Advanced Data Management (CMM524)

Solution/Discussion to Laboratory #1: Data Modelling

1. Aims

- To experience data modelling in a domain.

2. Outcomes

In completing this exercise, you should be able to:

- Model the data requirements of a given domain as an Entity Relationship (ER) model.
- Map an ER model into the Relational Data Model as tables.

3. Data Modelling

Storing structured data in a database management systems requires modelling the data requirement as a database schema. This process involves the design of a conceptual model using ER model, then mapping the ER model into relational tables.

3.1. The "Online Shop" Domain

"An online shop sells products to customers.

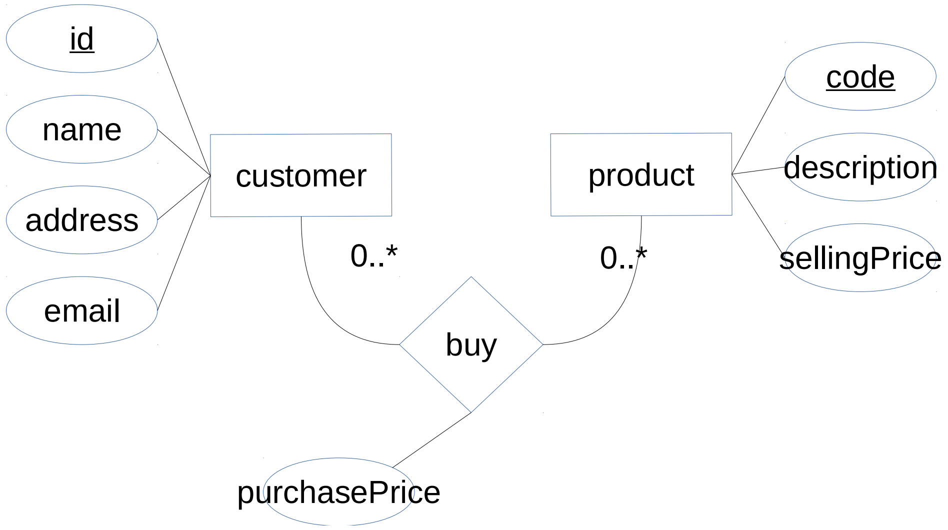
Each product is identified by its unique product code/ID which is a number entered by the online shop. A product also has a textual description and a current selling price.

Each customer has an internal ID which is allocated by the system. The system also needs to store the name, postal address, and email address of a customer.

A customer can buy multiple products in an order. As the selling price of a product may change, an order must keep the price of a product at the time of ordering."

- Design an ER model to capture data in the domain.
    - You only need to model the 3 concepts of product, customer and buying described above.
        - We are only interested in these 3 concepts. When you design the ER model, you include details of these 3 but ignore others irrelevant concepts.
    - What entities do you end up with?
        - You should end up with 2 entities: product and customer. The names may be different though.
    - What relationships do you have?
        - You should have 1 relationship "`buy`" linking customer and product.

        - Note that “purchasePrice” is an attribute of the “buy” relationship as it is specific to a buying relationship instance. e.g. Even if you buy the same product again, the price you pay may be different as every purchase is a different “relationship instance”.



- Map your ER model design into relational tables.
    - You may end up with the following tables.

Customer table				
Column	Type	NULL	Key	Extra
id	int	no	primary key	
name	varchar(255)	no		
address	varchar(255)	no		
email	varchar(255)	no		

Product table				
Column	Type	NULL	Key	Extra
code	int	no	primary key	
description	varchar(255)	no		
sellingPrice	float	no		default 0.0

Buy table				
Column	Type	NULL	Key	Extra
customerID	int	no	key is (customerID,productCode)	
productCode	int	no		
purchasePrice	float	no		default 0.0

    - How many tables do you have?
        - Each entity is mapped into 1 table. The relationship (with attribute) is also mapped into 1. You should have 3 tables.

- What is the key in each table?
    - The key in each table is the set of attributes which is enough to uniquely identified a single row.

3.2. The “Hospital” Domain

The followings describe the requirements of a database storing patient records in a hospital:

“In a hospital, a patient is admitted to a ward on the admission date. There will be a discharge date when the patient leaves.

A patient will be give a bed on admission.

Throughout his/her lifetime, a patient may be admitted several times (and potentially to different wards).

Each patient has a patient number which uniquely identifies his/her records in the hospital.

