
Fast learning

Case Study 1: Student Course Enrollment

Scenario:

A university maintains records of students and the courses they enroll in. Each student can enroll in multiple courses, and each course can have many students. The university wants to track which student is enrolled in which course, along with the enrollment date.

Questions:

1. What are the main entities involved in this scenario?
2. What is the type of relationship between Student and Course?
3. What extra entity or table is needed to manage the enrollment?

Answers:

1. **Entities:** Student, Course
2. **Relationship:** Many-to-Many
3. **Linking Entity:** Enrollment

Explanation:

Because a student can take multiple courses and a course can have multiple students, it's a **many-to-many** relationship. An **Enrollment** entity helps store additional information like *enrollment date*, *grade*, etc.

Case Study 2: Library Book Borrowing

Scenario:

In a library system, members borrow books. A member can borrow many books, and each book can be borrowed by many different members at different times. The library wants to track when a book was borrowed and when it was returned.

Questions:

1. What are the main entities?
2. What kind of relationship exists between Member and Book?
3. Which entity can store borrow and return dates?

Answers:

1. **Entities:** Member, Book
2. **Relationship:** Many-to-Many
3. **Linking Entity:** Borrow

Explanation:

Since borrowing occurs multiple times, a **Borrow** entity is needed with attributes like *Borrow_Date* and *Return_Date*. This handles the many-to-many relationship.

Case Study 3: Employee Department Assignment

Scenario:

A company assigns each employee to one department. Each department can have many employees. Every department also has one designated manager.

Questions:

1. What is the relationship between Employee and Department?
2. Can an employee be part of more than one department?
3. Where should the manager information be stored?

Answers:

1. **Relationship:** One-to-Many (Department → Employee)
2. **No**, each employee belongs to one department only
3. **Department entity** (as a foreign key to Employee)

Explanation:

A **One-to-Many** relationship exists here. To model the **manager**, you can use a self-linking or foreign key within the department pointing to the employee.

Case Study 4: Online Store Orders

Scenario:

An online store system tracks customers, their orders, and the products included in those orders. One order can include multiple products, and the same product can appear in many orders.

Questions:

1. What are the main entities?
2. What is the relationship between Order and Product?

3. What entity is needed to manage this many-to-many relationship?

Answers:

1. **Entities:** Customer, Order, Product
2. **Relationship:** Many-to-Many (Order–Product)
3. **Linking Entity:** OrderDetails (or LineItem)

Explanation:

OrderDetails stores extra info like *Quantity*, *Unit Price*, and forms a bridge between orders and products.

Case Study 5: Doctor-Patient Consultation

Scenario:

Hospitals keep track of which doctors consult which patients. A doctor can consult multiple patients, and each patient can visit different doctors. Every consultation has a date and diagnosis.

Questions:

1. What entities are involved?
2. What relationship exists between Doctor and Patient?
3. Where should consultation details like date and diagnosis be stored?

Answers:

1. **Entities:** Doctor, Patient
2. **Relationship:** Many-to-Many
3. **Entity:** Consultation

Explanation:

The **Consultation** entity links doctors and patients with fields such as *Consult_Date*, *Diagnosis*, and *Prescription*.

Case Study 6: College Hostel Room Allocation

Scenario:

In a college hostel, each student is allotted one room. A room can be shared by two students at most. Each room has a unique number and a warden assigned.

Questions:

1. What are the main entities?

2. What is the relationship between Student and Room?
3. Where is warden info stored?

Answers:

1. **Entities:** Student, Room, Warden
2. **Relationship:** One-to-Many (Room to Student)
3. **Warden Info:** In the Room entity or as a separate Warden entity with relationship

Explanation:

If a room has a warden, the room can have a foreign key to a **Warden** entity. Student-to-Room is **Many-to-One** (or One-to-Many from Room side).

Case Study 7: Movie Booking System

Scenario:

A movie ticket booking system tracks users who book tickets for movies. A user can book multiple movies, and each movie can be booked by multiple users. Each booking stores seat number and show timing.

Questions:

1. Identify the main entities.
2. What is the relationship between User and Movie?
3. What entity stores seat number and timing?

Answers:

1. **Entities:** User, Movie
2. **Relationship:** Many-to-Many
3. **Entity:** Booking

Explanation:

The **Booking** entity links the user and movie with details like *Seat Number, Show Time, Booking Date*.

Moderate Types

Case Study 1: University Research Project Management

Scenario:

In a university, multiple professors supervise research projects. Each professor can guide several projects, and more than one professor can supervise each project. Students work on projects, and each student is assigned to only one project.

Questions:

1. Identify all entities and relationships.
2. What kind of relationship exists between the Professor and the Project?
3. Can you identify any weak entity?

