TD2 : Solution

Data Organization

Exercise 1: Organize Customer Data

Task: Convert this messy data into a structured table:

- Ahmed Nasri, $50, credit card, purchased shoes, 2024-11-20
- Aya bentaj, $70, cash, 2024-11-21, purchased a bag,

Solution:

Customer Name	Item Purchased	Price	Payment Method	Date
Ahmed Nasri	Shoes	$50	Credit Card	2024-11-20
Aya bentaj	Bag	$70	Cash	2024-11-21

Exercise 2: Split Information into Categories

Task: A company has employees with the following data:

- Name, Department, Salary, Hire Date

1. Create a table with appropriate columns.
2. Add the following rows:
    - Amine , 2020-05-15, HR, $60,000
    - $70,000, Linda, 2019-03-12, Marketing

Solution:

Name	Department	Salary	Hire Date
Amine	HR	$60,000	2020-05-15
Linda	Marketing	$70,000	2019-03-12

Relational Thinking

Exercise 1: Identify Relationships

Task: For a university database:

1. Identify the tables for **Students**, **Courses**, and **Enrollments**.
2. Specify how these tables are related.

Solution:

1. **Tables**:
    - *Students*: Student ID, Name, Major
    - *Courses*: Course ID, Title, Instructor
    - *Enrollments*: Enrollment ID, Student ID, Course ID, Grade
2. **Relationships**:
    - The *Students* table connects to the *Enrollments* table using **Student ID**.
    - The *Courses* table connects to the *Enrollments* table using **Course ID**.

Exercise 2: Design a Simple Database

Task: You manage a bookstore. Create tables and their relationships.

1. Books (ISBN, Title, Price)
2. Customers (Customer ID, Name, Email)
3. Sales (Sale ID, Customer ID, ISBN, Sale Date)

Scenario: Cinema Management System

You are tasked with designing a database for a **cinema management system** that tracks movies, screenings, customers, and bookings. Below is a textual description of the features and relationships in the system. Your job is to extract the tables, define the features for each table, and establish relationships with their cardinalities.

Description of Dataset 1: Movies

The cinema shows a variety of movies. For each movie, the following details are recorded:

1. A unique identifier for the movie.
2. The title of the movie.
3. The genre (e.g., Action, Comedy, Drama).
4. The duration of the movie (in minutes).
5. The release date.

Description of Dataset 2: Screenings

The cinema schedules screenings for different movies. For each screening, the following details are recorded:

1. A unique identifier for the screening.
2. The movie being screened.
3. The date and time of the screening.
4. The screen number where the movie is shown.
5. The total number of available seats in the screen.

Description of Dataset 3: Customers

The cinema allows customers to book tickets online or in person. For each customer, the following details are recorded:

1. A unique identifier for the customer.
2. The full name of the customer.
3. The customer's phone number.
4. The list of tickets the customer has booked.

Description of Dataset 4: Bookings

For every ticket booked by a customer, the cinema records the following details:

1. A unique identifier for the booking.
2. The customer who made the booking.
3. The screening for which the ticket was booked.
4. The number of seats booked by the customer.
5. The total price of the booking.

Task for Students

Task 1: Extract Tables and Features

Using the descriptions above, identify the tables and define the fields (columns) for each table. Write the structure of the tables with their respective features.

Task 2: Define Relationships

Analyze the relationships between the extracted tables. Identify how the entities are connected and specify the cardinalities (e.g., one-to-many, many-to-many).

Example Output (What students should produce)

Tables

1. **Movie Table**
    - Movie ID (Primary Key)
    - Title
    - Genre
    - Duration
    - Release Date
2. **Screening Table**
    - Screening ID (Primary Key)
    - Movie ID (Foreign Key linking to the Movie Table)
    - Date and Time
    - Screen Number
    - Available Seats
3. **Customer Table**
    - Customer ID (Primary Key)
    - Full Name
    - Phone Number

4. **Booking Table**
    - Booking ID (Primary Key)
    - Customer ID (Foreign Key linking to the Customer Table)
    - Screening ID (Foreign Key linking to the Screening Table)
    - Number of Seats
    - Total Price

Relationships and Cardinalities

1. **Movie ↔ Screenings**
    - A **movie** can have multiple **screenings**: **1:N**.
    - A **screening** is associated with exactly one **movie**: **N:1**.
2. **Customer ↔ Bookings**
    - A **customer** can make multiple **bookings**: **1:N**.
    - A **booking** belongs to exactly one **customer**: **N:1**.
3. **Screenings ↔ Bookings**
    - A **screening** can have multiple **bookings**: **1:N**.
    - A **booking** belongs to exactly one **screening**: **N:1**.

