Project 4: Database Normalization

Samuel B. Scalf

School of Cybersecurity and Information Technology, University of Maryland Global Campus

CMIS 320: Relational Database Concepts and Applications

Dr. Karl B. Lloyd

May 11, 2021

1. Is this relation in at least 1NF? Why or why not?

This relation is in at least 1NF, because it satisfies the requirements for 1NF: each field of the table contains exactly one item, even if it is multiple words; each field in a column means the same thing; there are no identical tuples, as they all differ by at least one attribute; and there are no repeating columns, such as GAME1, GAME2, etc. Furthermore, we were instructed to assume tuples with the same name in the GIRL attribute are the same person. This, along with another attribute, allow us to uniquely identify each tuple.

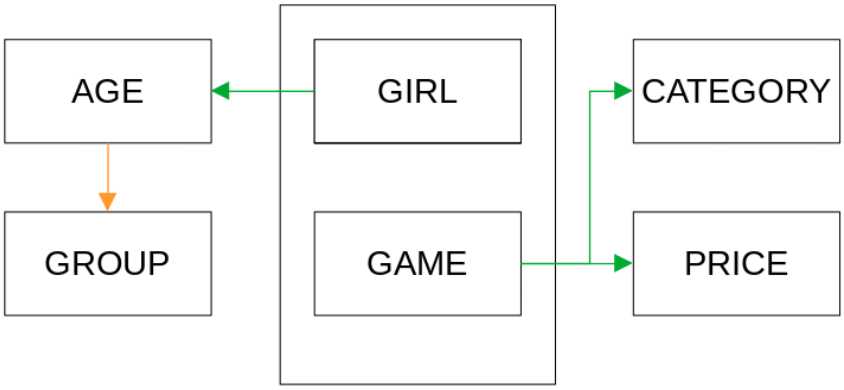
2. What is the primary key of the initial relation (assume the values shown are the only possible tuples for all time)? Remember that a primary key must be unique and not null.

Since we can assume "girls with the same name are the same person," we can use the GIRL attribute as a primary key. This alone, however, is not enough, because there are duplicate entries, or more than one tuple with the same GIRL value. We see that the GAME attribute does not repeat for tuples with the same GIRL value. In other words, there are no tuples with the same GIRL-GAME pair, making the pair an ideal composite primary key. All said, the primary key of the initial relation is the composite primary key, GIRL-GAME.

3. Describe the specific data anomalies that exist if we DELETE the tuple containing Jacqueline.

If we DELETE the tuple containing Jacqueline, we lose all information about a GAME, CATEGORY, and PRICE. More specifically, we lose Visual Basic, Prog. Languages, and 199.99, respectively, from our database, despite the intent to only remove data about Jacqueline.

4. Draw a functional dependency diagram for the inital relation. This diagram should agree with the primary key you selected in above.



5. Based on your diagram, what normal form is the initial relation in ? Why?

Based on my diagram, the initial relation is 1NF. Although no duplicate, or repeating, attributes exist, partial key dependencies do exist: at least two distinct groups of dependencies are present.

6. If necessary, decompose the initial relation into a set of non-loss 3NF relations by showing the relations, attributes, and tuples. Show complete relations with attribute headings and all data values in the tuples of your relations. Determine the number of 3NF relations you end up with after normalization, write this number, and the circle the number.

Normalization to 2NF

The 1NF relation, PURCHASES1 (GIRL, GROUP, AGE, GAME, CATEGORY, PRICE) contains the following functional dependencies, which includes partial key dependencies.
GIRL, GAME → GROUP,
GIRL → GROUP,
GIRL, GAME → AGE,
GIRL → AGE,
GIRL, GAME → CATEGORY,
GAME → CATEGORY,
GIRL, GAME → PRICE, and
GAME → PRICE

In order for the relation to be in 2NF, all partial key dependencies must be “broken.” To accomplish this, we can decompose the 1NF relation
PURCHASES1 (GIRL, GROUP, AGE, GAME, CATEGORY, PRICE)
in to the PURCHASES1B (GIRL, GROUP, AGE, GAME) and GAMES2 (GAME, CATEGORY, PRICE).

