Chapter 05 Relational Database Model

A relational database model is an abstract model that organizes data into a collection of inter-related two-dimensional tables.

1. Entity Relationship Model

Entity Relationship Model (or Entity Relationship Schema) is a model for identifying entities to be represented in the database and representation of how those entities are related.

1.1 Basic Building Blocks

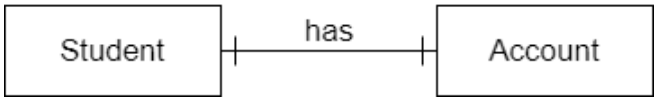
- **Entity**: can be a thing, person, place, object, etc.
- **Attributes**: is a characteristic of an entity.
    - For example, a Customer entity would be described by attributes such as ID, name, address.
- **Relationship**: Describes an association among entities.

1.2 Types of Relationships

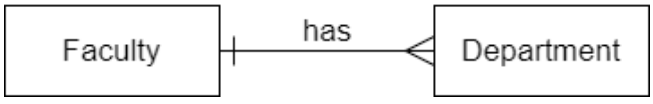
There are three types of relationships that can exist between two entities.

- One-to-one (1:1)
- One-to-many (1:M)
- Many-to-many (M:N or M:M)

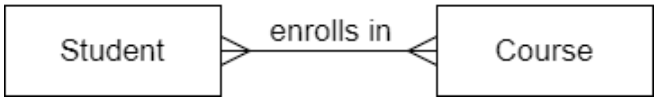
1.2.1 One-to-one Relationships



1.2.2 One-to-many Relationships



1.2.3 Many-to-many Relationships



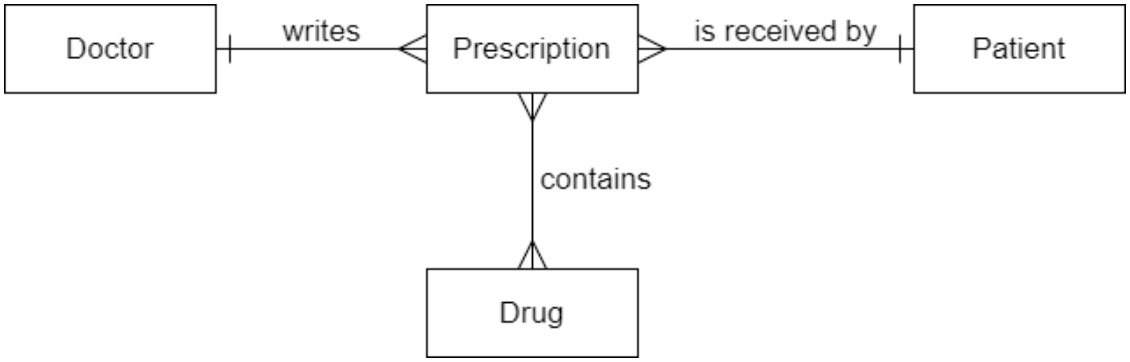
1.3 Naming Conventions and Reading Direction

Entity names should be noun and singular.

Relationship

- Should have a third-person singular verb to describe the relationship.
- Reading direction should be from left-to-right, or from top-to-bottom

Example: An Entity Relationship Diagram (ERD) of a Small Clinic.



Practice

Define the relationship between the entities below:

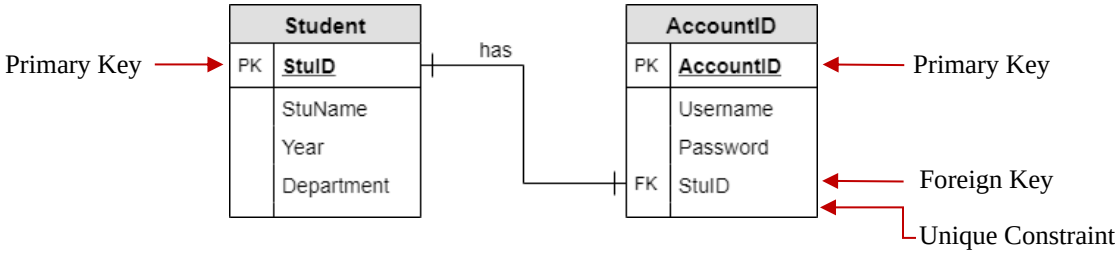
1. Student and University
2. University and Department
3. Teacher and Student
4. Person and House
5. Person and ID Card
6. Person and Facebook Account
7. Homework and Course

2. Relational Model

Relational Model (or Relational Schema) is an implementation model representing a database structure, while an ER model is conceptual or high-level model. Relational model uses tables to show relationships between entities. Relational Models consist of tables (aka. relations), columns (aka. attributes) and rows (aka. records or tuples).