Scenario: E-Commerce Store Analysis

You are tasked with organizing and analyzing data for an **e-commerce store** that tracks orders, customers, products, and feedback. Below is a textual description of the features and relationships within the datasets. Your job is to extract the tables, define the features for each table, and establish relationships with their cardinalities.

Description

The store records orders placed by customers. Each order contains the following details:

1. A unique identifier for each order.
2. The customer who placed the order.
3. The product(s) included in the order.
4. The quantity of each product in the order.
5. The date when the order was placed.

6. The total price of the order.

The store also collects customer feedback about the products they purchased. The feedback includes the following details:

1. A unique identifier for each feedback entry.
2. The customer who left the feedback.
3. The product being reviewed.
4. A numerical rating between 1 and 5.
5. A textual comment or review provided by the customer.
6. The date when the feedback was submitted.

You are tasked to create Customer and product details

Task for Students

Task 1: Extract Tables and Features

From the description above, identify the tables and their corresponding features. Write the structure of each table, specifying the fields (columns).

Task 2: Define Relationships

Analyze the relationships between the extracted tables. Identify the cardinalities (e.g., one-to-many, many-to-many) between the entities.

Example Output (What students should produce)

Tables

1. **Orders Table**
    - Order ID (Primary Key)
    - Customer ID (Foreign Key linking to the Customer Table)
    - Product ID (Foreign Key linking to the Product Table)
    - Quantity
    - Order Date
    - Total Price
2. **Feedback Table**

- Feedback ID (Primary Key)
- Customer ID (Foreign Key linking to the Customer Table)
- Product ID (Foreign Key linking to the Product Table)
- Rating
- Feedback Text
- Feedback Date

Relationships and Cardinalities

1. **Customer ↔ Orders**
   - A **customer** can place multiple **orders** (1:N).
2. **Customer ↔ Feedback**
   - A **customer** can leave multiple **feedback entries** (1:N).
3. **Product ↔ Orders**
   - A **product** can appear in many **orders** (1:N).
4. **Product ↔ Feedback**
   - A **product** can receive multiple **feedback entries** (1:N).

Scenario: University Management System

You are tasked with designing a data structure for a **university management system** that tracks students, courses, and instructors. Below is a textual description of the features and relationships in the system. Your job is to extract the tables, define the features for each table, and establish relationships with their cardinalities.

Description of Dataset 1: Student Information

The university keeps track of all its students. For each student, the following details are recorded:

1. A unique identifier for the student.
2. The full name of the student.
3. The program the student is enrolled in (e.g., Computer Science, Business).
4. The student's enrollment year.
5. The list of courses the student is enrolled in.

Description of Dataset 2: Course Information

The university offers multiple courses. For each course, the following details are recorded:

1. A unique course code.
2. The name of the course.
3. The number of credit hours for the course.
4. The instructor responsible for the course.
5. The list of students enrolled in the course.

Description of Dataset 3: Instructor Information

The university also keeps track of its instructors. For each instructor, the following details are recorded:

1. A unique identifier for the instructor.
2. The full name of the instructor.
3. The department the instructor belongs to.
4. The list of courses taught by the instructor.

Task for Students

Task 1: Extract Tables and Features

Using the descriptions above, identify the tables and define the fields (columns) for each table. Write the structure of the tables with their respective features.

Task 2: Define Relationships

Analyze the relationships between the extracted tables. Identify how the entities are connected and specify the cardinalities (e.g., one-to-many, many-to-many).

Example Output (What students should produce)

Tables

1. **Student Table**
    - Student ID (Primary Key)
    - Full Name
    - Program
    - Enrollment Year
2. **Course Table**

- Course Code (Primary Key)
- Course Name
- Credit Hours
- Instructor ID (Foreign Key linking to the Instructor Table)

3. **Instructor Table**
    - Instructor ID (Primary Key)
    - Full Name
    - Department
4. **Enrollment Table** *(for managing many-to-many relationships between students and courses)*
    - Enrollment ID (Primary Key)
    - Student ID (Foreign Key linking to the Student Table)
    - Course Code (Foreign Key linking to the Course Table)

Relationships and Cardinalities

1. **Student ↔ Courses (via Enrollment Table)**
    - A **student** can enroll in multiple **courses**: **1:N** through the Enrollment Table.
    - A **course** can have multiple **students** enrolled: **1:N** through the Enrollment Table.
2. **Instructor ↔ Courses**
    - An **instructor** can teach multiple **courses**: **1:N**.
    - A **course** is taught by one **instructor**: **N:1**.
3. **Student ↔ Enrollment ↔ Courses**
    - The **Enrollment Table** creates a many-to-many relationship between **students** and **courses**.