**Practice Questions on SQL DDL & DML**

**Hint : Use below sample records and create yours too if required for practice either using INSERT INTO COMMANDS or you can UPLOAD a csv files and insert data using UPLOAD option both in MySQL & Snowflake**

**Instructions : Use Previous Bulk Load Data Lecture In Order To Finish the task & upload the assignment in GitHub and share it on LinkedIn. You may have to do a bit research in few of the items – Learn the art of Googling 😊**

1. **Create an S3 Bucket (DDL\_DML) and below three respective folder STUDENTS,EMPLOYEE,DEPARTMENTS and add the three files (this will help you practice bulk data upload and can help you in interview and while working for a company)**
2. **Create Staging and Using Copy Command try to get data into respective tables in snowflake and for MySQL you can upload using Load Command**

**  **

**Sample SQL Commands for Creating the Database and Tables**

**-- Create databases**

CREATE DATABASE CompanyDB;

CREATE DATABASE SchoolDB;

**-- Create Employees table in CompanyDB**

CREATE TABLE CompanyDB.Employees (

EmployeeID INT PRIMARY KEY,

FirstName VARCHAR(50),

LastName VARCHAR(50),

DateOfBirth DATE,

Position VARCHAR(50),

Salary DECIMAL(10, 2)

);

**-- Create Departments table in CompanyDB**

CREATE TABLE CompanyDB.Departments (

DepartmentID INT PRIMARY KEY,

DepartmentName VARCHAR(100)

);

**-- Create Students table in SchoolDB**

CREATE TABLE SchoolDB.Students (

StudentID INT PRIMARY KEY,

FirstName VARCHAR(50),

LastName VARCHAR(50),

EnrollmentDate DATE,

Major VARCHAR(50)

);

**1. Database Creation and Table Creation (DDL)**

1. **Create a Database:**
   * Create a database named CompanyDB.
   * Create another database named SchoolDB.
2. **Create Tables:**
   * In the CompanyDB database, create a table named Employees with the following columns:
     + EmployeeID (INT, Primary Key)
     + FirstName (VARCHAR(50))
     + LastName (VARCHAR(50))
     + DateOfBirth (DATE)
     + Position (VARCHAR(50))
     + Salary (DECIMAL(10,2))
   * Create a table named Departments in CompanyDB with the following columns:
     + DepartmentID (INT, Primary Key)
     + DepartmentName (VARCHAR(100))
   * In the SchoolDB database, create a table named Students with the following columns:
     + StudentID (INT, Primary Key)
     + FirstName (VARCHAR(50))
     + LastName (VARCHAR(50))
     + EnrollmentDate (DATE)
     + Major (VARCHAR(50))
3. **Modify Table Structure:**
   * Alter the Employees table to add a new column HireDate (DATE).
   * Modify the Departments table to change the DepartmentName column's data type to VARCHAR(150).
4. **Drop Tables and Database:**
   * Drop the Students table from SchoolDB.
   * Drop the SchoolDB database.

**2. Data Insertion, Updating, and Deletion (DML)**

1. **Insert Data:**
   * Insert five records into the Employees table.
   * Insert three records into the Departments table.
2. **Update Data:**
   * Update the Salary of the employee with EmployeeID = 3 to 75000.
   * Update the Position of all employees where Position is Intern to Junior Developer.
3. **Delete Data:**
   * Delete the employee record with EmployeeID = 4.
   * Delete all records from the Departments table where the DepartmentName is HR.
4. **Complex Insert and Update:**
   * Insert a new employee, ensuring that their DepartmentID exists in the Departments table.
   * Update all employees who have NULL in the HireDate column to the current date.

**3. Data Selection and Filtering (DML)**

1. **Select Data:**
   * Select all columns from the Employees table.
   * Select the FirstName, LastName, and Salary of all employees who have a salary greater than 60000.
2. **Filtering and Sorting:**
   * Select all employees from the Employees table who were hired after 2018-01-01.
   * Select all employees from the Employees table and order them by LastName in ascending order.
3. **Aggregate Functions:**
   * Count the total number of employees in the Employees table.
   * Calculate the average Salary of all employees.

