NYU Tandon School of Engineering
CS-GY 6083, Principles of Database Systems, Fall 2024
Prof Phyllis Frankl

HOMEWORK #4

Suppose someone designed a database with **only one relation** schema as shown below to model current orders for a shoe store, similar to the scenario in HW 1 problem 2c.

The schema is:
ShoeOrder(brand, styleID, size, color, email, orderDate, basePrice, pricePaid, status, qInStock, qOrdered, bonusPts, phone)

We have the following detailed specifications for the database system:
Each type of shoe the store carries is identified by its *brand, styleID, size*, and *color*; in addition, the inventory keeps track of the quantity (*qInStock)* of each type of shoe that is currently in stock. Each customer has a unique *email,* one *phone number*, and a number of *bonus points*. In addition to storing the prices customers paid, they would like to store the *base price* of each type of shoe. Furthermore, the base price is determined by the brand and styleID. In other words, shoes that have the same brand and styleID always have the same base price, even though they may have different size and/or color.

1. Using your group members as customers (and additional fictional data as needed) show an example of a relation on this schema where
    a. Two customers have ordered the same brand/style/color/size of shoes, and
    b. There are multiple colors and/or sizes of at least one brand/style of shoes, and
    c. There is a customer who has no orders, and
    d. There are some black size 9 Adidas Sambas in stock, and no one has ordered any of that shoe type.

   Comment briefly on problems with this schema that these data illustrate.

2. Give an example of a trivial functional dependency in this schema (Recall that a functional dependency alpha $\alpha \rightarrow \beta$ is trivial if $\beta \subseteq \alpha$) and an example of a functional dependency in which the left-hand side is a superkey. (Such functional dependencies do not violate BCNF)

3. Write functional dependencies corresponding to each of the following parts of the data description:
    a. Each type of shoe the store carries is identified by its *brand, styleID, size*, and *color*; in addition, the inventory keeps track of the quantity (*qInStock)* of each type of shoe that is currently in stock.

   b. The base price is determined by the brand and styleID. In other words, shoes that have the same brand and styleID always have the same base price, even though they may have different size and/or color.
   c. Each customer has a unique *email,* one *phone number*, and a number of *bonus points*.

4. What is the canonical cover of the set of dependencies implied by the description.

5. Derive a candidate key for the ShoeOrder schema.

6. Show that the ShoeOrder schema is not in BCNF

7. Decompose ShoeOrder into a collection of schemas each of which is in BCNF. **Show your work: at each stage show which schema you are decomposing, which functional dependency violating BCNF you are using for the decomposition, and the resulting two schemas.** At the end, show the entire decomposed database schema and give meaningful names to each relation schema.

8. Show how the data in (1) is stored using the decomposed database schema from (7). Comment on how the decomposition addresses the anomalies you noted in part (1).

9. Suppose the Shoe store changed its price structure to guarantee that all purchases of shoes with the same *brand* and *styleID* on the same *orderDate* have the same *PricePaid*.
   a. Write a functional dependency to characterize this
   b. Decompose your answer from (7) further (if necessary) so it will be in BCNF with respect to the expanded set of functional dependencies
   c. Is your result still dependency-preserving? Justify your answer.
   d. **OPTIONAL**: If the BCNF decomposition is not dependency-preserving, decompose the original schema using the algorithm for decomposition into 3NF. (We did not go over this in the lecture and you are not responsible for it, but you can find the algorithm in the notes and in the textbook and you might find this interesting or useful in the real world.)

10. Repeat part 9 with the following supposition (starting with your decomposed database schema from part (7)): Purchases made on the same *orderDate* of shoes with the same *basePrice* have the same *pricePaid*