

DDL Questions

1. **Create a new database** called SchoolDB.
2. **Show all available databases** to verify that SchoolDB has been created.
3. Switch to the SchoolDB database.
4. **Create a table** called Students with the following structure:

- student_id (INT) - Primary key
- first_name (VARCHAR(50))
- last_name (VARCHAR(50))
- grade (VARCHAR(5))
- age (INT)

5. **Create another table** called Courses with the following structure:

- course_id (INT) - Primary key
- course_name (VARCHAR(100))
- teacher (VARCHAR(50))

6. Modifying the Tables:

Add a column email to the Students table.

Change the data type of the age column in the Students table to TINYINT.

7. Renaming and Dropping Tables

A. Rename the Courses table to Subjects

B. Delete the Subjects table permanently from the database

8. Removing Data from the Tables

Truncate the Students table to remove all records but keep the table structure intact.

9. **Drop the SchoolDB database** after completing the assignment.

10. Creating a Table with Constraints

1.

Create a table called Teachers with the following structure:

- teacher_id (INT) - Primary key
- first_name (VARCHAR(50))
- last_name (VARCHAR(50))
- email (VARCHAR(100)) - Must be unique
- hire_date (DATE) - Cannot be NULL

11. Creating a Table with a Foreign Key:

Create a table called Enrollments that tracks which students are enrolled in which courses. The structure should be:

1. enrollment_id (INT) - Primary key
2. student_id (INT) - Foreign key referencing Students(student_id)
3. course_id (INT) - Foreign key referencing Courses(course_id)
4. enrollment_date (DATE)

12. Dropping a Column from a Table

Drop the email column from the Students table.

13. Adding a Default Value to a Column

Add a status column to the Students table with a default value of 'Active'. This column will track whether a student is currently active.

14. Creating an Index

Create an index on the last_name column in the Students table to improve query performance when searching by last name.

15. Rebuild an index that you created in Task 10 to improve its performance after data changes.

Advanced Level SQL DDL Assignment Questions

Partitioning a Table:

Create a Students table with partitioning based on the grade column, where students in different grades are stored in separate partitions.

Creating a View:

Create a view called ActiveStudentsView that displays the student_id, first_name, and last_name of all students who have a status of 'Active' from the Students table.

Using Check Constraints:

Create a Grades table with the following columns: student_id, course_id, and grade. Add a CHECK constraint to ensure that the grade value is between 0 and 100.

Creating a Composite Primary Key:

Create a table called ClassroomAssignments with a composite primary key consisting of the teacher_id and classroom_id.

Adding and Removing Foreign Key Constraints:

Add a foreign key constraint to the Enrollments table that references the teacher_id column in the Teachers table. Then, remove that foreign key constraint.

Using Auto-Increment:

Create a LibraryBooks table where the book_id is automatically incremented whenever a new book is added to the library.

Cloning an Existing Table Structure:

Create a new table called ArchivedStudents by cloning the structure of the existing Students table without copying the data.

Creating a Trigger:

Write a trigger that automatically updates the status column in the Students table to 'Inactive' when the enrollment_date is older than 4 years in the Enrollments table.

Defining a Unique Constraint on Multiple Columns:

In the Enrollments table, add a UNIQUE constraint on the combination of student_id and course_id to ensure a student cannot enroll in the same course more than once.

Renaming a Column:

Rename the first_name column in the Students table to given_name without losing the existing data.