**4. Primary Key and Foreign Key Constraints**

1. **Enforce Uniqueness:**
   * Ensure that the EmployeeID in the Employees table is unique and cannot be NULL.
   * Ensure that each department in the Departments table has a unique DepartmentID that is also not NULL.
2. **Establish Relationships:**
   * Modify the Employees table to add a DepartmentID column (if not already present) and create a foreign key relationship between the Employees table and the Departments table on DepartmentID.
   * Ensure that the DepartmentID in the Employees table cannot have a value that does not exist in the Departments table.
3. **Cascade on Delete:**
   * Modify the foreign key in the Employees table so that if a department is deleted from the Departments table, all employees associated with that department are also deleted.
4. **Enforce Referential Integrity:**
   * Add a foreign key constraint to the Students table, linking the Major column to a hypothetical Majors table that contains all valid majors offered by the school.

**5. Unique and Not Null Constraints**

1. **Ensure Unique Values:**
   * Add a unique constraint on the FirstName and LastName combination in the Employees table, ensuring that no two employees can have the same first and last name combination.
   * Ensure that the DepartmentName in the Departments table is unique.
2. **Prevent NULL Values:**
   * Modify the Employees table to ensure that the FirstName, LastName, DateOfBirth, and Salary columns cannot contain NULL values.
   * In the Departments table, ensure that the DepartmentName cannot be NULL.

**6. Default and Check Constraints**

1. **Set Default Values:**
   * Add a default value of 'Unknown' to the Position column in the Employees table, so if no position is specified, it will default to 'Unknown'.
   * Set a default value of '1000' for the Salary column in the Employees table.
2. **Enforce Valid Data Ranges with CHECK:**
   * Add a check constraint to the Employees table that ensures the Salary is greater than 0.
   * Add a check constraint to the Departments table to ensure that the DepartmentName is at least 3 characters long.
3. **Ensure Valid Date Values:**
   * Add a check constraint to the Employees table to ensure that the HireDate is not in the future.
   * Add a check constraint to the Students table to ensure that the EnrollmentDate is not earlier than 2000-01-01.

**7. Composite Keys and Indexes**

1. **Composite Primary Key:**
   * Create a composite primary key on the combination of FirstName and LastName in the Students table (assuming StudentID is no longer the primary key).
   * In the Employees table, create a composite primary key on the combination of EmployeeID and HireDate (if HireDate is unique for each employee).
2. **Composite Unique Key:**
   * Add a composite unique key on the combination of FirstName and LastName in the Departments table to ensure no two departments can have the same head's first and last name combination.

**Practice Questions on UPDATE, ALTER, and MODIFY Commands**

**1. Altering and Modifying Table Structures**

1. **Modify Data Types:**
   * Change the Salary column in the Employees table to a larger precision, for example, DECIMAL(12,2) to allow for higher salaries.
   * Modify the DateOfBirth column in the Employees table from DATE to DATETIME to include time of birth.
2. **Add New Columns:**
   * Add a new column Email (VARCHAR(100)) to the Employees table to store employee email addresses.
   * Add a new column DepartmentHead (BOOLEAN) to the Departments table to indicate if a department has a head.
3. **Rename Columns:**
   * Rename the Major column in the Students table to Course to better reflect the data it stores.
   * Rename the Position column in the Employees table to JobTitle.
4. **Remove Columns:**
   * Drop the HireDate column from the Employees table as it is no longer needed.
   * Remove the DepartmentHead column from the Departments table.
5. **Change Default Values:**
   * Modify the default value for the Position column in the Employees table to 'Employee' instead of 'Unknown'.
   * Change the default value for the EnrollmentDate in the Students table to the current date.

