

Week-1: Design an ER Diagram

Objective:

The goal of this week's task is to design ER diagrams for various database scenarios, starting with simple systems and progressively moving toward more complex structures. You will identify the entities, their attributes, and the relationships among them, and then represent these visually in an ER diagram.

Task 1: Construct an ER diagram for a Student Enrollment System

Entities:

Student: (sid, sname, ccode, dob, address)

Course: (ccode, cname, fees)

Problem Statement:

Design an ER diagram for a system where students can enroll in courses. Each student has a unique ID (sid), a name (sname), a course code (ccode), date of birth (dob), and an address. Each course has a unique code (ccode), a course name (cname), and associated fees.

Task 2: Construct an ER diagram for a University System

Entities:

Student: (sid, sname, ccode, dob, address)

Course: (ccode, cname, fees)

Department: (did, dname, location)

Problem Statement:

Expand the student enrollment system to include departments. Each student is enrolled in a course, and each course belongs to a department. The department has a unique ID (did), a name (dname), and a location. Create an ER diagram for this scenario.

Task 3: Construct an ER diagram for a University System with Faculty

Entities:

Student: (sid, sname, ccode, dob, address)

Course: (ccode, cname, fees)

Department: (did, dname, location)

Faculty: (fid, fname, sal, designation, doj, did)

Problem Statement:

Now, include faculty in your university system. Faculty members teach courses and belong to departments. Each faculty member has a unique ID (fid), a name (fname), a salary (sal), a designation (designation), a date of joining (doj), and they are associated with a specific department (did). Design an ER diagram for this system.

Task 4: Construct an ER diagram for an Employee-Department System

Entities:

Employee: (eid, ename, salary, doj, comm)

Department: (did, dname, location)

Problem Statement:

Design an ER diagram for a system where employees work in departments. Each employee has a unique ID (eid), a name (ename), salary (salary), date of joining (doj), and commission (comm). Each department has a unique ID (did), a name (dname), and a location. Capture the relationships between employees and departments.

Week-2: Installation of MYSQL

Task 1: MySQL Server and Client Installation

- **Objective:** Install MySQL server and client on your system, ensuring that both the server and client components are set up correctly for communication.
- **Steps:**
 1. Download the MySQL installer from the official website.
 2. Install the MySQL server and client, following the default installation instructions.
 3. After installation, verify that both the server and client are working by connecting to the MySQL server from the MySQL client.
- **Question:** What are the key configuration steps during the installation of MySQL server and client, and how do you verify that the MySQL client can successfully communicate with the MySQL server?

Task 2: MySQL Custom Installation

- **Objective:** Perform a custom installation of MySQL, specifying different installation paths for the server and data directories, and configure additional options such as port number and authentication mode.
- **Steps:**
 1. Download the MySQL installer and select the custom installation option.
 2. Choose custom installation paths for the server, data directories, and other components.

3. Configure additional options such as changing the default port, enabling secure authentication, and adjusting memory usage.
 4. After installation, connect to the server and check that the custom settings have been applied.
- **Question:** How do the options selected during a custom MySQL installation (e.g., installation paths, port changes, and authentication methods) affect the overall functionality and security of the MySQL server?

Week-3: Practicing DDL commands

Objective: To enhance understanding of DDL commands by creating tables, adding constraints, and manipulating table structures.

Task 1: Basic Table Creation Without Constraints

Q1) Write an SQL statement to create a table named Student with the following columns without applying any constraints.:

- StudentID
- FirstName:
- LastName
- DOB

Note: Assume that appropriate data types have been identified for the columns, but no constraints (such as Primary Key, Not Null, or Unique) are applied.

Q2) Write an SQL statement to add a column Address to the above Student table, along with Phone number.

Q3) After creating the table, your task is to Insert several records into the Student table and observe what happens when you enter duplicate or incomplete data.

Task 2: Dropping table and Creating Tables with Constraints

Q4) Write an SQL statement to drop the “student” table which was created in the Task 1.

Q5) Write an SQL statement to create a table named Student with the following columns:

- StudentID
- FirstName
- LastName
- Address
- Phone number
- DOB

- **Note:** Identify appropriate data types and Constraints for the above columns and create the table.

Q6) Create the following tables with a foreign key constraint:

- Departments (Columns: DepartmentID INT PRIMARY KEY, DepartmentName VARCHAR(100))
- Employees (Columns: EmployeeID INT PRIMARY KEY, Name VARCHAR(100), DepartmentID INT referencing Departments)

Task 3: Creating Tables with Default Values and Constraints

Q7) Write an SQL statement to create a table named Orders with the following structure:

- OrderID: Integer, Primary Key
- OrderDate: Date, Default to Current Date
- OrderStatus: Varchar(20), Default value is 'Pending'
- TotalAmount: Decimal, should be greater than 0

Q8) Create a table named Customers with the following structure:

- CustomerID: Integer, Primary Key
- Email: Varchar(100), Unique
- FirstName: Varchar(50), Not Null
- LastName: Varchar(50), Not Null

Task 4: Creating Temporary Tables and AUTO_INCREMENT Columns

Q9) Write an SQL statement to create a temporary table named TempStudents with the same structure as the Student table.

