Michael Wise
Dr. Alan Labouseur
CMPT 308 Database Management
5 November 2020

Lab #7: Normalization One

Part One:

1. Fred, buddy, listen - thank you for collecting this data, I think you have the right idea. However, I notice some problems in this spreadsheet that could lead to redundancy, especially once you add more data into here. The way you group PackageID isn't ideal. You only list it for the first entry/row of each software package, and leave it blank for any remaining machines that it is also installed on. What if I mixed the rows around or added more installations? How would you know which package has been installed on what system? I don't even know what software packages these are referring to. Fred, if you want to incorporate this into a database we should really normalize this table.

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	1-12-2006	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

First normal form - every intersection of a row & column is atomic.

3. The primary key is a composite key: (PackageID, TagNumber).

Part Two:

4. Add two columns of new data: one column for software package name and one for computer model. Be sure that your data is consistent with the original data.

PackageID	SoftwareName	TagNumber	ComputerModel	InstallDate	SoftwareCost USD
AC01	McAfee AntiVirus	32808	HP	09-13-2005	754.95
DB32	Oracle	32808	HP	12-03-2005	380.00
DB32	Oracle	37691	Apple	06-15-2005	380.00
DB33	MySQL	57772	Lenovo	05-27-2005	412.77
WP08	Discord	32808	HP	1-12-2006	185.00
WP08	Discord	37691	Apple	06-15-2005	227.50
WP08	Discord	57222	Dell	05-27-2005	170.24
WP09	Skype	59836	Lenovo	10-30-2005	35.00
WP09	Skype	77740	Asus	05-27-2005	35.00

5. Functional Dependencies:
    - (PackageID, TagNumber) → InstallDate, SoftwareCostUSD
        - The packageID & tagNumber together determine the date it was installed, as well as the total installation cost (different model computers can have different software costs for the same piece of software)
    - PackageID → SoftwareName
        - The name of the software depends on the software package ID
    - TagNumber → ComputerModel
        - The tag number determines the model of the computer

6. This table is **not** in third normal form because we have partial key dependencies (so it's not even in 2NF). SoftwareName and ComputerModel depend on PackageID and TagNumber respectively, which are both parts of the primary key (they are non-key attributes when accounted for separately). Thus attributes are not solely defined by the primary key but also non-key attributes. This could lead to update/delete anomalies.

Part Three:

Computers

TagNumber	ComputerModel
32808	HP
37691	Apple
57222	Dell
57772	Lenovo
59836	Lenovo
77740	Asus

Software

PackageID	SoftwareName
AC01	McAfee AntiVirus
DB32	Oracle
DB33	MySQL
WP08	Discord
WP09	Skype

Installations

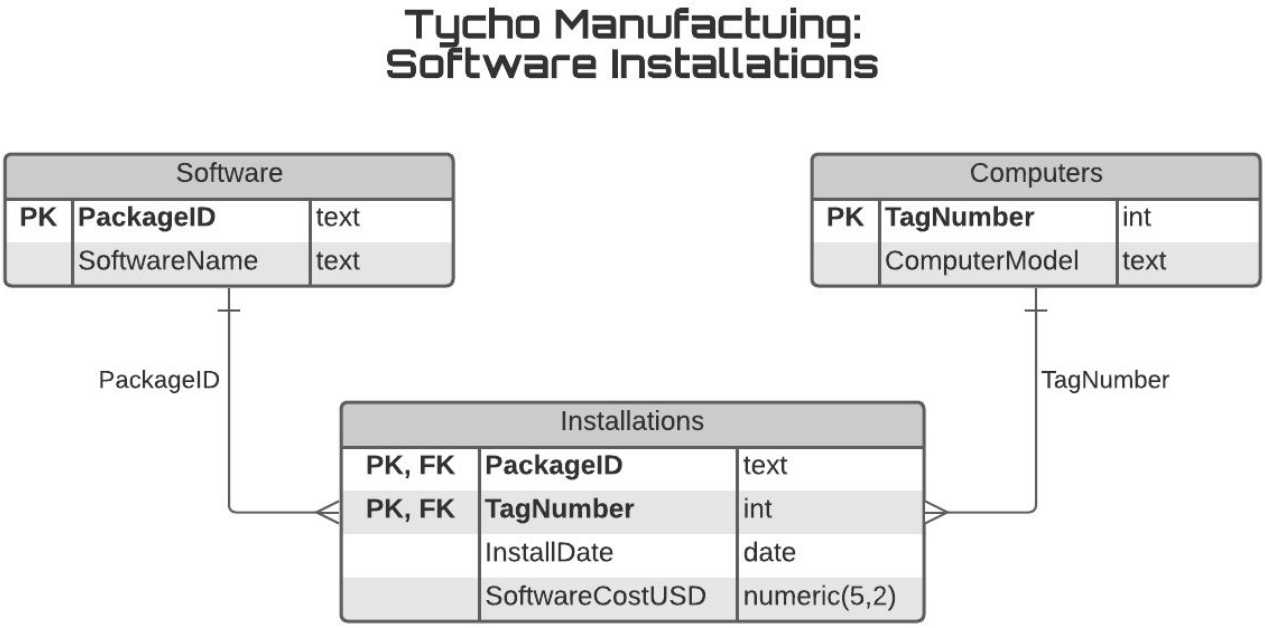
PackageID	TagNumber	InstallDate	SoftwareCostUSD
AC01	32808	09-13-2005	754.95
DB32	32808	12-03-2005	380.00
DB32	37691	06-15-2005	380.00
DB33	57772	05-27-2005	412.77
WP08	32808	1-12-2006	185.00
WP08	37691	06-15-2005	227.50
WP08	57222	05-27-2005	170.24
WP09	59836	10-30-2005	35.00
WP09	77740	05-27-2005	35.00

7. Primary Keys:
    - **Computers** Table: TagNumber
    - **Software** Table: PackageID
    - **Installations** Table: (PackageID, TagNumber)

8. Functional Dependencies:
    - (PackageID, TagNumber) → InstallDate, SoftwareCostUSD
    - PackageID → SoftwareName
    - TagNumber → ComputerModel

9. These tables **are** in third normal form because there are no partial key/multiple key dependencies. Every determinant is the key of a table. There are no non-prime attributes that determine other attributes. As you can see, each table only has one candidate key. This should help reduce update/delete anomalies that could have previously messed up our data.

10. Beautiful E/R Diagram:



Wonderful!