**ASSIGNEMENT -1**

**CREATE DATABASE CompanyDB;**

**CREATE DATABASE SchoolDB;**

**-- Create Employees table in CompanyDB**

**CREATE TABLE 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 Departments (**

**DepartmentID INT PRIMARY KEY,**

**DepartmentName VARCHAR(100)**

**);**

**-- Create Students table in SchoolDB**

**CREATE TABLE Students (**

**StudentID INT PRIMARY KEY,**

**FirstName VARCHAR(50),**

**LastName VARCHAR(50),**

**EnrollmentDate DATE,**

**Major VARCHAR(50)**

**);**

**--1. Modify Table Structure**

**--a. Alter the Employees table to add a new column HireDate (DATE).**

**ALTER TABLE Employees**

**ADD COLUMN HireDate DATE;**

**SELECT \* FROM EMPLOYEES;**

**--b. Modify the Departments table to change the DepartmentName column's data type to VARCHAR(150).**

**ALTER TABLE DEPARTMENTS**

**ALTER COLUMN DepartmentName VARCHAR(150);**

**--2. Drop Tables and Database:**

**-- a. Drop the Students table from SchoolDB.**

**DROP TABLE STUDENTS;**

**-- b. Drop the SchoolDB database.**

**DROP DATABASE SchoolDB;**

**--3.Data Insertion, Updating, and Deletion (DML)**

**--1.Insert Data:**

**-- a. Insert five records into the Employees table.**

**SELECT \* FROM EMPLOYEES;**

**INSERT INTO EMPLOYEES**

**VALUES**

**(1001, 'Nakul', 'Malik', '1998-05-15', 'Software Engineer', 60000.00, NULL),**

**(1002, 'Anand', 'Jha', '1985-08-20', 'Project Manager', 75000.00, NULL),**

**(1003, 'Prachi', 'Kabra', '1992-12-10', 'Business Analyst', 55000.00, NULL),**

**(1004, 'Suraj', 'Singh', '2000-03-05', 'Data Scientist', 70000.00, NULL),**

**(1005, 'Charlie', 'Brown', '1995-07-30', 'UX Designer', 50000.00, NULL);**

**--b .Insert three records into the Departments table.**

**INSERT INTO DEPARTMENTS**

**VALUES**

**(6,'IT'),**

**(7, 'Admin'),**

**(8,'Strategy');**

**select \* from departments;**

**--Update Data:**

**--a. Update the Salary of the employee with EmployeeID = 3 to 75000.**

**select \* from employees;**

**UPDATE EMPLOYEES**

**SET Salary = 75000.00**

**WHERE EmployeeID = 3;**

**--b. Update the Position of all employees where Position is Intern to Junior Developer.**

**UPDATE EMPLOYEES**

**SET POSITION = 'Junior\_Developer'**

**where POSITION = 'Intern';**

**--3.Delete Data:**

**--a. Delete the employee record with EmployeeID = 4.**

**DELETE FROM EMPLOYEES**

**WHERE EMPLOYEEID = 4;**

**--b. Delete all records from the Departments table where the DepartmentName is HR.**

**DELETE FROM DEPARTMENTS**

**Where departmentname = 'HR';**

**--4.Complex Insert and Update:**

**--a. Insert a new employee, ensuring that their DepartmentID exists in the Departments table.**

**INSERT INTO EMPLOYEES**

**VALUES**

**(1006, 'Mukul', 'Baweja', '1995-05-15', 'Software Engineer', 60000.00, NULL);**

**-- b. Update all employees who have NULL in the HireDate column to the current date.**

**UPDATE EMPLOYEES**

**SET HireDate = CURRENT\_DATE**

**Where HireDate IS Null;**

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

**--1. Select Data:**

**--b. Select all columns from the Employees table**

**SELECT EMPLOYEEID, FIRSTNAME,LASTNAME,POSITION,SALARY,HIREDATE**

**FROM EMPLOYEES;**

**--OR**

**SELECT \* FROM EMPLOYEES;**

**--c. Select the FirstName, LastName, and Salary of all employees who have a salary greater than 60000.**

**SELECT FIRSTNAME, LASTNAME, SALARY**

**FROM EMPLOYEES**

**WHERE SALARY>60000;**

**--2. Filtering and Sorting:**

**--a. Select all employees from the Employees table who were hired after 2018-01-01.**

**SELECT \* FROM EMPLOYEES**

**WHERE HIREDATE > '2018-01-01';**

**--b. Select all employees from the Employees table and order them by LastName in ascending order.**

**SELECT \* FROM EMPLOYEES**

**ORDER BY LASTNAME ASC;**

**--3.Aggregate Functions:**

**--a. Count the total number of employees in the Employees table.**

**SELECT COUNT(EMPLOYEEID) FROM EMPLOYEES;**

**--b. Calculate the average Salary of all employees.**

**SELECT AVG(SALARY) FROM EMPLOYEES;**

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

