Tadd Bindas

Lab 7

Database Management

Part 1)

1) This data is a mess. It is not even in first normal form, and it makes no sense when reading. The reader has no true idea of what is the primary key and each value is not atomic.

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

The primary key is TagNumber-PackageID

Part 2)

TagNumber-PackageID	SoftwareCostUSD	InstallDate	SoftwarePackageName	ComputerModel
AC01-32808	754.95	09-13-2005	Zork	IBM
DB32-32808	380.00	12-03-2005	Portal	Apple
DB32-37691	380.00	06-15-2005	Chrome	Apple
DB33-57772	412.77	05-27-2005	FireFox	Apple
WP08-32808	185.00	01-12-2006	IE8	IBM
WP08-37691	227.50	06-15-2005	MW2	IBM
WP08-57222	170.24	05-27-2005	Call Of Duty	IBM
WP09-59836	35.00	10-30-2005	QuantumofSolace.mp4	Apple
WP09-77740	35.00	05-27-2005	TheSpyWhoShaggedMe.mp4	Apple

SoftwarePackageName and ComputerModel have a functional Dependency with SoftwareCostUSD. The software name/the computer model determine the price that it will cost for the software. There is also a functional dependency that SoftwareCostUSD is functionally dependent on SoftwarePackageName.

This will never be in third normal form because it does not follow 2NF. Second normal form is when a program is in 1NF and has no partial key dependencies. Since there is a partial key dependency at {SoftwarePackageName, ComputerModel} -> SoftwareCostUSD and {SoftwarePackageName} -> SoftwareCostUSD there will never be 2NF, therefore never 3NF.

Part 3)

I have three tables. I have a table for Software Orders where TagNumber-Package order is the Primary Key. I have another table for Software Pricing where SoftwarePackageName is the Primary Key. Lastly, I have a table for software models where SoftwarePackageName is the Primary Key.

For my functional dependencies, SoftwareCostUSD depends on the SoftwarePackageName. Those are the only functional dependencies in my tables.

The new tables are in third normal form because all values are atomic, there are no mulit-key dependencies, and there are no partial key dependencies. It is atomic because all values are in their simplest forms. It has no partial key dependencies because nowhere {A,B} → {C} but also {A} → {C} then {C} is partially functionally dependent on {A,B}. There are no mulit-key dependencies because there are not multiple candidate keys in each table.