Project 4

Vadym Kharchenko

University of Maryland Global Campus

CMSC 320: Relational Database Concepts and Applications

Prof. Candice Adams

October 2, 2024

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

The relation is in 1NF form. There are two cretarias that makes it 1NF, the first is that every column value is an “atomic” entity, and the second is that every entry must have a single value. This relation satisfies both criteria.

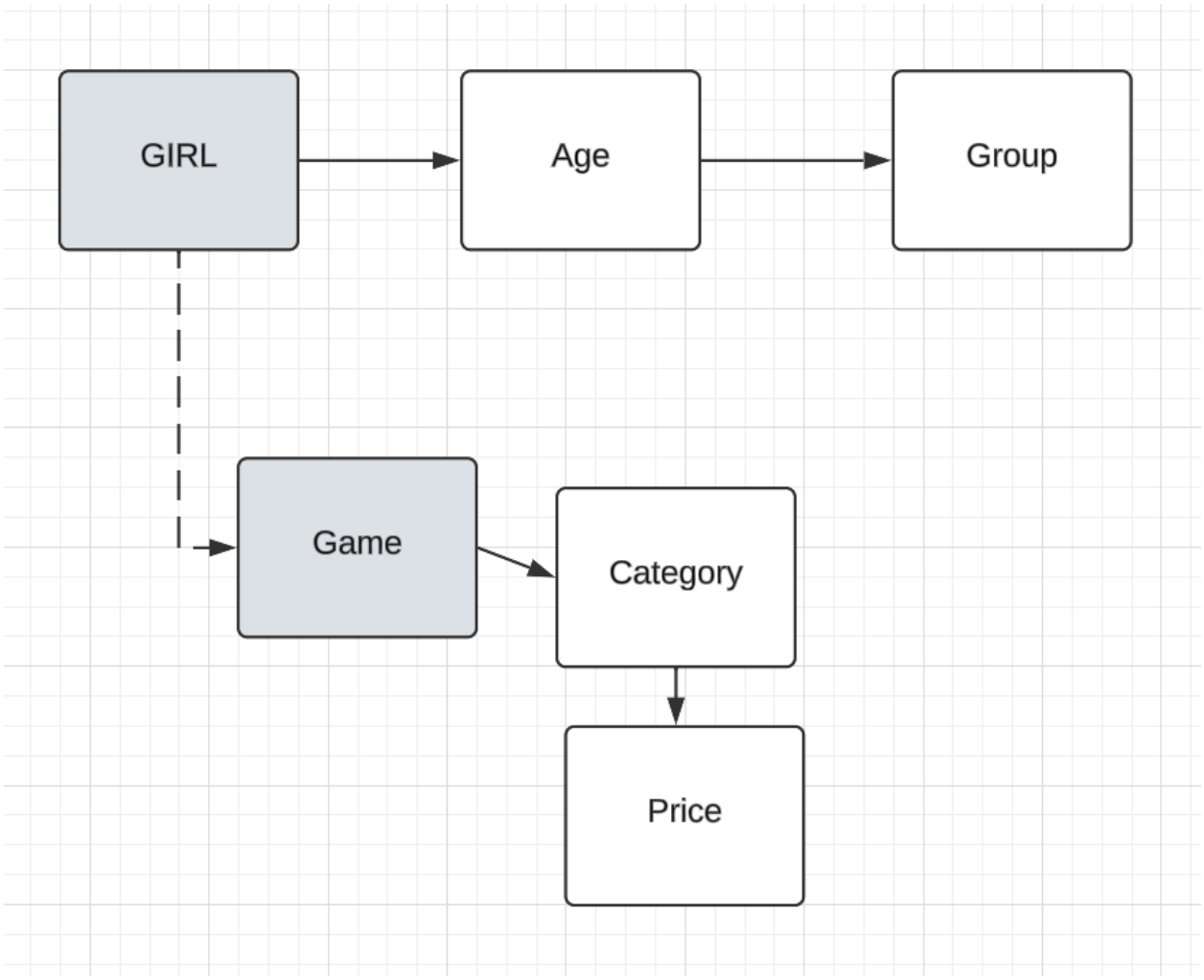
2. What is the primary key of the initial relation (assume the values shown are the only possible tuples for all time)? Remember that a primary key must be unique and not null.

Since the same person can have a different game and the same game could have different people. Thus, the primary key is a combination and it is (Name, Game, Age), it identifies a unique tuple value.

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

After the deletion of ‘Jacqueline’, we lose all the references to ‘Visual Basic’ and it’s price. This is because it’s the only entry that contains this entry. Thus, we lose the data. Furthermore, to update anomaly if we need to update the price of "Visual Basic," we can't do so after the deletion because all information about the game is gone.

4. Draw a functional dependency diagram for the initial relation. This diagram should agree with the primary key you selected in above. This can be drawn in any drawing tool. Be sure to identify dependencies like full, partial, transient, etc. between the attributes.



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

As I already mentioned in question #1, the relation is in 1NF form. It is not in 2NF form because the are not all partial dependencies require both primary key composites. Lastly, it is not 3NF because it is not 2NF, and there are no transitive dependencies.

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 then circle the number.

Step 1: lets make a relation 2NF.

We need to remove partial dependencies and split the relation.

{GIRL, AGE, GROUP} //What girl, what's her age, and what age group she belongs to - relation.

{GAME, CATEGORY, PRICE} //What game belongs to what category and at what price - relation.

{GIRL, GAME} //What girl play what game - relation.

Step 2. Now when we got rid of all the partial dependencies, we need to work on the transitive dependencies. Since, as we discussed previously in question #5, there are no transitive dependencies. Thus, after normalization the form that we have qualifies to be 3NF.

Optional: We can also merge AGE and GROUP.

{GIRL, AGE}

{GAME, CATEGORY, PRICE}

{GIRL, GAME}