Faculty of Technology and Engineering

Chandubhai S. Patel Institute of Technology (CSPIT)

Department of Computer Science & Engineering

Date: / /

Laboratory Manual

Academic Year	:	2024-25	Semester	:	4
Course code	:	CSE206	Course name	:	DATABASE MANAGEMENT SYSTEM

Practical - 7

Aim: As a database administrator for a university, you are tasked with designing and implementing a database schema in the MS Access tool. This database should efficiently manage relationships between departments, courses, students, and their academic performance. To ensure data integrity and consistency, you will define master-slave relationships with appropriate integrity constraints.

maintains the following schemas:

1. Department (Master Table):
   o Stores information about university departments.
   o Attributes: Dept_ID (Primary Key), Dept_Name (Not Null, Unique).
2. Course (Slave Table):
   o Stores courses offered by departments.
   o Attributes: Course_ID (Primary Key), Course_Name (Not Null), Dept_ID (Foreign Key referencing Department).
3. Student (Slave Table):
   o Stores details of enrolled students.
   o Attributes: Student_ID (Primary Key), Student_Name (Not Null), Dept_ID (Foreign Key referencing Department).
4. Enrollment (Slave Table):
   o Tracks student enrollments in courses.
   o Attributes: Enrollment_ID (Primary Key), Student_ID (Foreign Key referencing Student), Course_ID (Foreign Key referencing Course), Grade.

Tasks:-

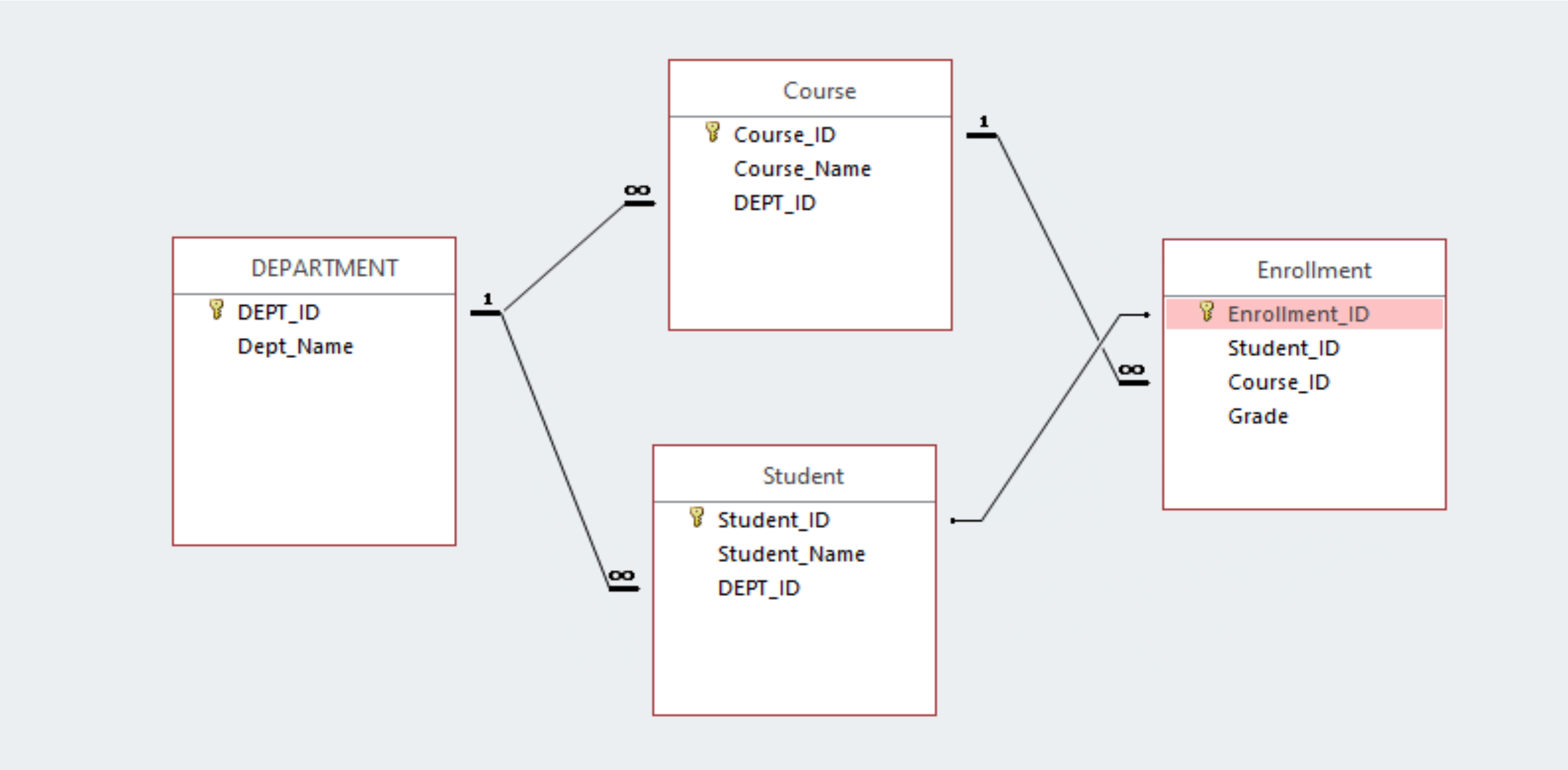
1. Create Master Table (Department):
   o Design a Department table with Dept_ID as the Primary Key.
   o Enforce the following constraints:
   ▪ Dept_Name must be unique.
   ▪ No null values in Dept_Name.
   o Test Case: Verify that duplicate Dept_Name values cannot be inserted.