2.1 One-to-one Relationships

Option 1:



In this case, the student record should have existed before we can insert his/her account into table Account.

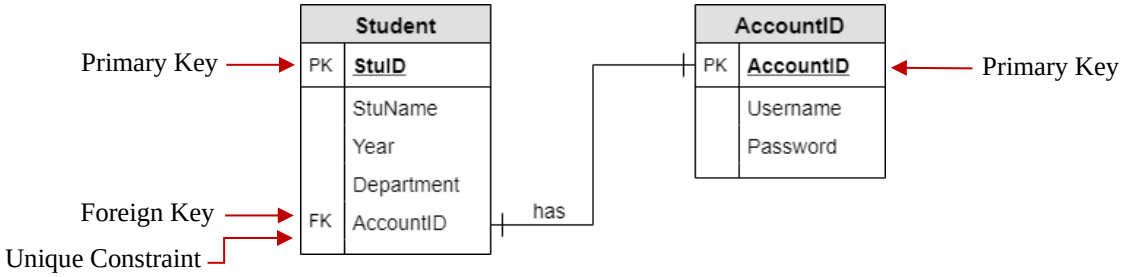
Student

StuID	StuName	Year	Department
1	Lucy	1	CSE
2	Sam	1	TEE
3	Mary	2	BE

Account

AccountID	Username	Password	StuID
1	Lucy	pw01	1
2	Sam	pw02	2
3	abc	123	3

Option 2:



In this case, the account record should have existed before we can insert its owner into table Student.

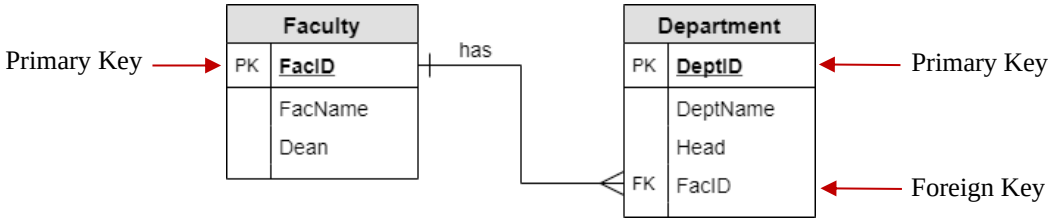
Student

StuID	StuName	Year	Department	AccountID
1	Lucy	1	CS	**1**
2	Sam	1	TEE	**2**
3	Student3	2	ITE	**3**