**2. Updating Data**

1. **Basic Updates:**
   * Update the Salary of all employees with the Position of 'Junior Developer' to 70000.
   * Change the DepartmentName of the department with DepartmentID = 2 to 'Research & Development'.
2. **Conditional Updates:**
   * Update the JobTitle of employees who were hired before 2015-01-01 to 'Senior Developer'.
   * For all students who enrolled before 2020-01-01, change their Course to 'Alumni'.
3. **Bulk Updates:**
   * Increase the Salary of all employees by 10%.
   * Set the DepartmentID of all employees currently in the HR department to NULL (assuming the department is being dissolved).
4. **Update Using Joins:**
   * Update the Salary of employees in the Sales department to 80000 using a join between Employees and Departments on DepartmentID.
   * Set the DepartmentHead to TRUE for the department that has an employee named 'John Doe'.

**3. Advanced Table Modifications**

1. **Reorganize Table:**
   * Change the order of columns in the Employees table to have LastName appear before FirstName.
   * Reorder the Departments table so that DepartmentID is the last column.
2. **Drop and Add Constraints:**
   * Drop the foreign key constraint on DepartmentID in the Employees table, and then add it back with ON DELETE CASCADE.
   * Drop the unique constraint on the combination of FirstName and LastName in the Employees table.
3. **Add and Drop Indexes:**
   * Add an index on the EnrollmentDate column in the Students table to speed up queries.
   * Drop the composite index on FirstName and LastName in the Employees table.

**4. Combined Operations**

1. **Update and Alter Combined:**
   * First, update all employee Salaries to NULL, then modify the Salary column to set a NOT NULL constraint with a default value of 50000.
   * Add a new column PhoneNumber to the Employees table and immediately populate it with a default value for all existing rows.
2. **Modify and Update Combined:**
   * Modify the JobTitle column to accept a maximum of 100 characters, then update all employees with the title 'Intern' to have the title 'Temporary Employee'.
   * Add a column Graduated (BOOLEAN) to the Students table, then update this column to TRUE for all students whose Course is 'Alumni'.

**Practice Questions on DELETE Commands with Conditions**

**1. Basic DELETE Operations**

1. **Delete a Specific Record:**
   * Delete the employee from the Employees table where the EmployeeID is 5.
   * Remove the student from the Students table whose StudentID is 3.
2. **Delete Multiple Records Based on a Condition:**
   * Delete all employees from the Employees table who have a Salary less than 50000.
   * Remove all students from the Students table who enrolled before 2019-01-01.

**2. DELETE with Complex Conditions**

1. **Delete Using AND/OR Conditions:**
   * Delete all employees from the Employees table who are either in the HR department or have a JobTitle of 'Intern'.
   * Remove all students from the Students table whose Course is 'History' and who enrolled before 2021-01-01.
2. **Delete Using EXISTS Clause:**
   * Delete all employees from the Employees table where the department they belong to no longer exists in the Departments table.
   * Remove all students from the Students table if there is no record of their major in a hypothetical Majors table.

**3. DELETE with Cascading and Foreign Keys**

1. **Cascade DELETE Operations:**
   * Delete a department from the Departments table and ensure all related employees are also deleted (requires foreign key with ON DELETE CASCADE).
   * Remove a course from the Courses table (hypothetical) and ensure all students enrolled in that course are also removed.
2. **Delete Using a Restriction:**
   * Attempt to delete a department from the Departments table where there are still employees assigned to that department, and observe what happens (if foreign keys are not set to cascade).
   * Attempt to delete a student from the Students table if they are referenced in another table, such as a Grades table (hypothetical).

**4. Deleting All Records with a Condition**

1. **Delete All Records Based on a Common Attribute:**
   * Delete all records from the Employees table where the JobTitle is 'Consultant'.
   * Remove all records from the Students table where the Course is 'Art'.
2. **Delete Records with NULL Values:**
   * Delete all employees from the Employees table where the Salary is NULL.
   * Remove all students from the Students table where the EnrollmentDate is NULL.
3. **Conditional DELETE Using Aggregate Functions:**
   * Delete employees from the Employees table who earn less than the average salary of all employees.
   * Remove students from the Students table who enrolled after the latest enrollment date in the table.