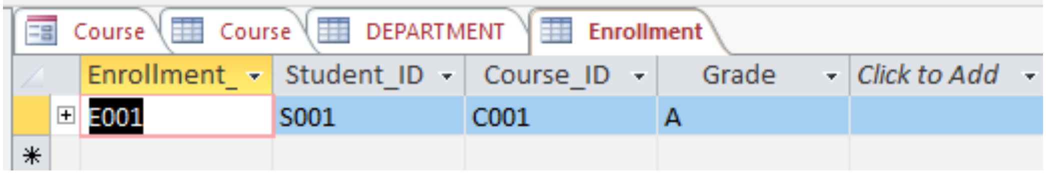
2. Create Slave Table (Course):
   o Design a Course table with Course_ID as the Primary Key.
   o Establish a relationship with the Department table using the Dept_ID foreign key.
   o Enforce referential integrity with the following rules:
   ▪ Cascade updates: If a Dept_ID is updated in the Department table, the corresponding Dept_ID in the Course table should update automatically.
   ▪ Restrict deletions: Prevent deleting a department if courses are linked to it.
   o Test Case: Verify that deleting a department linked to courses is restricted.

3. Create Slave Table (Student):
   o Design a Student table with Student_ID as the Primary Key.
   o Establish a relationship with the Department table using the Dept_ID foreign key.
   o Enforce referential integrity to ensure that each student belongs to a valid department.
   o Test Case: Verify that a student cannot be added with a Dept_ID that does not exist in the Department table
4. Create Slave Table (Enrollment):
   o Design an Enrollment table with Enrollment_ID as the Primary Key.
   o Establish relationships:
   ▪ Student_ID as a foreign key referencing the Student table.
   ▪ Course_ID as a foreign key referencing the Course table.
   o Enforce referential integrity for cascading updates and restricting deletions:
   ▪ Cascade updates: If a Student_ID or Course_ID is updated in their respective tables, the changes should reflect in the Enrollment table.
   ▪ Restrict deletions: Prevent deleting a student or course if enrollment records exist.
   o Test Case: Verify that deleting a student or course linked to enrollments is restricted.

