Bryan Rockwood

Normalization HW 1

10/28/13

CMPT 308

PART 1:

1. If Mr. Meservy asked me my opinion on the current spreadsheet, I'd politely tell him it is awful. The initial data is all crammed into one large and goofy spreadsheet that can cause numerous amounts of problems.

2.

PackageID	TagNumber	InstallDate	SoftwareCostUSD
AC01	32808	09-13-2005	754.95
DB32	32808	12-03-2005	380.00
DB32	37691	06-15-2005	380.00
DB33	57772	05-27-2005	412.77
WP08	32808	01-12-2006	185.00
WP08	37691	06-15-2005	227.50
WP08	57222	05-27-2005	170.24
WP09	59836	10-30-2005	35.00
WP09	77740	05-27-2005	35.00

3. One of the primary keys would be PackageID, as would TagNumber. These keys together create a composite key for this table. Because each Tag is unique to an users computer, it can be used as a primary key. The same idea goes with the PackageID, as software packages only contain one ID.

PART 2:

1.

PackageID	PackageName	TagNumber	CompModel	InstallDate	SoftwareCostUSD
AC01	Bastion	32808	Dell	09-13-2005	754.95
DB32	Magic: The Gathering	32808	Dell	12-03-2005	380.00
DB32	Magic: The Gathering	37691	Lenovo	06-15-2005	380.00
DB33	Adobe Illustrator	57772	Apple	05-27-2005	412.77
WP08	Pokemon	32808	Dell	01-12-2006	185.00

	Emulator				
WP08	Pokemon Emulator	37691	HP	06-15-2005	227.50
WP08	Pokemon Emulator	57222	HP	05-27-2005	170.24
WP09	Domino's Pizza Tracker	59836	Apple	10-30-2005	35.00
WP09	Domino's Pizza Tracker	77740	Apple	05-27-2005	35.00

2. Some examples of functionally dependent attributes would be PackageName and PackageID. PackageName would not exist without the PackageID, so it is dependent on the ID. We also have a functional dependency between the CompModel and TagNumber. Because there can be many different types of CompModels, the TagNumber has to identify what CompModel we are trying to find. The last dependency involves four of the attributes : PackageID, TagNumber, InstallDate, SoftwareCostUSD.

Because we need to know what computer and what software we are downloading, we need to know the PackageID and the TagNumber for the InstallDate and the SoftwareCostUSD to be related to the table.

PackageID -> PackageName

TagNumber -> CompModel

PackageID , TagNumber -> InstallDate

PackageID, TagNumber - > SoftwareCostUSD

3. This new table is not 3NF because not every attribute is directly dependent on the composite key. If attributes cannot be directly related to the composite key, then the table can still be broken down into a more focused relation.

PART 3:

1. To make our table 3NF, we need to create a couple more tables; so let's create Computer, Packages, and Installation. For our Computer table, we are going to have TagNumber as our primary key and CompModel as an attribute. In our Package table we will have PackageID as our primary key while PackageName is an attribute. And lastly in Installation we will actually have a composite key of PackageID and TagNumber for the attributes of InstallDate and SoftwareCostUSD.

2.

PackageID -> PackageName

TagNumber -> CompModel

PackageID , TagNumber -> InstallDate

PackageID, TagNumber - > SoftwareCostUSD

3. The new relation is in 3NF because of the re-arranging of tables. After we move the information to different tables, we are able to have each attribute directly depend on the primary key of that table. Thus making it 3NF

4.