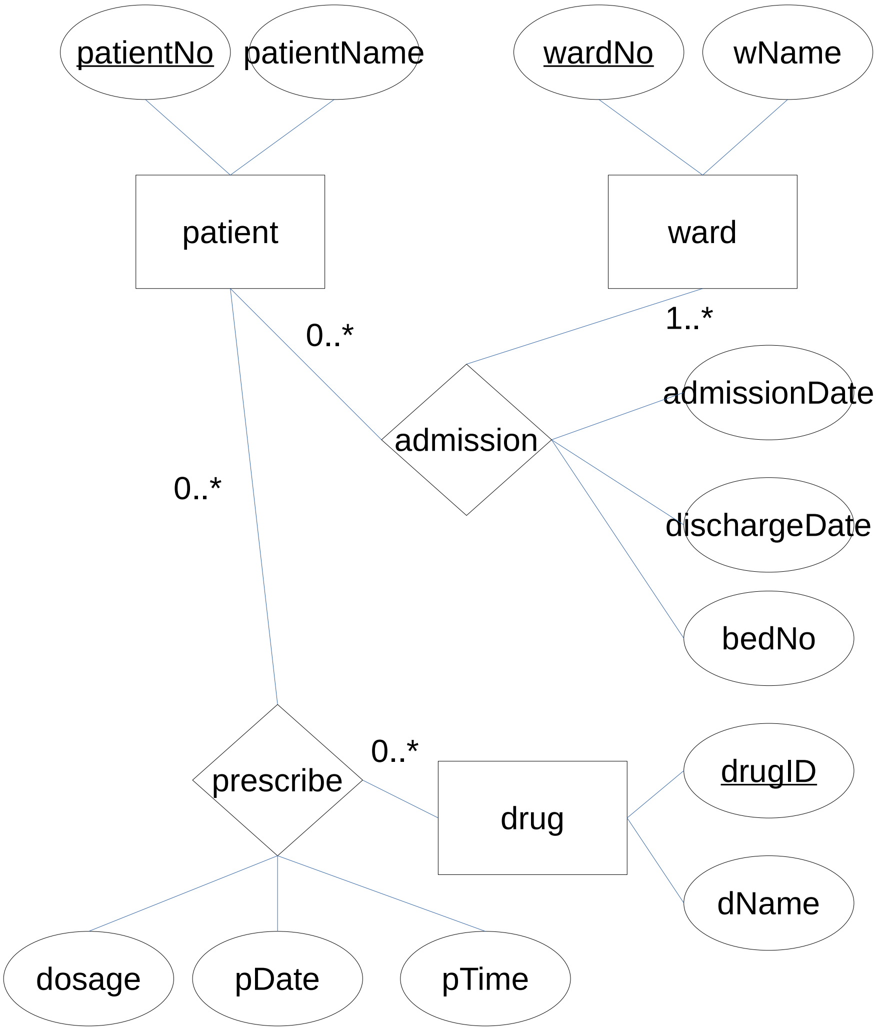
Each ward has a unique ward number, and a name.

While the patient is in hospital, he/she may be prescribed with drugs on a certain date and time. Note that multiple different drugs can be prescribed at the same time.

Each drug has a unique drug ID, and a name.

Each prescription has a dosage.”

- Design an ER model to capture data in the domain.
- Map your ER model design into relational tables.



Notes:

- Some of you may want to include "doctor" as an entity. This is completely fine if you want to know who is the doctor prescribing a drug.
- There is no foreign key in an ER model. I have said this many times. See the lecture slides.
- See how "`admission`" is modelled as a relationship between "`patient`" and "`ward`". This is based on my understanding. Yours can be different.
    - "`admissionDate`", "`dischargeDate`" and "`bedNo`" are all attributes of the "`admission`" relationship as they are attached to an instance of "`admission`". Try to attach them to either "`patient`" or "`ward`". They won't fit as every time a patient is admitted to a ward, these attribute values may change.
    - See that 1 patient can be in "1 or many" wards. It cannot be 0 ward. It can be multiple wards as the patient may have been admitted to this hospital before, in different wards.
    - A ward can have "0 or many" patients.

- Similarly, I model “`prescribe`” (or “`prescription`”) as a relationship between “`patient`” and “`drug`”. You can model “`prescription`” as an entity if you wish. We will then see how the final schema may differ.
  - Again, see how I model “`dosage`”, “`pDate`”, and “`pTime`” as attributes of the “`prescribe`” relationship.
  - As mentioned above, I didn’t model “doctor”. If you do have a “doctor”, it can be an attribute of this “prescribe” relationship, or the “doctor” entity can take part in this relationship (It will then be a N-ary relationship).
  - A patient can be related to “0 or many drugs”.
  - A drug can be prescribed to “0 or many patients”.

From the ER model, we can mapped it into tables.

First we map entities into tables:

Table: patient	
patientNo	varchar(16)
patientName	varchar(255)

Table: ward	
wardNo	int
wName	varchar(32)

Table: drug	
drugID	varchar(32)
dName	varchar(64)

Notes:

- Your chosen data types for the attributes may differ. e.g. I assume `patientNo` is a max-16-character string. You may want it to be an int, or a string of different length.

Next we map the “`admission`” relationship to a table:

Table: admission	
patientNo	varchar(16)
wardNo	int
admissionDate	datetime
dischargeDate	datetime
bedNo	int

Notes:

- This table has a composite primary key `(patientNo,admissionDate).`
  - Knowing the patientNo+wardNo is not enough to know which admission instance it is, as a patient can be admitted to the same ward on different occasions (i.e. dates)

    - We assume a patient cannot be admitted to the same ward more than once on 1 day. If this is not true, you will need to modify the primary key.
    - We also assume a patient cannot be admitted to different wards on the same day. Otherwise you will need to include “wardNo” in the key.

Similarly, we map the “`prescribe`” relationship into a table:

Table: prescribe	
patientNo	varchar(16)
drugID	varchar(32)
pDate	datetime
pTime	datetime
dosage	int

Notes:

- This table has a composite primary key `(patientNo,drugID,pDate,pTime).`
    - If we assume a patient can be prescribed the same drug multiple times on the same day, we need to know the different occasions. That’s why “`pTime`” is part of the key.
    - “`dosage`” can be a `float`. It is a numeric value, although we did not specify the unit.