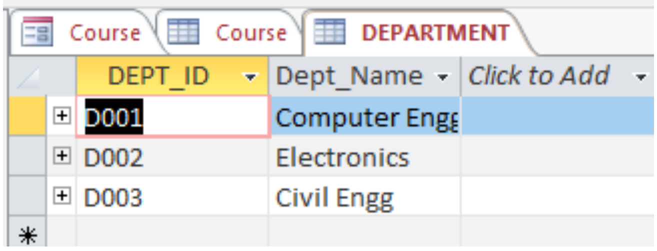
5. Data Validation Rules:
   o Ensure that grades in the Enrollment table only accept valid values (A, B, C, D, F).
   o Test Case: Verify that invalid grades (e.g., E or empty values) cannot be inserted into the Grade field.

6. Data Entry:
   o Populate the tables with sample data for departments, courses, students, and enrollments
   Test Case: Verify that the inserted sample data adheres to all constraints and relationships.

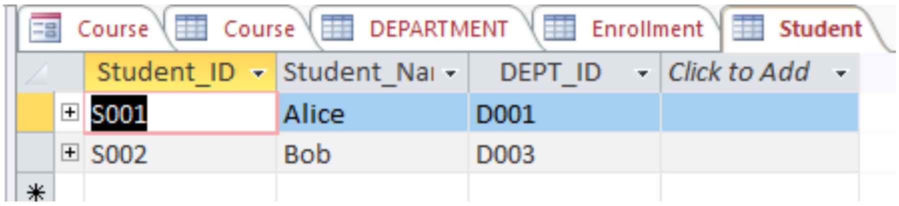




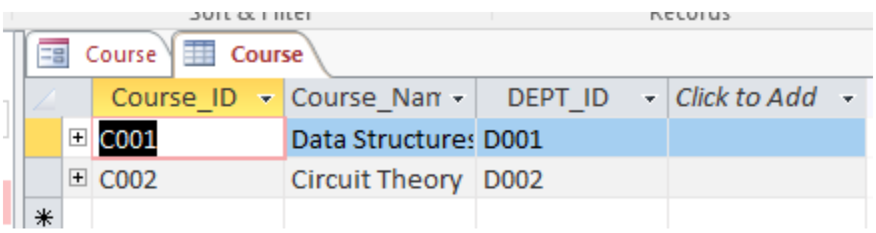
Enrollment_	Student_ID	Course_ID	Grade	Click to Add
E001	S001	C001	A	



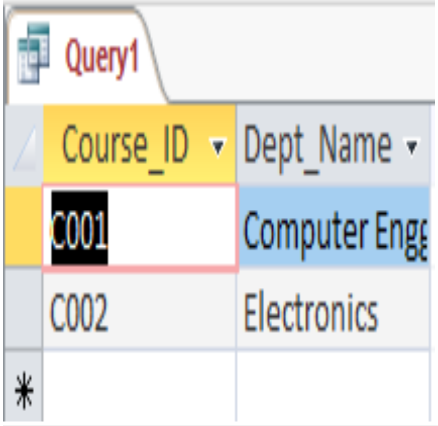
DEPT_ID	Dept_Name	Click to Add
D001	Computer Engg	
D002	Electronics	
D003	Civil Engg	



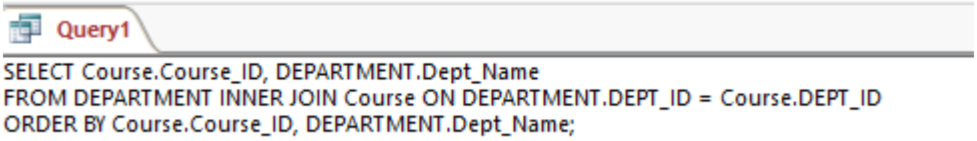
Student_ID	Student_Nai	DEPT_ID	Click to Add
S001	Alice	D001	
S002	Bob	D003	



Course_ID	Course_Nan	DEPT_ID	Click to Add
C001	Data Structure:	D001	
C002	Circuit Theory	D002	



Course_ID	Dept_Name
C001	Computer Engg
C002	Electronics



```
SELECT Course.Course_ID, DEPARTMENT.Dept_Name
FROM DEPARTMENT INNER JOIN Course ON DEPARTMENT.DEPT_ID = Course.DEPT_ID
ORDER BY Course.Course_ID, DEPARTMENT.Dept_Name;
```