Assignment 2

Kumar Rohan

University Database System:Relational Schema Design and Relational Algebra Queries.

Purpose

This assignment requires is to design a relational database schema for a university data management system and implement relational algebra for answering meaningful business questions. This design showcases the understanding of the relational model, primary and foreign keys, integrity rules, and fundamental relational algebra operations as discussed in book Database Systems: Design, Implementation, & Management by Coronel and Morris (2022).

The university database handles information from Departments, Courses, Professors, Students, and Enrollments, including which courses are offered by which departments.

Part 1: Relational Database Schema Design

1. **Entities and Attributes**

   The university database includes the following five entities:

DEPARTMENT

```
DEPARTMENT(
  DepartmentID PK INT,
  DeptName VARCHAR(100) NOT NULL UNIQUE,
  OfficeLocation VARCHAR(50),
  Phone VARCHAR(20)
)

```

Primary Key Justification:

DepartmentID uniquely identifies each academic department.

COURSE

```
COURSE(
  CourseID PK INT,
  CourseName VARCHAR(100) NOT NULL,
  Credits INT,
  DepartmentID FK REFERENCES DEPARTMENT(DepartmentID)
)
```

Relationship:

Each course belongs to one department.

The foreign key enforces that a course cannot exist without a valid department.

PROFESSOR

```
PROFESSOR(
  ProfessorID PK INT,
  ProfName VARCHAR(100) NOT NULL,
  Rank VARCHAR(30),
  HireDate DATE,
  DepartmentID FK REFERENCES DEPARTMENT(DepartmentID)
)
```

Relationship:

A professor works in one department.

STUDENT

```
STUDENT(
  StudentID PK INT,
  FirstName VARCHAR(50),
  LastName VARCHAR(50),
  MajorDepartmentID FK REFERENCES DEPARTMENT(DepartmentID)
)
```

Relationship:

A student majors in one department.

ENROLLMENT

```

ENROLLMENT(
  EnrollmentID PK INT,
  StudentID FK REFERENCES STUDENT(StudentID),
  CourseID FK REFERENCES COURSE(CourseID),
  Semester VARCHAR(20),
  Grade VARCHAR(2)
)
```

Relationship:

This table resolves the many-to-many relationship between students and courses.

2. **Referential Integrity Enforcement**

   Foreign keys apply referential integrity by averting inconsistent data:

- A COURSE cannot be inserted unless its DepartmentID exists in DEPARTMENT.
- An ENROLLMENT cannot make reference a StudentID or CourseID that does not exist.
- A DEPARTMENT cannot be deleted if courses, students, or professors still reference it.

Example:

If attempting to enroll a student in a course that does not in the system, then enrollment will be rejected by the DBMS.

3. **Entity Integrity Discussion**

Entity integrity requires that:

- Each table has a primary key
- Primary keys are unique and NOT NULL

  Example:

  The database will reject inserting a STUDENT record without a StudentID or inserting two students with the same StudentID. This ensures every record can be uniquely identified (Coronel & Morris, 2022).

Part 2: Relational Algebra Queries

Query 1 – SELECT + PROJECT

Business Question:

Find the names of all students majoring in the Computer Science department.

```
π_FirstName, LastName
(σ_DeptName = 'Computer Science'
 (STUDENT ⋈ STUDENT.MajorDepartmentID = DEPARTMENT.DepartmentID
DEPARTMENT))
```

Logic:

Join STUDENT with DEPARTMENT, filter by department name, then project student names

Query 2 – JOIN

Business Question:

List all courses along with their department names.

```
π_CourseName, DeptName
(COURSE ⋈ COURSE.DepartmentID = DEPARTMENT.DepartmentID DEPARTMENT)
```

Query 3 – SELECT

Business Question:

Find all students enrolled in the “Database Systems” course.

```
π_FirstName, LastName
(STUDENT ⋈ STUDENT.StudentID = ENROLLMENT.StudentID
 (σ_CourseName = 'Database Systems'
  (COURSE ⋈ COURSE.CourseID = ENROLLMENT.CourseID ENROLLMENT)))
```

Query 4 – UNION

Business Question:

Find all students who major in Computer Science or Information Systems.

```
A = π_StudentID(σ_DeptName='Computer Science'
   (STUDENT ⋈ STUDENT.MajorDepartmentID = DEPARTMENT.DepartmentID DEPARTMENT))

B = π_StudentID(σ_DeptName='Information Systems'
   (STUDENT ⋈ STUDENT.MajorDepartmentID = DEPARTMENT.DepartmentID DEPARTMENT))
A ∪ B
```

Query 5 – DIFFERENCE

Business Question:

Find students who are not enrolled in any course.

```
π_StudentID(STUDENT) - π_StudentID(ENROLLMENT)
```

Query 6 – JOIN + SELECT

Business Question:

List professors working in the Computer Science department.

```
π_ProfName
(σ_DeptName='Computer Science'
 (PROFESSOR ⋈ PROFESSOR.DepartmentID = DEPARTMENT.DepartmentID DEPARTMENT))
```

Conclusion

This university database design applies relational model principles to real academic operations. Primary and foreign keys enforce data integrity, while relational algebra queries support meaningful reporting such as enrollment analysis and departmental insights. Compared to file-based systems, this approach improves accuracy, consistency, and scalability for long-term university data management.

References

Coronel, C., & Morris, S. (2022). Database systems: Design, implementation, & management (14th ed.). Cengage Learning.