Assignment: Decomposition and Normal Forms

COMPSCI 2DB3: Databases–Winter 2024

Deadline: March 30

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 are aware of the *Plagiarism and Academic Dishonesty* policy of this course outlined in this section, that you are aware of the Academic Integrity Policy, and that you have completed the submitted work entirely yourself. Furthermore, 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 cinema event data:

Event(id, date, movie_id, genre, actor, snack, visitor, weekend).

This relational schema will be used in a database designed to maintain data on *movie-events*. A movie-event is a point in time during which the cinema promotes a certain film by placing it in the spotlight. During such an event, some important actors of the film will visit the cinema, provide autographs, and talk about their role. These events are typically intended for marketing and come with free snacks provided by the cinema.

Each event has a unique identifier (*id*), a date of organization, a movie identifier identifying the shown movie, the *main* genre of the movie, one-or-more actors that will show up, one-or-more snacks that will be provided, the estimated number of visitors, and a Boolean flag indicating whether the event is in the weekend. There can be multiple movie-events per day.

See below for an example of this event data.

id	date	movie_id	genre	actor	snack	visitor	weekend
1	17/08/2013	7	scifi	Joe	chips	800	yes
1	17/08/2013	7	scifi	Joe	hotdog	800	yes
1	17/08/2013	7	scifi	Jane	chips	800	yes
1	17/08/2013	7	scifi	Jane	hotdog	800	yes
2	17/08/2013	8	comedy	Jane	icecream	200	yes
3	20/08/2013	9	scifi	Bart	chips	400	no
3	20/08/2013	9	scifi	Jane	chips	400	no

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 web shop wants to keep track of all their sales. To do so, they developed the following relational schema to hold all relevant information:

Order(id, employer_id, employer_rank, user_id, date, product_id
amount, price, description, category, category_id,
option, option_price).

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

i. The identifier *id* of the order, the identifier *employer_id* of the employer that placed the order, the rank of the employer, the identifier *user_id* of the user that will receive the order, and the date *date* at which the order was placed.

ii. The information on the products ordered. Each order can contain several products and for each product the ordered amount is stored, the price (per unit), and the description. Linked to each product are all options bought with the product and the price of adding these options.

iii. Each product belongs to one or more categories.

Next, an example of an instance of this relational schema (we use shorthand notations for each attribute: *Id* is *id*, *E* is *employer_id*, *Er* is *employer_rank*, *U* is *user_id*, *D* is *date*, *P* is *product_id*, *A* is *amount*, *Pr* is *price*, *Ds* is *description*, *C* is *category*, *Ci* is *category_id*, *O* is *option*, and *Op* is *option_price*:

Id	E	Er	U	D	P	A	Pr	Ds	C	Ci	O	Op
1	3	Junior	21	18 Apr.	4	2	$5.00	Ink	Printer	43	Warranty	$12.00
1	3	Junior	21	18 Apr.	4	2	$5.00	Ink	Office	12	Warranty	$12.00
1	3	Junior	21	18 Apr.	7	1	$55.00	Scanner	Printer	43	Installation	$76.00
1	3	Junior	21	18 Apr.	7	1	$55.00	Scanner	Office	12	Installation	$76.00
1	3	Junior	21	18 Apr.	7	1	$55.00	Scanner	Printer	43	Warranty	$14.00
1	3	Junior	21	18 Apr.	7	1	$55.00	Scanner	Office	12	Warranty	$14.00
2	8	Senior	17	19 Apr.	12	1	$1200.00	Laptop	PCs	23	Mouse	$15.00
2	8	Senior	17	19 Apr.	12	1	$1200.00	Laptop	Office	12	Mouse	$15.00
2	8	Senior	17	19 Apr.	12	1	$1200.00	Laptop	PCs	23	Warranty	$140.00
2	8	Senior	17	19 Apr.	12	1	$1200.00	Laptop	Office	12	Warranty	$140.00

The primary key of this table is "*id, product_id, category_id, option*". Furthermore, the following functional dependencies hold on this table:

$$\begin{aligned}
\text{id} &\longrightarrow \text{employer_id, employer_rank, user_id, date;}\\
\text{employer_id} &\longrightarrow \text{employer_rank;}\\
\text{id, product_id} &\longrightarrow \text{amount, price;}\\
\text{product_id} &\longrightarrow \text{description;}\\
\text{category_id} &\longrightarrow \text{category;}\\
\text{product_id, option} &\longrightarrow \text{option_price.}
\end{aligned}$$

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:

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

   in which $\mathfrak{I}$ are all attributes except the four attributes *option*, *option_price*, *category*, and *category_id*. 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 include your student number and MacID;
2. must be a PDF file;
3. must have clearly labeled solutions to each of the stated questions;
4. must include explanation of the steps taken when applying a decomposition algorithm in Questions 3–5;
5. must be clearly presented;
6. 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.