**--1.Enforce Uniqueness:**

**--a. Ensure that the EmployeeID in the Employees table is unique and cannot be NULL.**

**SELECT DISTINCT EMPLOYEEID FROM EMPLOYEES;**

**--b. Ensure that each department in the Departments table has a unique DepartmentID that is also not NULL.**

**--2. Establish Relationships:**

**--a. 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**

**Alter table EMPLOYEES**

**ADD COLUMN DepartmentID INT;**

**ALTER TABLE EMPLOYEES**

**ADD CONSTRAINT FK\_DepartmentID**

**FOREIGN KEY (DepartmentID) REFERENCES DEPARTMENTS(DepartmentID);**

**--b. Ensure that the DepartmentID in the Employees table cannot have a value that does not exist in the Departments table.**

**INSERT INTO DEPARTMENTS**

**VALUES**

**(1,'HR');**

**SELECT \* FROM EMPLOYEES;**

**SELECT \* FROM DEPARTMENTS;**

**UPDATE EMPLOYEES**

**SET DepartmentID =**

**CASE**

**WHEN POSITION = 'HR Manager' THEN 1**

**WHEN POSITION = 'Senior Developer' THEN 3**

**WHEN POSITION = 'Consultant' THEN 8**

**WHEN POSITION = 'Sales Representative' THEN 2**

**WHEN POSITION = 'Junior\_Developer' THEN 6**

**WHEN POSITION = 'Project Manager' THEN 8**

**WHEN POSITION = 'Business Analyst' THEN 6**

**WHEN POSITION = 'Data Scientist' THEN 6**

**WHEN POSITION = 'UX Designer' THEN 6**

**WHEN POSITION = 'Software Engineer' THEN 3**

**END;**

**--3.Cascade on Delete:**

**--a. 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.**

**DELETE FROM DEPARTMENTS**

**WHERE DEPARTMENTID = 1;**

**--b. Enforce Referential Integrity:**

**--c. 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.**

**--2.Unique and Not Null Constraints**

**--1.Ensure Unique Values:**

**--a. 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.**

**--Need to be discussed**

**--b. Ensure that the DepartmentName in the Departments table is unique.**

**SELECT \* FROM DEPARTMENTS;**

**--They are all unique**

**--2.Prevent NULL Values:**

**--a. Modify the Employees table to ensure that the FirstName, LastName, DateOfBirth, and Salary columns cannot contain NULL values.**

**select FirstName , LastName, DateOfBirth, salary**

**from employees**

**where FirstName is null**

**or LastName is null**

**or DateOfBirth is null**

**or salary is Null;**

**--b. In the Departments table, ensure that the DepartmentName cannot be NULL.**

**select \* from departments**

**where departmentid is null**

**or departmentname is null;**

**--3.Default and Check Constraints**

**--1.Set Default Values:**

**--a. 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'.**

**UPDATE EMPLOYEES**

**SET Position = 'Unknown'**

**WHERE Position IS NULL OR Position = '';**

**--b. Set a default value of '1000' for the Salary column in the Employees table.**

**UPDATE EMPLOYEES**

**SET Salary = 1000**

**WHERE Salary IS NULL OR Salary = '';**

**--2.Enforce Valid Data Ranges with CHECK:**

**--a. Add a check constraint to the Employees table that ensures the Salary is greater than 0.**

**--b. Add a check constraint to the Departments table to ensure that the DepartmentName is at least 3 characters long.**

**--3.Ensure Valid Date Values:**

**--a.Add a check constraint to the Employees table to ensure that the HireDate is not in the future.**

**--b. Add a check constraint to the Students table to ensure that the EnrollmentDate is not earlier than 2000-01-01.**

**--4.Composite Keys and Indexes**

**--NEED TO BE DISCUSSED**

**--1.Composite Primary Key:**

**-- a. Create a composite primary key on the combination of FirstName and LastName in the Students table (assuming StudentID is no longer the primary key).**

**ALTER TABLE Students**

**DROP CONSTRAINT STUDENTID\_PK;**

**select FirstName, LastName from employees**

**concat(firstname,lastname);**

**--b. 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:**

**--a. 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.**

**--1.Altering and Modifying Table Structures**

**--1.Modify Data Types:**

**--a. Change the Salary column in the Employees table to a larger precision, for example, DECIMAL(12,2) to allow for higher salaries.**