GAMES2 (GAME, CATEGORY, PRICE) contains the following functional dependencies, which do not include any partial key dependencies.
GAME → CATEGORY, and
GAME → PRICE

PURCHASES1B (GIRL, GROUP, AGE, GAME), however, is still not in 2NF, as it contains the following functional dependencies, including some partial key dependencies.
GIRL, GAME → GROUP,
GIRL → GROUP,
GIRL, GAME → AGE, and
GIRL → AGE

The 1NF relation PURCHASES1B (GIRL, GROUP, AGE, GAME) can be decomposed in to the 2NF relations, GIRLS2 (GIRL, GROUP, AGE) and PURCHASES2 (GIRL, GAME).

PURCHASES2 (GIRL, GAME) does not contain any non-key attributes, which means there are, of course, no partial key dependencies.

GIRLS2 (GIRL, GROUP, AGE) contains the following functional dependencies, which do not include any partial key dependencies.
GIRL → GROUP, and
GIRL → AGE

Normalization to 3NF

The 2NF relation GAMES2 (GAME, CATEGORY, PRICE) contains the functional dependencies,
GAME → CATEGORY and
GAME → PRICE, but not
CATEGORY → PRICE nor
PRICE → CATEGORY,
so it is already in 3NF.

The 2NF relation PURCHASES2 (GIRL, GAME) does not contain any non-key attributes, so it is automatically in 3NF.

The 2NF relation GIRLS2 (GIRL, GROUP, AGE) contains the functional dependencies,
GIRL → GROUP,
GIRL → AGE,and
AGE → GROUP,
which includes the transitive dependency
GIRL → AGE → GROUP.

The transitive dependeny can be removed from GIRLS2 (GIRL, GROUP, AGE) by decomposing it to the 3NF relations GIRLS3 (GIRL, AGE) and AGEGROUPS3 (AGE, GROUP).

The complete set of 3NF relations for the data model is:
GIRLS3 (GIRL, AGE),
AGEGROUPS3 (AGE, GROUP),
PURCHASES3 (GIRL, GAME), and
GAMES3 (GAME, CATEGORY, PRICE).

The foreign key GIRLS3.AGE relates to the primary key AGEGROUPS3.AGE.

The foreign key PURCHASES3.GIRL relates to the primary key GIRLS3.GIRL.

The foreign key PURCHASES3.GAME relates to the primary key GAMES3.GAME.

The complete set of 3NF relations for the data model is illustrated in the tables below:

GIRLS3(GIRL, **AGE**)

GIRL	**AGE**
Charlotte	5
Susan	6
Jane	5
Carrie	6
Jacqueline	5

AGEGROUPS3(AGE, GROUP)

AGE	GROUP
5	5 year olds
6	6 year olds

PURCHASES3(GIRL, **GAME**)

GIRL	**GAME**
Charlotte	Mirror
Susan	Lipstick
Jane	Chess
Susan	Checkers
Susan	Mirror
Carrie	Lipstick
Jacqueline	Visual Basic

GAMES3(GAME, CATEGORY, PRICE)

GAME	CATEGORY	PRICE
Mirror	Makeup	4.88
Lipstick	Makeup	5.95
Chess	Games	7.55
Checkers	Games	5.95
Visual Basic	Prog. Languages	199.99

Converting the 1NF relations to 3NF relations has resolved the following problems:

- no duplication of data remains in PURCHASES3
- GAMES and GIRLS can be inserted without the other existing
- deleting GIRL, Jacqueline, does not delete all references to Visual Basic | Prog. Languages | 199.99
- updating the price of Lipstick | Makeup | 5.95 requires changing only one record.

After normalization, I ended up with four (4) 3NF relations.