Assignment: Decomposition and Normal Forms

COMPSCI 2DB3: Databases–Winter 2023

Deadline: April 2, 2023

Department of Computing and Software
McMaster University

Please read the *Course Outline* for the general policies related to assignments.

Plagiarism is a *serious academic offense* and will be handled accordingly.
All suspicions will be reported to the *Office of Academic Integrity*
(in accordance with the Academic Integrity Policy).

This assignment is an *individual* assignment: do not submit work of others. All parts of your submission *must* be your own work and be based on your own ideas and conclusions. Only *discuss or share* any parts of your submissions with your TA or instructor. You are *responsible for protecting* your work: you are strongly advised to password-protect and lock your electronic devices (e.g., laptop) and to not share your logins with partners or friends! If you *submit* work, then you are certifying that you have completed the work for that assignment by yourself. By submitting work, you agree to automated and manual plagiarism checking of all submitted work.

Late submission policy. Late submissions will receive a late penalty of 20% on the score per day late (with a five hour grace period on the first day, e.g., to deal with technical issues) and submissions five days (or more) past the due date are not accepted. In case of technical issues while submitting, contact the instructor *before* the deadline.

Description

Part 1: The analysis of a table

Consider the following table that stores data for pet-owners on an online pet-fan website:

Pet(pet_id, name, pet_age, owner_id, since, owner_age, pet_friend_id).

In this relational schema, each *pet* has a unique identifier (pet_id), a name (name), and an age (pet_age). Each pet can have multiple *owners*, e.g., a family that has a single dog. We identify each owner by its identifier (owner_id). Furthermore, each owner has an age (owner_age). Furthermore, we keep track of the date at which the owner started *ownership* of the pet (since). Finally, we keep track of all the other pets that are *friends* of a pet. Each such pet is identified by its identifier (pet_friend_id). Next, an example of an instance of this relational schema:

pet_id	name	pet_age	owner_id	since	owner_age	pet_friend_id
1	Alicia	12	3	Jan. 12	24	2
1	Alicia	12	3	Jan. 12	24	3
1	Alicia	12	5	Feb. 17	29	2
1	Alicia	12	5	Feb. 17	29	3
2	Bo	9	7	Mar. 5	21	1
2	Bo	9	7	Mar. 5	21	3
3	Celeste	5	1	Dec. 22	99	1

Question

1. Provide a minimal cover of *all realistic* non-trivial functional dependencies that hold on the above relational schema. Argue, for each functional dependency, why this functional dependency hold.

   Hint: We only require a *minimal cover*. Hence, there is no need for trivial functional dependencies and functional dependencies that can be derived from other functional dependencies.

2. Are there any other *non-trivial* dependencies that hold on this table? If so, provide an example of such a dependency and argue why this dependency holds.

   Hint: E.g., multi-valued dependencies, inclusion dependencies, or join dependencies.

Part 2: Refinement of a table

A book-ordering website wants to keep track of all their sales. To do so, they developed the following relational schema to hold all relevant information:

Order(id, user_id, user_location, date,
book_id, format, amount, isbn, title
author, publisher, age_rating, child_friendly, category.

In this relational schema, the following order information will be stored:

i. The identifier *id* of the order, the identifier *user_id* of the user that placed the order, the location *user_location* of that user, and the date *date* at which the order was placed.

ii. The information on the books ordered. Each order can order several books and several different formats (e.g., hardcover, paperback, e-book) of the same book. Hence, we store the identifier *book_id* of the ordered book (as used internally by the website), the format *format* of the ordered book, and the *amount* of copies ordered.

iii. Publishing information of the book, including the *ISBN*, the *title*, one-or-more *authors*, the *publisher*, the *age-rating* of the book and whether the book is suitable for children, and, finally, one-or-more *categories* of the book.

Next, an example of an instance of this relational schema (we use shorthand notations for each attribute: *Id* is *id*, *Ui* is *user_id*, *Ul* is *user_location*, *D* is *date*, *B* is *book_id*, *F* is *format*, *Am* is *amount*, *Is* is *isbn*, *T* is *title*, *Au* is *author*, *P* is *publisher*, *Ar* is *age-rating*, *Cf* is *child_friendly*, and *Ca* is *category*):