**ALTER TABLE EMPLOYEES**

**ALTER COLUMN Salary DECIMAL(15, 2);**

**--b. Modify the DateOfBirth column in the Employees table from DATE to DATETIME to include time of birth.**

**SELECT \* FROM EMPLOYEES;**

**ALTER TABLE EMPLOYEES**

**ALTER COLUMN DATEOFBIRTH;**

**--ANAND**

**--2.Add New Columns:**

**--a. Add a new column Email (VARCHAR(100)) to the Employees table to store employee email addresses.**

**alter table employees**

**add column Email VARCHAR(100);**

**--b. Add a new column DepartmentHead (BOOLEAN) to the Departments table to indicate if a department has a head.**

**ALTER TABLE DEPARTMENTS**

**ADD COLUMN DepartmentHead BOOLEAN;**

**--3.Rename Columns:**

**--a. Rename the Major column in the Students table to Course to better reflect the data it stores.**

**CREATE DATABASE SchoolDB;**

**SELECT \* FROM STUDENTS;**

**ALTER TABLE students**

**RENAME COLUMN MAJOR TO COURSE;**

**--b. Rename the Position column in the Employees table to JobTitle.**

**ALTER TABLE EMPLOYEES**

**RENAME COLUMN POSITION TO JOBTITTLE;**

**--4. Remove Columns:**

**--a. Drop the HireDate column from the Employees table as it is no longer needed.**

**ALTER TABLE EMPLOYEES**

**DROP COLUMN HireDate;**

**--b. Remove the DepartmentHead column from the Departments table.**

**SELECT \* FROM DEPARTMENTS;**

**ALTER TABLE DEPARTMENTS**

**DROP COLUMN DEPARTMENTHEAD;**

**--5. Change Default Values:**

**--a. Modify the default value for the Position column in the Employees table to 'Employee' instead of 'Unknown'.**

**SELECT \* FROM EMPLOYEES;**

**UPDATE EMPLOYEES**

**SET JOBTITTLE = 'Employee'**

**where JOBTITTLE = 'Unknown' ;**

**--b. Change the default value for the EnrollmentDate in the Students table to the current date.**

**SELECT \* FROM STUDENTS;**

**UPDATE STUDENTS**

**SET ENROLLMENTDATE = current\_date;**

**--2. Updating Data**

**--1.Basic Updates:**

**--a. Update the Salary of all employees with the Position of 'Junior Developer' to 70000.**

**Update Employees**

**set SALARY = 70000**

**where JOBTITTLE = 'Junior\_Developer';**

**--b. Change the DepartmentName of the department with DepartmentID = 2 to 'Research & Development'.**

**Update Departments**

**Set DEPARTMENTNAME = 'Research & Development'**

**where DepartmentID = 2;**

**--2.Conditional Updates:**

**--a. Update the JobTitle of employees who were hired before 2015-01-01 to 'Senior Developer'.**

**UPDATE EMPLOYEES**

**SET JobTitTle = 'Senior Developer'**

**WHERE HIREDATE < 2015-01-01;**

**--The code is this but it will noit happen since we dropped the ;hiredate' column above.**

**--b. For all students who enrolled before 2020-01-01, change their Course to 'Alumni'.**

**Select \* from students;**

**Update Students**

**set course = 'Alumni'**

**where ENROLLMENTDATE <'2020-01-01';**

**--3.Bulk Updates:**

**--a. Increase the Salary of all employees by 10%.**

**UPDATE employees**

**SET salary = salary + (salary \* 0.10);**

**--b. Set the DepartmentID of all employees currently in the HR department to NULL (assuming the department is being dissolved).**

**Update DEPARTMENTS**

**set DepartmentID = null**

**where departmentname = 'HR';**

**--4.Update Using Joins:**

**--a. Update the Salary of employees in the Sales department to 80000 using a join between Employees and Departments on DepartmentID.**

**select \* from employees;**

**select \* from departments;**

**UPDATE employees**

**--JOIN departments AS d ON e.departmentid = d.departmentid**

**SET salary = 80000**

**WHERE departmentid= 2;**

**--b. Set the DepartmentHead to TRUE for the department that has an employee named 'John Doe'.**

**--anand**

**--3.Advanced Table Modifications**

**--1.Reorganize Table:**

**--a. Change the order of columns in the Employees table to have LastName appear before FirstName.**

**--b. Reorder the Departments table so that DepartmentID is the last column**

**--2.Drop and Add Constraints:**

**--a. Drop the foreign key constraint on DepartmentID in the Employees table, and then add it back with ON DELETE CASCADE.**

