_Info.

Task:

Write a SQL command to rename the table.

Expected Output:

The table name is successfully changed to Patient_Info.

SQL Question 5: TRUNCATE vs DROP

Scenario:

At year-end, they want to delete all patient records but keep the table. Later, they want to remove it permanently.

Task:

Write SQL commands for TRUNCATE and DROP operations.

Expected Output:

TRUNCATE keeps structure but clears data; DROP removes the table completely.

Day 29: Constraints

SQL Question 1: PRIMARY KEY & FOREIGN KEY

Scenario:

You are creating a database for an online bookstore.

Task:

Define a primary key for Books(BookID) and a foreign key in Orders(BookID) referencing Books.

Expected Output:

Constraints are applied successfully.

SQL Question 2: UNIQUE Constraint

Scenario:

Each book must have a unique ISBN.

Task:

Add a UNIQUE constraint to the ISBN column in Books.

Expected Output:

ISBN values are enforced as unique.

SQL Question 3: DELETE vs TRUNCATE

Scenario:

The store wants to clear test orders but sometimes preserve structure.

Task:

Demonstrate DELETE with WHERE clause and TRUNCATE for Orders table.

Expected Output:

DELETE removes selected rows; TRUNCATE clears all rows quickly.

Day 30: Clauses & Operators

SQL Question 1: DISTINCT & WHERE

Scenario:

In a university database, you want to list unique departments.

Task:

Write a SQL query to return distinct department names.

Expected Output:

Only unique departments are returned.

SQL Question 2: IS NULL & NOT NULL

Scenario:

Some students don't have email addresses recorded.

Task:

Write queries to find students with NULL and NOT NULL emails.

Expected Output:

The queries return correct subsets of students.

SQL Question 3: IN, BETWEEN, NOT BETWEEN

Scenario:

Filter students enrolled in specific courses or within certain GPA ranges.

Task:

Write queries using IN, BETWEEN, and NOT BETWEEN operators.

Expected Output:

Correct sets of students are returned.

Day 31: Sorting & Aggregates

SQL Question 1: ORDER BY & LIMIT

Scenario:

In an e-commerce system, show the top 3 highest-priced products.

Task:

Write a SQL query using ORDER BY and LIMIT.

Expected Output:

Top 3 products by price are displayed.

SQL Question 2: Aggregate Functions

Scenario:

Management wants statistics of sales data.

Task:

Write queries using COUNT, SUM, AVG, MAX, MIN on Sales table.

Expected Output:

Aggregated results are returned.

SQL Question 3: GROUP BY & HAVING

Scenario:

Find departments with more than 10 employees.

Task:

Write a query using GROUP BY and HAVING.

Expected Output:

Only departments with >10 employees are returned.

Day 32: Joins & Union

SQL Question 1: INNER JOIN

Scenario:

Show students with their enrolled course names.

Task:

Write a query joining Students and Courses.

Expected Output:

Results include only students with valid enrollments.

SQL Question 2: LEFT & RIGHT JOIN

Scenario:

List all students and their courses, including those without matches.

Task:

Use LEFT JOIN and RIGHT JOIN between Students and Enrollments.

Expected Output:

All students/courses are shown with NULLs where no match exists.

SQL Question 3: UNION vs UNION ALL

Scenario:

Combine lists of current and past employees.

Task:

Write queries demonstrating UNION and UNION ALL.

Expected Output:

UNION removes duplicates, UNION ALL keeps all rows.

Day 33: Functions

SQL Question 1: String Functions

Scenario:

Clean up employee names for reporting.

Task:

Write queries using UPPER, LOWER, SUBSTRING, CONCAT.

Expected Output:

Formatted name outputs are displayed.

SQL Question 2: Date Functions

Scenario:

Calculate employee tenure in years.

Task:

Use DATE functions like YEAR(), DATEDIFF(), NOW().

Expected Output:

Employee tenure is calculated correctly.

SQL Question 3: User-defined Function

Scenario:

Create a reusable function to return full name of a student.

Task:

Write and test a UDF combining FirstName + LastName.

Expected Output:

Full name is returned when function is called.

Day 34: Procedures & Views

SQL Question 1: Stored Procedure

Scenario:

HR wants a quick way to fetch employee details by ID.

Task:

Create a stored procedure accepting EmployeeID as input.

Expected Output:

Employee details are returned when procedure is executed.

SQL Question 2: Simple View

Scenario:

Management wants a view for employee name and department.

Task:

Write a CREATE VIEW statement.

Expected Output:

A new view is created successfully.

SQL Question 3: Complex View

Scenario:

Create a view joining Employees, Departments, and Salaries.

Task:

Write SQL to define a complex view with multiple joins.

Expected Output:

The view returns combined data from all three tables.

Day 35: Triggers & Transactions

SQL Question 1: Trigger

Scenario:

Log every deletion in the Orders table.

Task:

Write a trigger to insert deleted rows into Order_History.

Expected Output:

Deleted records are logged automatically.

SQL Question 2: DCL Commands

Scenario:

Grant reporting access to junior analysts.

Task:

Write SQL using GRANT and REVOKE commands.

Expected Output:

User privileges are updated accordingly.

SQL Question 3: TCL Commands

Scenario:

During a bank transfer, ensure atomicity.

Task:

Write SQL using COMMIT, ROLLBACK, SAVEPOINT.

Expected Output:

Transaction integrity is maintained.