

Team 100 - Group sudo_rm_rf

PT1 - Stage 2

Relational schema:

User(UserID:INT [PK],
Email:VARCHAR(50),
Password: SOME KIND OF HASH IDK)

Result(ResultID: INT [PK],
UserID:INT [FK to User.UserID],
CreationDate:DATETIME,
ResultName:VARCHAR(50))

InteractionPair(dfID1: VARCHAR(6) [FK to DrugFoodIDs.FoodDrugID],
dfID2: VARCHAR(6) [FK to DrugFoodIDs.FoodDrugID],
ResultID:INT [FK to Result.ResultID],
(dfID1, dfID2, ResultID) [PK],
InteractionResult: INT)

DrugFoodIDs(FoodDrugID: VARCHAR(6) [PK],
FoodDrugName: VARCHAR(50))

DrugBrand(ReferenceDrug: VARCHAR(50) [FK to DrugFoodIDs.FoodDrugName],
BrandedName: VARCHAR(50))

Relationships:

- User-Result
 - Description: One user can create many sets of results, each with their own unique set of medications/diets which contain results from different drug food/drug drug interactions.
 - Cardinality: One-to-Many
- DDI/DFI pair - Result
 - Description:
 - One drug-drug/drug-food/drug-herb interaction pair should correspond to exactly one result since this represents interactions with medications which users input.
 - Cardinality: One-to-One
- Food/Drug-BrandName
 - Description:

- We can use the Brand Name of drugs to figure out its active ingredient, for instance Tylenol is actually the drug acetaminophen.
 - Cardinality:
 - Many to one, as there could be many branded names for a given drug
 - Food/Drug-DDI/DFI pair
 - Description:
 - The drug/food IDs listed in the DDI/DFI table correspond to drug/food names and, in the case of drugs, possible branded names. There is also the possibility of linking additional information to the drug/food IDs in DDI/DFI such as caloric content or chemical formulas
 - Cardinality: Many-to-Many
 - Each drug/food can be used in multiple DDI/DFI interaction pairs and each DDI/DFI interaction pair can have multiple drug/foods linked.
-

BCNF Normalization

To ensure that this schema adheres to BCNF we must make sure that for every functional dependency $X \rightarrow Y$, X is a superkey.

DrugBrand

- Reference Drug
- BrandName

$(\text{ReferenceDrug}, \text{BrandName}) \rightarrow \text{DrugName}$

In this situation DrugID is the superkey in this table and associates BrandedName. There are no other dependencies.

DrugFoodIDs

- FoodDrugID
- FoodDrugName

FoodDrugID is a primary key that uniquely identifies each FoodDrugName. Given that you can determine FoodDrugName through the FoodDrugID, FoodDrugID is a superkey.

User

- UserID
- Email
- Password

UserId -> Email, Password

UserID is a superkey that identifies Email and password. There are no other functional dependencies.

InteractionPair

- dfID1
- dfID2
- ResultID
- InteractionResult

The primary key is (dfID1,dfID2) which identifies each interaction and the result. The rest of the result is associated with a resultID which handles the other dependencies.

Result

- ResultID
- UserId
- CreationDate
- ResultName

The primary key is ResultID which is a primary key that identifies all of the other values. ResultID is a superkey and can identify every other attribute.

Since all of these different tables contain a superkey and all other values are dependent on these superkeys they comply with BCNF.