**--b. Drop the unique constraint on the combination of FirstName and LastName in the Employees table.**

**--3.Add and Drop Indexes:**

**--a. Add an index on the EnrollmentDate column in the Students table to speed up queries.**

**--b. Drop the composite index on FirstName and LastName in the Employees table.**

**--NEED TO BE DISCUSSED**

**--4.Combined Operations**

**--1. CURRENT\_DATE1.Update and Alter Combined:**

**--a. First, update all employee Salaries to NULL, then modify the Salary column to set a NOT NULL constraint with a default value of 50000.**

**SELECT \* FROM EMPLOYEES;**

**update employees**

**set salary = Null;**

**--ANAND**

**--b. Add a new column PhoneNumber to the Employees table and immediately populate it with a default value for all existing rows.**

**ALTER TABLE EMPLOYEES**

**ADD COLUMN PHONENUMBER INT;**

**--ANAND**

**--2. Modify and Update Combined:**

**--a. 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'.**

**Alter table employees**

**alter column jobtittle varchar(100);**

**update employees**

**set jobtittle = 'Temporary Employee'**

**where jobtittle = 'Intern';**

**--b. Add a column Graduated (BOOLEAN) to the Students table, then update this column to TRUE for all students whose Course is 'Alumni'.**

**Alter table students**

**add column Graduated BOOLEAN;**

**update students**

**set Graduated = 'True'**

**where course = 'Alumni';**

**--Practice Questions on DELETE Commands with Conditions**

**--1.Basic DELETE Operations**

**--1.Delete a Specific Record:**

**--a. Delete the employee from the Employees table where the EmployeeID is 5.**

**DELETE FROM EMPLOYEES**

**WHERE EmployeeID = 5;**

**--b. Remove the student from the Students table whose StudentID is 3.**

**DELETE FROM STUDENTS**

**WHERE STUDENTID = 3;**

**--2.Delete Multiple Records Based on a Condition:**

**--a. Delete all employees from the Employees table who have a Salary less than 50000.**

**DELETE FROM EMPLOYEES**

**WHERE SALARY < 50000;**

**--b. Remove all students from the Students table who enrolled before 2019-01-01.**

**DELETE FROM STUDENTS**

**WHERE ENROLLMENTDATE < '2019-01-01';**

**--2. DELETE with Complex Conditions**

**--1.Delete Using AND/OR Conditions:**

**--a. Delete all employees from the Employees table who are either in the HR department or have a JobTitle of 'Intern'.**

**DELETE FROM EMPLOYEES**

**WHERE JOBTITTLE = 'INTERN' AND JOBTITTLE LIKE '%HR%';**

**--b. Remove all students from the Students table whose Course is 'History' and who enrolled before 2021-01-01.**

**DELETE FROM STUDENTS**

**WHERE COURSE = 'HISTORY' AND**

**ENROLLMENTDATE < '2021-01-01';**

**--2. Delete Using EXISTS Clause:**

**--a. Delete all employees from the Employees table where the department they belong to no longer exists in the Departments table.**

**-- everything present**

**--b. 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**

**--a. Cascade DELETE Operations:**

**--b. Delete a department from the Departments table and ensure all related employees are also deleted (requires foreign key with ON DELETE CASCADE).**

**--c. 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).**

**-- NEED TO BE DISCUSSED**

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

**--1.Delete All Records Based on a Common Attribute:**

**--a. Delete all records from the Employees table where the JobTitle is 'Consultant'.**

**delete from employees**

**where JOBTITTLE = 'Consultant';**

**--b. Remove all records from the Students table where the Course is 'Art'.**

**delete from STUDENTS**

**where COURSE = 'ART';**

**--2. Delete Records with NULL Values:**

**--a. Delete all employees from the Employees table where the Salary is NULL.**

**delete from EMPLOYEES**

**where SALARY = 'NULL';**

**--b. Remove all students from the Students table where the EnrollmentDate is NULL.**

**DELETE FROM STUDENTS**

**WHERE EnrollmentDate IS NULL;**

**--3. Conditional DELETE Using Aggregate Functions:**

**--a. Delete employees from the Employees table who earn less than the average salary of all employees.**

**DELETE FROM EMPLOYEES**

**WHERE SALARY > (SELECT AVG(SALARY) FROM EMPLOYEES);**