Answers:

1. **Entities:** Professor, Project, Student
Relationships: Supervises (Professor–Project), Works_On (Student–Project)
2. **Many-to-Many** between Professor and Project
3. A **student** can be treated as a weak entity (if the project determines student participation)

Explanation:

Because professors can co-supervise and projects can have multiple mentors, it's a many-to-many relationship. Students are linked one-to-one with a project.

Case Study 2: Airline Reservation System

Scenario:

Passengers book flights through an airline reservation system. Each flight has a unique number and may carry multiple passengers. Each booking stores the seat number and booking status.

Questions:

1. List all entities and their key attributes.
2. What type of relationship exists between a Passenger and a Flight?
3. What entity stores the booking status and seat number?

Answers:

1. **Entities:** Passenger (Passenger_ID), Flight (Flight_ID), Booking
2. **Many-to-Many**
3. **Booking** entity stores seat number and status

Explanation:

The Booking entity links passengers and flights, and tracks seat numbers and whether a booking is confirmed or waitlisted.

Case Study 3: Hospital Staff and Department Management

Scenario:

A hospital has departments, and each department employs multiple staff members (doctors, nurses, etc.). A staff member can work in more than one department. Staff are also assigned shifts.

Questions:

1. Which entities are involved in this model?
2. What relationship exists between Staff and Department?
3. How can shift details be modeled?

Answers:

1. **Entities:** Staff, Department, Shift
2. **Many-to-Many** between Staff and Department
3. Create a **Staff_Shift** entity with attributes like Shift_Date and Shift_Time

Explanation:

Because staff can work in multiple departments and shifts, we use many-to-many relationships and a bridge table for shift assignment.

Case Study 4: Event Management System

Scenario:

An event management company organizes multiple events. Clients request events, and each event can have several assigned employees (like decorators, technicians). Events are scheduled at different venues.

Questions:

1. What are the primary entities and attributes?
2. What is the relationship between Event and Employee?
3. How would you track venue information?

Answers:

1. **Entities:** Event, Client, Employee, Venue
2. **Many-to-Many** between Event and Employee
3. Venue is either an entity or an attribute of Event

Explanation:

Each event has a venue and a team. A many-to-many link between Event and Employee with an assignment table tracks who works on what.

Case Study 5: Vehicle Service Center

Scenario:

A vehicle service center manages customer vehicle servicing. Each customer may own multiple vehicles. A vehicle can undergo many service appointments. Each service appointment records service date and cost.

Questions:

1. Identify the main entities and relationships.
2. What kind of relationship exists between Vehicle and ServiceAppointment?
3. What attribute belongs to ServiceAppointment?

Answers:

1. **Entities:** Customer, Vehicle, ServiceAppointment
2. **One-to-Many** from Vehicle to ServiceAppointment
3. *Service_Date, Cost, Service_Details*

Explanation:

Each customer owns one or more vehicles, and each vehicle can have several service entries. ServiceAppointment is connected directly to Vehicle.

Slow Learning

Case Study 1: Library System (Basic)

Scenario:

A library has many books. Each book has a unique ID and a title. Students can borrow books. One student can borrow many books, but each book is borrowed by only one student at a time.

Questions:

1. Name the entities.
2. What is the relationship between Student and Book?
3. Is it one-to-many or many-to-one?

Answers:

1. **Entities:** Student, Book
2. **Borrows** relationship
3. **One-to-Many** (one student → many books)

Explanation:

Books are borrowed by students. The student ID is linked to several books.

Case Study 2: School and Teachers**Scenario:**

In a school, teachers are assigned to subjects. Each teacher teaches only one subject, and each subject is taught by one teacher.

Questions:

1. List the entities.
2. What type of relationship exists between Teacher and Subject?

Answers:

1. **Entities:** Teacher, Subject
2. **One-to-One**

Explanation:

Simple 1-to-1 mapping—each subject has one teacher and vice versa.

Case Study 3: Mobile Store**Scenario:**

A mobile store sells different mobile phones. Each phone has a model number and price. Customers buy one or more mobile phones.

Questions:

1. Name the entities.
2. What is the relationship between Customer and Mobile?

Answers:

1. **Entities:** Customer, Mobile
2. **One-to-Many** (One customer can buy multiple mobiles)

Explanation:

Each customer is linked with multiple mobile purchases.

Case Study 4: Online Course Enrollment**Scenario:**

Students can enroll in multiple courses. Each course has a course ID and name. A course can also have many students.

Questions:

1. What are the main entities?
2. What is the relationship type?

Answers:

1. **Entities:** Student, Course
2. **Many-to-Many**

Explanation:

Since students enroll in many courses and each course has many students, it's many-to-many.

Case Study 5: Bank and Accounts

Scenario:

A bank has many customers. Each customer can have multiple accounts. Each account has a number and a balance.

Questions:

1. Name the entities.
2. What is the relationship between Customer and Account?

Answers:

1. **Entities:** Customer, Account
2. **One-to-Many**

Explanation:

A customer may own multiple bank accounts.