Id	Ui	Ul	D	B	F	Am	Is	T	Au	P	Ar	Cf	Ca
1	1	Hamilton	24 Dec.	2	paperback	3	1234	Book!	Alicia	ThePrinter	18+	no	CS
1	1	Hamilton	24 Dec.	2	paperback	3	1234	Book!	Alicia	ThePrinter	18+	no	Theory
1	1	Hamilton	24 Dec.	2	paperback	3	1234	Book!	Dafni	ThePrinter	18+	no	CS
1	1	Hamilton	24 Dec.	2	paperback	3	1234	Book!	Dafni	ThePrinter	18+	no	Theory
1	1	Hamilton	24 Dec.	2	hardcover	6	1237	Book!	Alicia	ThePrinter	18+	no	CS
1	1	Hamilton	24 Dec.	2	hardcover	6	1237	Book!	Alicia	ThePrinter	18+	no	Theory
1	1	Hamilton	24 Dec.	2	hardcover	6	1237	Book!	Dafni	ThePrinter	18+	no	CS
1	1	Hamilton	24 Dec.	2	hardcover	6	1237	Book!	Dafni	ThePrinter	18+	no	Theory
2	2	Toronto	25 Dec.	2	e-book	1	1241	Book!	Alicia	ThePrinter	18+	no	CS
2	2	Toronto	25 Dec.	2	e-book	1	1241	Book!	Alicia	ThePrinter	18+	no	Theory
2	2	Toronto	25 Dec.	2	e-book	1	1241	Book!	Dafni	ThePrinter	18+	no	CS
2	2	Toronto	25 Dec.	2	e-book	1	1241	Book!	Dafni	ThePrinter	18+	no	Theory
3	1	Hamilton	26 Dec.	5	paperback	1	1298	Comic!	Bo	TheCopier	5+	yes	Comedy

The primary key of this table is "*id, book_id, format, author, category*". Furthermore, the following functional dependencies hold on this table:

$$\begin{aligned}
\text{id} &\longrightarrow \text{user_id, user_location, date;}\\
\text{user_id} &\longrightarrow \text{user_location;}\\
\text{book_id} &\longrightarrow \text{title, publisher, age_rating, child_friendly;}\\
\text{id, book_id, format} &\longrightarrow \text{amount;}\\
\text{book_id, format} &\longrightarrow \text{isbn;}\\
\text{isbn} &\longrightarrow \text{book_id, format;}\\
\text{age_rating} &\longrightarrow \text{child_friendly.}
\end{aligned}$$

Question

3. Is the relational schema **Order** in 3NF?

   If so, then explain why **Order** is in 3NF.

   Otherwise, decompose the schema using the 3NF Synthesis algorithm (Decompose-3NF) and document each step you make while applying the algorithm. Provide the functional dependencies that hold in each relational schema in your resulting decomposition (a minimal cover suffices). Explain whether this decomposition is lossless-join and whether it is dependency-preserving (with respect to the original functional dependencies). Finally, decompose the example dataset according to the relational schema obtained from the decomposition algorithm.

4. Is the relational schema **Order** in BCNF?

   If so, then explain why **Order** is in BCNF.

   Otherwise, decompose the schema using the BCNF Decomposition algorithm (Decompose-BCNF) and document each step you make while applying the algorithm. Provide the functional dependencies that hold in each relational schema in your resulting decomposition (a minimal cover suffices). Explain whether this decomposition is lossless-join and whether it is dependency-preserving (with respect to the original functional dependencies). Finally, decompose the example dataset according to the relational schema obtained from the decomposition algorithm.

5. According to the consultant, the following multi-valued dependencies also hold:

   $$\begin{aligned}
   \mathfrak{I} &\twoheadrightarrow \text{author; and}\\
   \mathfrak{I} &\twoheadrightarrow \text{category,}
   \end{aligned}$$

   in which $\mathfrak{I}$ are all attributes except the two attributes *author* and *category*. Is the relational schema **Order** in 4NF?

   If so, then explain why **Order** is in 4NF.

   Otherwise, decompose the schema using the 4NF Decomposition algorithm (Decompose-4NF) and document each step you make while applying the algorithm. Provide the functional dependencies that hold in each relational schema in your resulting decomposition (a minimal cover suffices). Explain whether this decomposition is lossless-join and whether it is dependency-preserving (with respect to the original functional dependencies). Finally, decompose the example dataset according to the relational schema obtained from the decomposition algorithm.

6. Does any of the above three decompositions of **Order** resolve all design issues of **Order**? If so, explain which decomposition(s) resolve all design issues. Else, provide an example of a design issue that was not resolved by decomposition.

Assignment

The goal of the assignment is to help out the local community leader and the local cinema chain owner. To do so, you will write a report in which you answer Questions 1–6. Your submission:

1. must be a PDF file;
2. must have clearly labeled solutions to each of the stated questions;
3. must include explanation of the steps taken when applying a decomposition algorithm in Questions 3–5;
4. must be clearly presented;
5. must *not* be hand-written: prepare your document in Microsoft Word or another word processor (printed or exported to PDF) or in LaTeX.

Submissions that do not follow the above requirements will get a grade of zero.

Grading

The presented solution for Question 1 will account for 10% of the maximum grade; the presented solution for Question 2 will account for 5% of the maximum grade; the presented solutions for Questions 3, 4, and 5 will account for 25% of the maximum grade each; and the presented solution for Question 6 will account for 10% of the maximum grade.