Lab 7

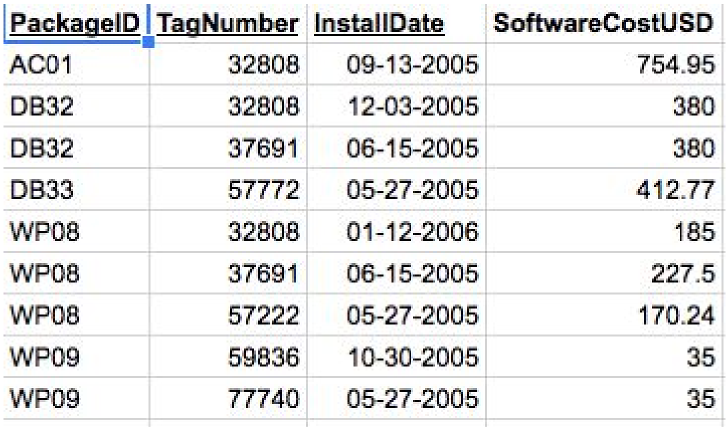
Thomas Magnusson

Part 1

1. Thank you for showing me this, Mr. Johnson. I love that you have been keeping track of all this data; seeing how you currently store your data will make it a lot easier to break it into an even safer, more robust form that will help you grow your business. Plus, we can make it so that you don't have to manually go into your spreadsheet and update the list by hand; we can make it so that your data is automatically updated, and preserved in a way that you can use flexibly in the future (in case you would like to expand your product lines, the types of computers you can install your software on, etc.). I appreciate the fact that you've labelled the software cost explicitly in USD; that gives your data appropriate context.

2. Table

Installations

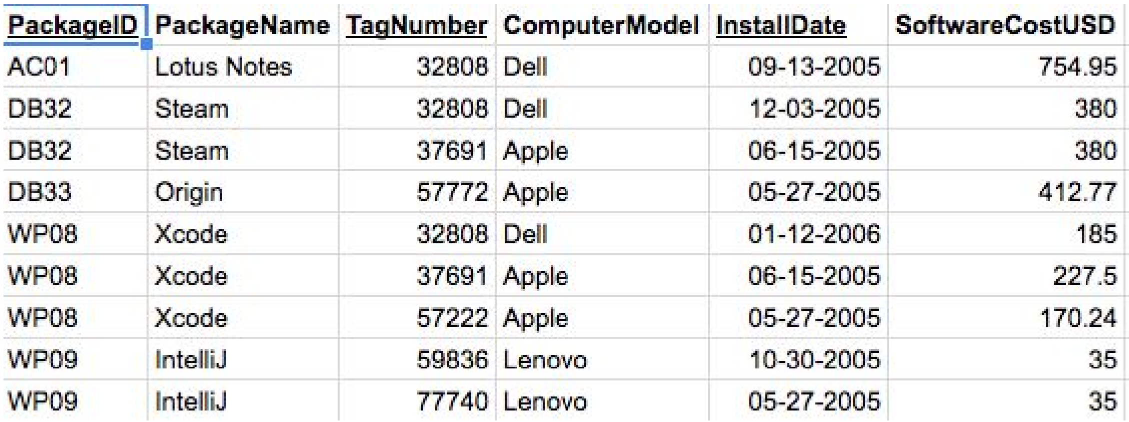


PackageID	TagNumber	InstallDate	SoftwareCostUSD
AC01	32808	09-13-2005	754.95
DB32	32808	12-03-2005	380
DB32	37691	06-15-2005	380
DB33	57772	05-27-2005	412.77
WP08	32808	01-12-2006	185
WP08	37691	06-15-2005	227.5
WP08	57222	05-27-2005	170.24
WP09	59836	10-30-2005	35
WP09	77740	05-27-2005	35

3. Primary key: underlined in the table. In Installations the primary key is the composite of (PackageID, TagNumber, InstallDate) because I'm assuming that the same package can be reinstalled on the same machine at different dates. The software cost of the installations are derived from the composite primary key and doesn't make sense to be a part of the primary key. If it were, it would enable the same package being installed on the same machine on the same day, but it would have to cost differently.

Part 2

4.



PackageID	PackageName	TagNumber	ComputerModel	InstallDate	SoftwareCostUSD
AC01	Lotus Notes	32808	Dell	09-13-2005	754.95
DB32	Steam	32808	Dell	12-03-2005	380
DB32	Steam	37691	Apple	06-15-2005	380
DB33	Origin	57772	Apple	05-27-2005	412.77
WP08	Xcode	32808	Dell	01-12-2006	185
WP08	Xcode	37691	Apple	06-15-2005	227.5
WP08	Xcode	57222	Apple	05-27-2005	170.24
WP09	IntelliJ	59836	Lenovo	10-30-2005	35
WP09	IntelliJ	77740	Lenovo	05-27-2005	35

5. Functional Dependencies

PackageID → PackageName

PackageID determines package name. Different software packages could have the same name (Assumption).

(PackageID, TagNumber, InstallDate) → SoftwareCostUSD

PackageID, TagNumber and InstallDate all determine the cost of a given Installation. This makes sense because Mr. Johnson's business model is pricing per software package per machine on a particular day.

TagNumber → ComputerModel

Each computer is exactly one type of model. Many computers are part of one model (which is why the functional dependency is not the other way).

