Jessica Rieger

Lab 7

10/27/2016

Professor Labouseur

Part 1:

1. In response to his question, I would tell him that the data he has given me is definitely a good start. It includes many of the fields of data that we will want to record. However, I would also let him know that it is not in the ideal form for input into a database, so there are a few things that we will need to change before we can proceed.
2. The table, when put in first normal form, looks like this:

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

This is in first normal form because all of the field values are atomic. Before, it was not in first normal form because there were several tag numbers in one field.

3. The primary key in this table is (PackageID, TagNumber.)

PackageID	TagNumber	InstallDate	SoftwareCostUSD	PackageName	ComputerModel
AC01	32808	09-13-2005	754.95	Microsoft Office	Apple
DB32	32808	12-03-2005	380.00	Photo Booth	Apple
DB32	37691	06-15-2005	380.00	Photo Booth	IBM
DB33	57772	05-27-2005	412.77	McAfee	Microsoft
WP08	32808	01-12-2006	185.00	Adobe Acrobat Pro	Apple
WP08	37691	06-15-2005	227.50	Adobe Acrobat Pro	Dell

WP08	57222	05-27-2005	170.24	Adobe Acrobat Pro	Lenovo
WP09	59836	10-30-2005	35.00	Adobe Acrobat Capture	Microsoft
WP09	77740	05-27-2005	35.00	Adobe Acrobat Capture	Microsoft

Part 2:

5. TagNumber -> Computer Model
Package ID -> Package Name
PackageID, TagNumber -> InstallDate, SoftwareCostUSD

6. This table is not in third normal form because it has partial dependencies. The ComputerModel is only dependent on the TagNumber, and the PackageName is only dependent on the PackageID. This shows partial dependency because the attributes are not dependent on the entire primary key (PackageID, TagNumber.) This means that the table is not in 2nd normal form. If the table is not in 2nd normal form then it cannot be in 3rd normal form since the normalization steps build on one another.

Part 3:

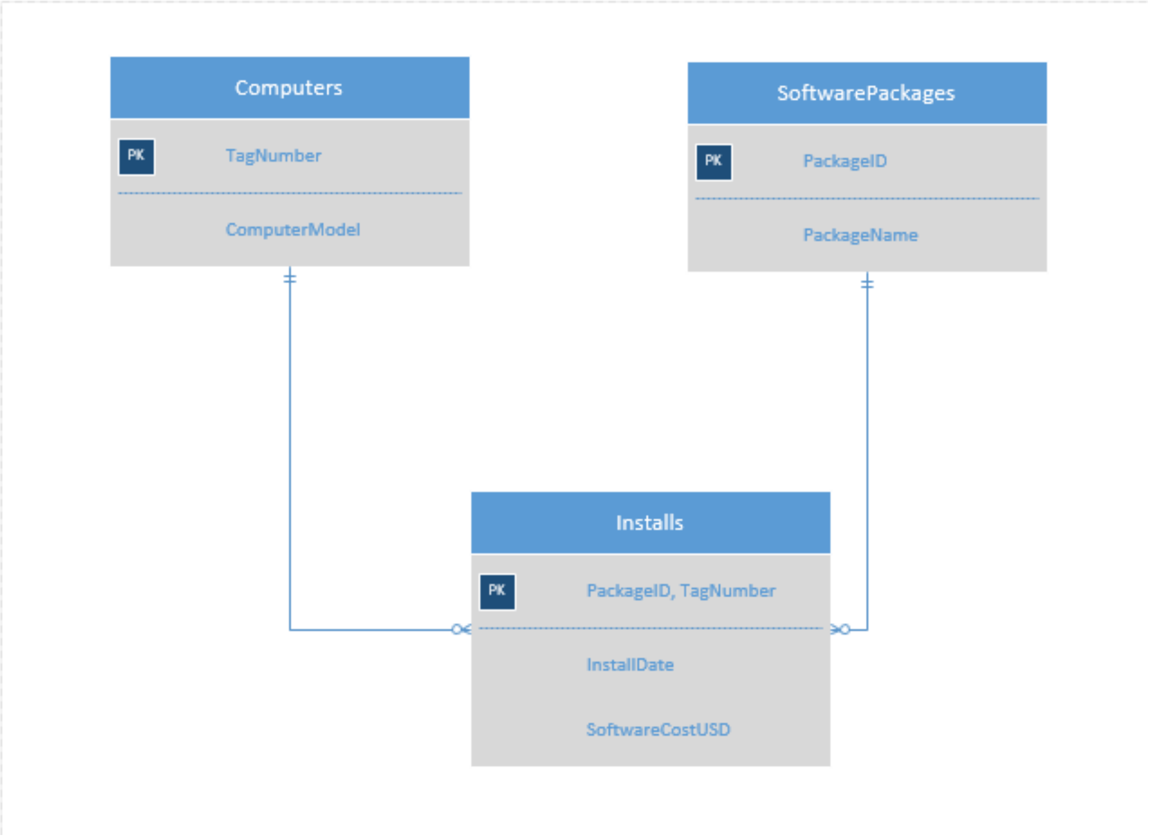
7. Once we normalize our table, it becomes 3 new separate tables. The primary keys of these tables are PackageID, TagNumber, and (PackageID, TagNumber.)

8. Computers Table: TagNumber -> Computer Model
Packages Table: Package ID -> Package Name
Installs Table: (PackageID, TagNumber) -> InstallDate, SoftwareCostUSD

9. The new tables created are in third normal form because they abide by the requirements for 1st, 2nd, and 3rd normal form. To explain this further, we see that the table is in 1st normal form because all the values in the fields are atomic and do not have internal structure. As well, the tables are in 2nd normal form because they don't have any partial dependencies. The columns in each table are dependent on their whole primary key. For example, you can not know the install date you are looking for without the packageID and the TagNumber of the computer. As well, you can only find the software cost by knowing the PackageID and the TagNumber. The other 2 tables do not have composite keys, thus it is impossible for them to have partial dependencies. Finally, the tables are in third normal form because there are no multi-key dependencies. This would only potentially be a problem in the Installs table, however it is not a problem because only the primary key leads to any of the other values. Given the InstallDate, the SoftwareCostUSD, or even both, you cannot get to the other values in the table. In the other tables this is not a problems

because it is clear that the dependencies only go one way. Thus, we have tables that are all in 3rd normal form.

10.


Computers
PK
TagNumber
ComputerModel
SoftwarePackages
PK
PackageID
PackageName
Installs
PK
PackageID, TagNumber
InstallDate
SoftwareCostUSD