EXERCISE LIST N°6:

More normalization

-

Database modelling using ER diagrams and top-down database design

Exercise 1: More normalization through warehouse storage

We want to build a database to store information about warehouses containing automobile parts.
A warehouse has one manager who has an employee number (unique) and a name.
A warehouse is identified by its warehouse name (unique) and its address.
Every part in the warehouse has a part number (unique), inventory date, quantity-on-hand, supplier name (at most one supplier per part).

Then, the parts are delivered from the warehouse to some place (destination). The deliveries are identified by their delivery number, delivery date, and quantity for each part delivered. In one delivery, we might send several parts (some of them, with delivered quantity > 1).

If we put all these attributes into just 1 big relation, then we get the following relation:
WAREHOUSE (manager-id, manager-name, warehouse-name, warehouse-address, part-no, inventory-date, qty-on-hand, supplier-name, delivery-no, delivery-date, delivery_destination, delivery-qty)

1. Modify this schema so that its relations are in 3NF.
2. Draw the ER diagram of the normalized schema (using UML notations).

Exercise 2: Even more normalization through books

I consider the following schema for a library (where PK denotes Primary Keys, and FK Foreign Keys). Each book (title) corresponds to at most one domain (sci-fi, novel…).

AUTHOR table

- Author_ID, PK
- First_Name
- Last_Name

TITLES table

- TITLE_ID, PK
- NAME
- Author_ID, FK

DOMAIN table

- DOMAIN_ID, PK
- NAME
- TITLE_ID, FK

READERS table

- READER_ID, PK
- First_Name
- Last_Name
- ADDRESS
- CITY_ID
- CITY_NAME
- PHONE

BORROWING table

- BORROWING_ID,PK
- READER_ID, FK
- TITLE_ID, FK
- DATE

HISTORY table

- READER_ID,FK,PK
- TITLE_ID,FK,PK
- DATE_OF_BORROWING
- DATE_OF_RETURNING

1. Is there any inconsistencies between this schema and its description? If yes, fix the schema accordingly.

2. Are these tables in 3rd Normal Form (3NF)? Why or why not?
3. If it was not in 3NF, then put it in 3NF.
4. If two authors work together on writing the same title, can we store that information in the above database? If not, then modify the schema so that it's possible to store more than 1 author per book.
5. I want to track the people who borrowed a book, so that, in case one book was damaged (pages torn for instance), I can contact the latest person who borrowed the book to check with them what happened. So, when a reader borrows a book, I make an entry in BORROWING table. After he returns the book, I created a trigger that automatically deletes that entry and I make another entry in the HISTORY table. Is this a good idea? Should I have instead one single BORROWING table with a DATE_OF_RETURNING column? Why or why not?
6. Draw the final ER diagram of the database.

Exercise 3: motorbike repair shop

You are designing a database for a motorbike repair shop.

When a customer brings in a vehicle, a service advisor writes up a repair order. This repair order will identify the customer and the vehicle, along with the date of service and the name of the service advisor in charge.

A vehicle might need several different types of services in a single visit. These could include oil change, lubrication, rotate tires, and so on. We don't consider here the case where spare parts of the motorbike need to be replaced.

Each type of service is billed at a pre-determined number of hours work, regardless of the actual time spent by the technician (example: oil change 0.75h).

The shop wants to keep track of which mechanics made which repair. Multiple mechanics can work on the same motorbike during a single visit, but each service is made by only one mechanic.

Each type of service also has a flat book rate of price-per-hour (example: oil change 200.000VND/h; rotate tires: 100.000VND/h).

1. Give the ER diagram corresponding to this database, using Chen's notations
2. Map it into a relational schema
3. Verify that the relational schema you obtain is in 3NF

Exercise 4: Lists for Santa Claus

You are designing a database to help Santa Claus to keep track of the toys he gives to children because, of course, Santa doesn't want to give a child any of the same toys this year as he gave them last year.

He obviously needs to know the name and address of each child on his list, and when they were born.

Every year, each child will give Santa a list of the toys that he/she wants (even though, sadly, some children don't, because they don't believe in Santa Claus).

Santa will record whether that child has been naughty or nice that year, then pick which toys to actually deliver.

A child won't get more than one of each toy model, and probably won't get everything that he/she asked for in the list. If the child has been naughty, he/she might get some disappointing toy model that wasn't asked for in the list (like a plastic lump of coal).

Hint: there is (at least one) simple solution to this exercise.

1. Give the ER diagram corresponding to this database, using UML notations
2. Map it into a relational schema
3. Verify that the relational schema you obtain is in 3NF

Exercise 5: list of degrees

A small faculty in a small university needs to keep track of their faculty members. For that purpose, they currently are using an Excel sheet containing the below information.

Some members (like secretaries) might have only one degree, whereas some other members (like full professors) have multiple degrees. They counted that, with the BSc degree, the MSc degree and the PhD, every faculty member should not have more than 3 degrees.

So in the Excel file, some members have only 1 field 'degree1' filled in (and 'degree2' and 'degree3' set to NULL), whereas other members might have all three degrees filled in. Of course, each degree of each staff member is described by its level (BSc, MSc…), year, major and university (not necessarily in this order).

1. What are the main drawbacks of the above-mentioned database design?

2. Give two ER diagrams which solve these drawbacks, using Chen's notations.
3. Now, let's say that we want to store as well the grade (GPA) of the faculty members, for each degree. Which of the two above designs is the best for that purpose?
4. Map it into a relational schema (with the grade).
5. Verify that the relational schema you obtain is in 3NF.