6. Why this table is NOT in 3NF

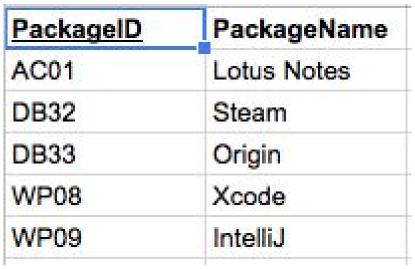
The table is not in 3NF because it is not in 2NF. It is not in 2NF because there exist non-key attributes that depend on only a portion of the primary key. For example, PackageName only depends on PackageID, but the primary key of the entire table includes not only PackageID but TagNumber and InstallDate. Since PackageName does not depend on InstallDate, the table violate 2NF. Since it violates 2NF, it violates 3NF.

Part 3

7. Tables

SoftwarePackages

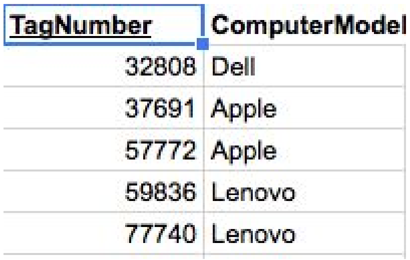
PackageID



PackageID	PackageName
AC01	Lotus Notes
DB32	Steam
DB33	Origin
WP08	Xcode
WP09	IntelliJ

Computers

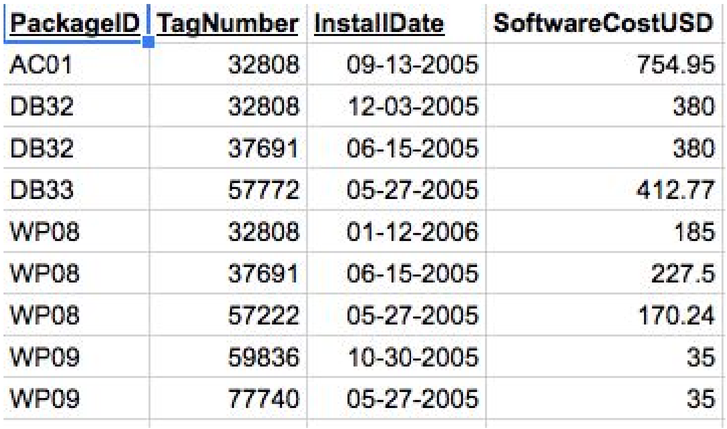
TagNumber



TagNumber	ComputerModel
32808	Dell
37691	Apple
57772	Apple
59836	Lenovo
77740	Lenovo

Installations

(PackageID, TagNumber, InstallDate)



PackageID	TagNumber	InstallDate	SoftwareCostUSD
AC01	32808	09-13-2005	754.95
DB32	32808	12-03-2005	380
DB32	37691	06-15-2005	380
DB33	57772	05-27-2005	412.77
WP08	32808	01-12-2006	185
WP08	37691	06-15-2005	227.5
WP08	57222	05-27-2005	170.24
WP09	59836	10-30-2005	35
WP09	77740	05-27-2005	35

8. Functional Dependencies

SoftwarePackages: PackageID → PackageName

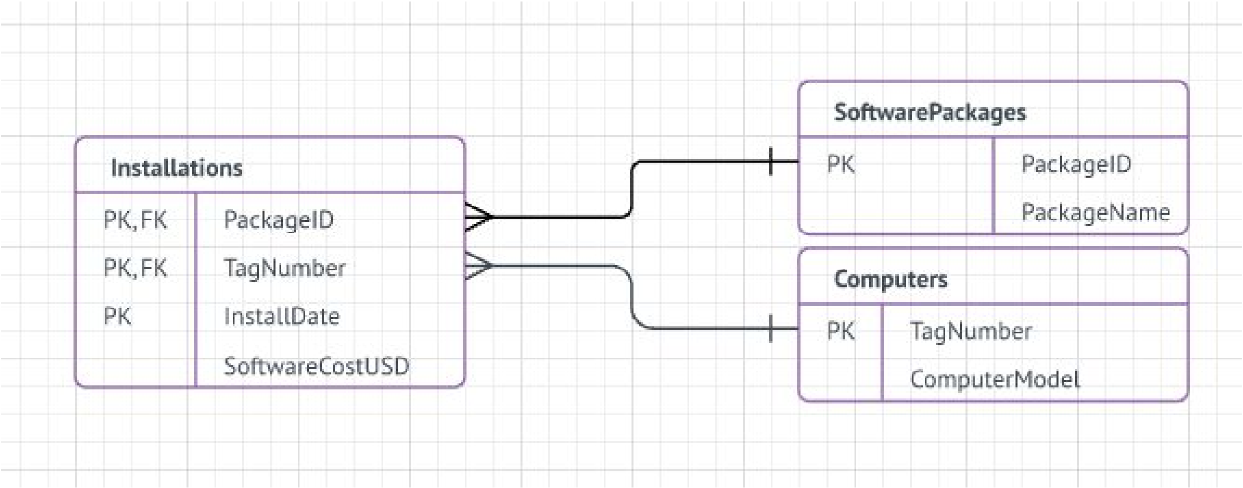
Computers: TagNumber → Computer Model

Installations: (PackageID, TagNumber, InstallDate) → SoftwareCostUSD

9. Why 3NF

The database is in 1NF because the intersection of each column and row is atomic, that is to say there are no reasonable ways to further break any columns into more columns. The database is in 2NF because it's in 1NF and there exist no partial key dependencies. Each nonkey attribute depends on every part of the primary key in its table. The database is in 3NF because it's in 2NF and there exist no multikey dependencies. The nonkey attributes of each table depend on nothing but the entire primary key of that table.

10. E/R Diagram


Installations
PK,FK PackageID
PK,FK TagNumber
PK InstallDate
SoftwareCostUSD
SoftwarePackages
PK PackageID
PackageName
Computers
PK TagNumber
ComputerModel