Normalization Practice #2

Posted 2/24/2012

Inventory Relation

manufacturer	brandName	type	weightInOz	store
Kellogs	Frosted Flakes	Cereal	16	Benson Sundry
Kraft	EZ Cheese	Dairy	6	Benson Sundry
Paul Newman's Own	Sockarooni Sauce	Sauce	12	North Campus Market

This relation stores the items that a grocery store company stocks in its varying stores. Each tuple in the Inventory relation represents the fact that a store sells items of a particular type and brand name manufactured by a particular company. Items may come in different sizes.

Functional Dependencies

- Given the following descriptions, provide the functional dependencies over Inventory attributes each description provides.
  - A manufacturer holds the trademark for a brand name of an item of a particular type, i.e., no two manufacturers can use the same brand name for items of the same type. For example, two different manufacturers cannot use the brand name Frosted Flakes for the food type Cereal

Functional Dependencies

- Given the following descriptions, provide the functional dependencies over Inventory attributes each description provides.
  - A manufacturer holds the trademark for a brand name of an item of a particular type, i.e., no two manufacturers can use the same brand name for items of the same type. For example, two different manufacturers cannot use the brand name Frosted Flakes for the food type Cereal

brandName → manufacturer

Functional Dependencies

- Given the following descriptions, provide the functional dependencies over Inventory attributes each description provides.
  - For each type, each store sells only one brand name made by each manufacturer. For example, Benson Sundry does not sell any Cereal other than Frosted Flakes that is manufactured by Kellogg's Company. It can see Fruity Pebbles, manufactured by Post, but that would be the only type of Post Cereal sold.

Functional Dependencies

- Given the following descriptions, provide the functional dependencies over Inventory attributes each description provides.
  - For each type, each store sells only one brand name made by each manufacturer. For example, Benson Sundry does not sell any Cereal other than Frosted Flakes that is manufactured by Kellogg's Company

store, type, manufacturer → brandName

Functional Dependencies

- Assume that the previous functional dependencies hold. What are the possible candidate keys for the Inventory relation, and why?

Functional Dependencies

- Assume that the previous functional dependencies hold. What are the possible candidate keys for the Inventory relation, and why?
  - Manufacturer, Store, Type, Weight

  OR
  - Brandname, Store, Type, Weight

  Store gets us to one store. Manufacturer and Type (or Brandname and Type) gets us to a specific item. Weight gets us to a specific version of the item (i.e. there is no constraint saying that a store doesn't sell two different sizes of Frosted Flakes Cereal, so we need Weight to differentiate those!)

Normalization

- What normal forms (1NF, 2NF, 3NF, BCNF) does the Inventory relation satisfy?
  - Your answer can be the highest form it satisfies, as that implies all lower forms are satisfied as well.
  - Use the generalized version of 2NF, 3NF, and make use of the functional dependencies you came up with earlier and those implied by the keys.

manufacturer	brandName	type	weightInOz	store
Kellogs	Frosted Flakes	Cereal	16	Benson Sundry
Kraft	EZ Cheese	Dairy	6	Benson Sundry
Paul Newman's Own	Sockarooni Sauce	Sauce	12	North Campus Market

Normalization

- What normal forms (1NF, 2NF, 3NF, BCNF) does the Inventory relation satisfy?
  - Your answer can be the highest form it satisfies, as that implies all lower forms are satisfied as well.
  - Use the generalized version of 2NF and 3NF, and make use of the functional dependencies you came up with earlier and those implied by the keys.

It is in 3NF, but not BCNF.

It is in 1NF – there are no multi-valued attributes.

It is in 2NF – there are no non-candidate-key-attributes that are partially dependent on some candidate key (all attributes are part of some candidate key, so actually there aren't even any non-candidate key attributes to worry about)

It is in 3NF – same reason, there are non-non-candidate-key-attributes to worry about; as well, there are no transitive dependencies

It is not in BCNF – there exist functional dependencies (FDs) where the left-hand-side is not a candidate key (in fact, both FDs we are aware of are not full candidate keys)

Make sure you understand the difference between a candidate key and a candidate-key-attribute (an attribute that composes part of a candidate key)

Decomposing Relations

- Consider the decomposition of Inventory into Inventory1(Manufacturer, Brandname, Type, Store) and Inventory2(Manufacturer, Brandname, Type, Weight).
  - Does this decomposition ensure that, if rejoined, all and only the information in the original Inventory table is re-constructed?

Decomposing Relations

- Consider the decomposition of Inventory into Inventory1(Manufacturer, Brandname, Type, Store) and Inventory2(Manufacturer, Brandname, Type, Weight).
  - Does this decomposition ensure that, if rejoined, all and only the information in the original Inventory table is re-constructed?

- No it does not – a join of these two tables on shared attributes will put all different weight versions of an item of a given type (Kellogs Frosted Flakes Cereal for example) at all stores – which may not have been true in the original table.