

CS2102 Database Systems

Semester 1 2019/2020

Assignment 04 (Attachment)

Problem Description

Question 9 and 12 of Assignment 4 are based on the following application about Stuff Sharing. This was part of project description from last semester. The topic is included to give you some idea about the problem.

Stuff Sharing. *This is a stuff sharing application that allows people to borrow or loan stuff that they own (e.g., tools, appliances, furniture or books) either for free or for a fee (e.g., <https://www.peerby.com>). Users can advertise their available stuff for loan (what stuff, where to pick up and return, when it is available, etc.) or can browse the available stuff and bid to borrow some stuff. The successful bidder for an advertised item could either be chosen by the item owner or automatically selected by the system based on some criteria. Each user has an account.*

We are interested in another approach to designing schema and to connect the concept on ER diagram with functional dependencies. As such, we will ignore key constraints and total participation constraints. The approach of schema design for the last two questions is based on normalization. We start with a unified table/relation that contains all the possible attributes that is required for a normal core operation of the system we are developing. We will be using the following acronym to simplify the presentation:

L_ID : Lender's ID

L_ADDR : Lender's address

B_ID : Borrower's ID

B_ADDR : Borrower's address

I_CAT : Item categories

I_NAME : Item names

S_DATE : Date of borrow

E_DATE : Date of return

AVAIL : Item availability flag

PRICE : Price of rent (can be 0 if free)

We can capture all the data we required for the normal core operation through the use of the following table/relation:

ALL_STUFF(L_ID, L_ADDR, B_ID, B_ADDR, I_CAT, I_NAME, S_DATE, E_DATE, AVAIL, PRICE).

The decomposition method of database design is to start from this unified relation, list all the constraints, and obtain the decomposed schema at the end. You are advised to revise the lecture on ER diagram as well as the lecture on functional dependencies for these questions.

Relational Algebra

We can further require the following constraints related to functional dependencies to be satisfied on the table/relation *ALL_STUFF*. The necessary functional dependencies are written below for clarity.

1. Every lender is uniquely identified by their ID
2. Lender's address must be recorded
 - $L_ID \rightarrow L_ADDR$
3. Every borrower is uniquely identified by their ID
4. Borrower's address must be recorded
 - $B_ID \rightarrow B_ADDR$
5. We allow multiple items to have the same name, but each lender can always identify their own item
6. Given a uniquely identified item, we can know its category
 - $\{I_NAME, L_ID\} \rightarrow I_CAT$
7. Given a uniquely identified item, we can know whether it is available
 - $\{I_NAME, L_ID\} \rightarrow AVAIL$
8. We record the ID of both the lender and the borrower in every borrow event, as well as the uniquely identified item being borrowed
9. Every borrow event must have a start date, an end date, and a price
 - $\{L_ID, B_ID, I_NAME\} \rightarrow \{S_DATE, E_DATE, PRICE\}$