

TUTORIAL 1: Creating a data model

Introduction

In this module you will learn how to develop a relational database based on a model designed with UML.

UML diagrams can model the **use cases** of a system and the **classes** representing the entities in the system. The domain classes in a class diagram represent a **data model** which will support the use cases. The data model will in turn form the basis for implementing the database.

In this tutorial you will think about the use cases and data model for a system. In the labs you will go on to develop a working database for this data model.

Scenario



GCUCars is a car rental company. They require a new computer system to manage their business. You have interviewed their managing director to find out how they expect the system to work. Here is a transcript of part of the interview:

Q. What happens when a customer phones up to hire a car?

A. The CSR (that's customer service representative, by the way) asks what the customer is looking for – when the car is needed and for how long, what kind of car, which of our locations it needs to be picked up from, and so on.

Q. Can the customer choose a specific make and model?

A. No, we have different categories of car, like compact hatchbacks, family cars, SUVs and so on. All cars in the same category cost the same to hire, but you could get different makes depending on what's available. For example, if you ask for an estate car you might get a Ford Mondeo or a Vauxhall Vectra, or whatever.

Q. Do cars have to be returned to the same location they are hired from?

A. No, a car can be returned to a different location, and then it becomes available for hire from that location.

Q. How are hire charges worked out?

A. There is a daily charge, and there is a charge for excess mileage. The CDW is optional.

Q. CDW?

A. Collision Damage Waiver – it's an extra charge for comprehensive insurance, and it's also charged per day.

Q. So, the CSR will ask the system to find a suitable car?

A. Yes, and if there is one available, then the CSR tells the customer how much it costs and offers the CDW option.

Q. And if the customer agrees?

A. The CSR asks if the customer has hired from us before, and looks up his or her details if so. If not, the customer's name, address and so on need to be taken and an account needs to be set up. The next time that customer comes back to us their details should then be in the system. The CSR might want to look at the customer's previous hires – the CSR is allowed to give a discount to a good customer to try to clinch the deal. The allocated car is then not available for hire until it is returned.

Q. So what happens when a car is returned?

A. Usually the customer pays in full by credit card, but sometimes the customer is sent an invoice later on. The hire is marked as complete and the car becomes available again at the location it was returned to.

Q. Is anyone else involved in your operations?

A. Well, the fleet manager needs to get a list of all the cars and their locations so they can schedule services. Oh, and the finance staff need to get a list of customer accounts which have not been paid in full so that they can follow up on them

Q. What about your staff? Does each employee work at the same location all the time?

A. Not always. Some employees are based at two or three locations and their time is split between those particular locations.

Task

Think about the requirements described in the interview. Now try to identify, and draw diagrams to describe, the following:

- **use cases** for the system
example: finance staff gets all unpaid accounts
- **entities** in the system (think of the type of information you think would need to be stored permanently)
example: customer, account
- **attributes** of the entities
example: a customer will have a name, address, telephone number, and so on.
- **relationships** between the entities
example: a customer will have one account

(To keep things from getting too complicated, assume for now that the system only deals with *immediate* hires, so it doesn't have to track where cars are going to be in the future.)