Account

AccountID	Username	Password
1	Lucy	pw01
2	Sam	pw02
3	Mary	pw03

2.2 One-to-many Relationships



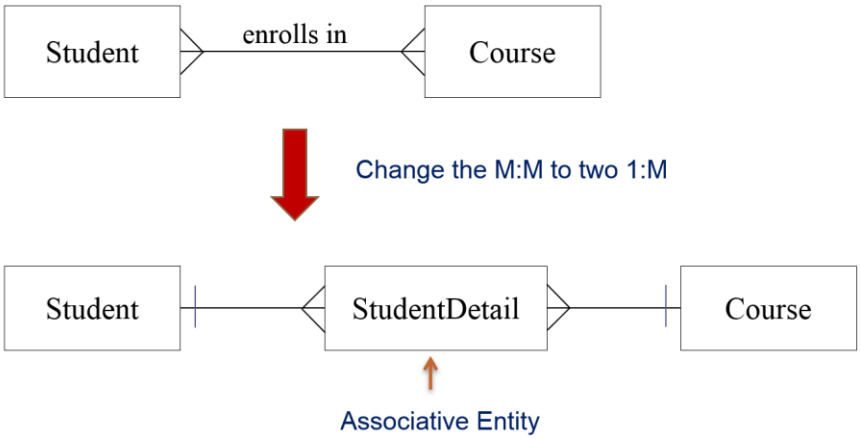
Faculty

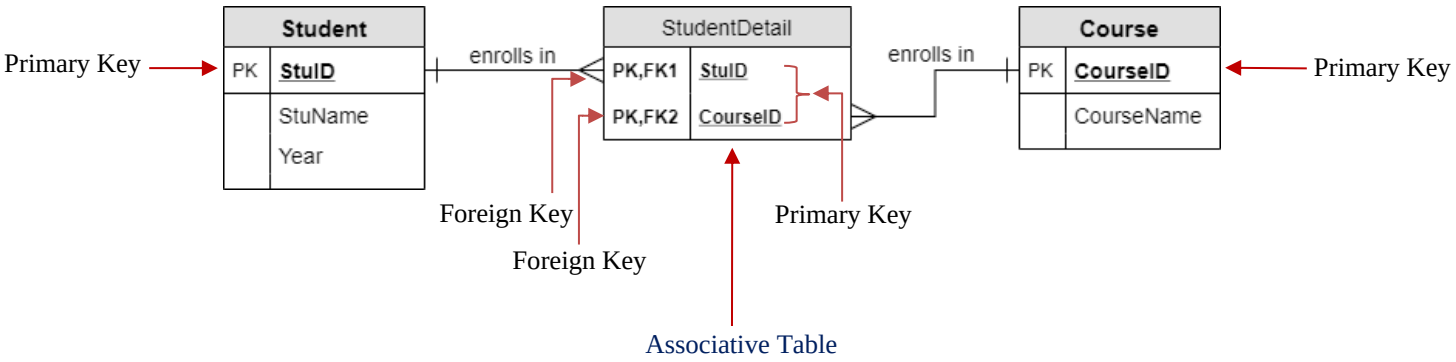
FacID	FacName	Dean
1	FE	Sam
2	Science	Lucy
3	Social	Mary
4	L	Sasa

Department

DeptID	DeptName	Head	FacID
1	EE	Sunny	1
2	TEE	Sokha	1
3	BE	Siteng	2
4	DSE	Sovila	1

2.3 Many-to-many Relationships





Student

StuID	StuName	Year
1	Lucy	1
2	Sam	1
3	Mary	2

StudentDetail

StuID	CourseID
1	1
1	2
2	2
3	2

Course

CourseID	CourseName
1	DB
2	SE
3	Python

Practices

Given the following tables:

Product (ProductID, ProductName, StockQty, UnitCost, UnitPrice, Discount)
Invoice (InvoiceID, InvoiceDate, TotalAmount, PaidAmount, OweAmount)
Customer (CusID, CusName, Gender, Phone, Address)
Staff (StaffID, StaffName, Gender, BirthDate, HiredDate, Position, Salary, Phone, StopWork)
Account (AccountID, Username, Password)

1. Identify the relationships between:
    a) Customer and Invoice
    b) Invoice and Product
    c) Invoice and Staff
    d) Staff and Account
2. Modify the structures of the tables to create relationships between related tables.
3. Draw the Relational Diagram

Exercises

I. Given the tables below:

Faculty (FacultyID, FacultyName, DeanName, OfficeNo)

Department (DeptID, DeptName, HeadName, OfficeNo)

Student (StudentID, StudentName, Gender, DOB, PhoneNo, Address, Year, Generation, Degree)

Teacher (TeacherID, TeacherName, Gender, DOB, PhoneNo, Address)

Course (CourseID, CourseName, Credit, Type)

Account (AccountID, Username, Password, PhoneNo)

1. Identify the relationships.
2. Modify the structures of the tables to create relationships between related tables.
3. Draw the Relational Diagram

Note: you can add new attributes to associative tables if you find them necessary.

II. Write the following programs. Note that you MUST use files to store the data.

1. Manage Faculties
   - CRUD (Create, Read, Update, and Delete) operations.

Menu:
   a. Add a new faculty
   b. Search a faculty by id
   c. Update a faculty
   d. Delete a faculty by id

2. Manage Departments
   - CRUD operations
   - Display all departments belong to a faculty. (Input a faculty's id)

Menu:
   a. Add a new department
   b. Search a department by id
   c. Update a department
   d. Delete a department by id
   e. Display all departments belong to a faculty

3. Manage Students
   - CRUD operations

4. Enroll Students into Departments
   - Enroll a student into a department.
   - Remove a student from a department.
   - Display all student study at given department.
   - Display all department studied by given student

5. Manage Courses
   - CRUD operations

6. Manage Teachers
   - CRUD operations
   - Display all courses taught by a teacher.

7. Assign Courses to Teachers
   - Assign a course to a teacher
   - Remove a course from a teacher
   - Display all courses taught by a teacher.

8. Create Teacher and Student Account
   - Create a teacher account
   - Create a student account

9. Login
   If login is successful, display Hi [Teacher / Student]: [Name].

Reference

[1] Carlos M. Coronel, "Database Systems Design, Implementation, & Management" – 2018