Q10) Create a table named Departments with the following columns:

- DepartmentID: Integer, Primary Key, Auto Increment
- DepartmentName: Varchar(100), Unique

Q11) Create a table named Grades with the following structure:

- GradeID: Integer, Primary Key
- StudentID: Integer, Foreign Key referencing Students(StudentID) with ON DELETE CASCADE
- CourseID: Integer, Foreign Key referencing Courses(CourseID) with ON DELETE CASCADE
- Grade: Char(2)

DBMS_Week-4 (Practicing DML commands)

Objective:

Verification Tasks: After successfully creating the tables using the provided DDL commands, the next step is to verify that the constraints are functioning as expected. Perform the following actions to check if the constraints (Primary Key, Foreign Key, Unique, Not Null, Default, Check, and Auto Increment) are working correctly.

1. Task 1: Insertion into the Student Table

- Insert valid records into the Student table and verify that the Primary Key constraint is enforced by attempting to insert a duplicate StudentID.
- Insert a record without specifying a value for FirstName or LastName and ensure the Not Null constraint prevents this action.

2. Task 2: Check the Courses Table Constraints

- Insert a record into the Courses table with valid CourseID, CourseName, and Credits. Then, try inserting another record with a duplicate CourseName and verify that the Unique constraint prevents the insertion.
- Attempt to insert a record with Credits set to 0 or a negative value and ensure the Check constraint triggers an error.

3. Task 3: Validating Departments and Employees Tables

- Insert valid records into the Departments and Employees tables, ensuring the Foreign Key constraint between Employees and Departments works correctly.
- Try inserting an Employee record with a DepartmentID that does not exist in the Departments table and verify that the Foreign Key constraint prevents the insertion.

4. Task 4: Insertion into the Orders Table

- Insert a record into the Orders table without specifying the OrderDate and OrderStatus values. Verify that the Default values (current date and 'Pending') are applied correctly.
- Attempt to insert a record with TotalAmount set to 0 or a negative value, and ensure the Check constraint prevents the insertion.

5. Task 5: Testing the Customers Table

- Insert records into the Customers table and ensure the Unique constraint on the Email column prevents inserting duplicate email addresses.
- Try inserting a customer without specifying a value for FirstName or LastName and check if the Not Null constraint prevents the operation.

6. Task 6: Auto Increment in Departments Table

- Insert records into the Departments table without specifying DepartmentID and ensure that the Auto Increment feature assigns unique IDs automatically.
- Insert a few more records and verify that the IDs are incrementing as expected.

7. Task 7: Testing ON DELETE CASCADE in Grades Table

- Insert records into the Grades table that reference valid StudentID and CourseID values from the Student and Courses tables, respectively.
- Delete a record from the Student table and verify that the corresponding records in the Grades table are automatically deleted due to the ON DELETE CASCADE constraint.

8. Task 8: Insertion into the Jobs Table

- Insert valid records into the Jobs table, including values for JobTitle, MinSalary, and MaxSalary.
- Verify that the Default value for MinSalary is applied correctly when the field is not explicitly provided during insertion.

Week-5: Practicing DML commands

Objective:

The goal of this task is to practice Data Manipulation Language (DML) commands, which include INSERT, UPDATE, DELETE, and SELECT statements. You will perform various operations on the following tables: Student, Course, Department, Faculty, and Employee.

Task 1: Student and Course Enrollment System

Tables:

Student(sid, sname, ccode, dob, address)

Course(ccode, cname, fees)

Problem Statement:

INSERT 5 students into the Student table, including their student ID, name, course code, date of birth, and address.

INSERT 3 courses into the Course table, with course code, name, and fees.

UPDATE the address of a student identified by sid = 2.

UPDATE the course fees for the course with ccode = 'C101' by increasing it by 10%.

DELETE the record of a student whose sid = 5.

SELECT all the students enrolled in the course ccode = 'C102'.

SELECT the names and fees of all courses where the fees are greater than 2000.

Task 2: University System with Departments

Tables:

Student(sid, sname, ccode, dob, address)

Course(ccode, cname, fees)

Department(did, dname, location)

Problem Statement:

- INSERT 3 departments into the Department table with department ID, name, and location.
- UPDATE the location of the department where did = 1 to 'New York'.
- DELETE the department record with did = 3.
- SELECT all students and display their names along with the department names they belong to (using a join between Student, Course, and Department).
- SELECT the details of courses taught by departments located in 'California'.

Task 3: University System with Faculty

Tables:

Student(sid, sname, ccode, dob, address)

Course(ccode, cname, fees)

Department(did, dname, location)

Faculty(fid, fname, sal, designation, doj, did)

Problem Statement:

- INSERT 4 faculty members into the Faculty table with their ID, name, salary, designation, date of joining, and department ID.
- UPDATE the salary of the faculty member with fid = 2 by 15%.
- DELETE the faculty member with fid = 4.
- SELECT all faculty members who have joined before 2022.
- SELECT the names of faculty members and their department names (using a join between Faculty and Department).
- SELECT the faculty member(s) with the highest salary in each department.

Task 4: Employee and Department System

Tables:

Employee(eid, ename, salary, doj, comm)

Department(did, dname, location)

Problem Statement:

- INSERT 5 employees into the Employee table with their employee ID, name, salary, date of joining, and commission.
- UPDATE the commission of the employee with eid = 3 by 5%.
- DELETE the record of the employee who has been with the company for the longest time (use a subquery to identify the employee with the earliest date of joining).

- SELECT the name and salary of employees who have a salary greater than 5000.
- SELECT the details of employees who work in departments located in 'San Francisco' (use a join between Employee and Department).

Task 5:

Problem Statement:

- Write a query to list the names of the students who are taking courses taught by faculty members with a salary greater than 5000.