1) There is inconsistency in the design of the graph. For example, one software package AC01 is associated with 1 tag number and one cost value, whereas another DB32 is associated with 2. The design is not consistent across all fields. It is not in normal form since 1 column entity can not have multiple values, as is the case here.

“The table looks great, and it appears as though you’ve put a lot of effort into it. I can, however, identify some consistency issues and I can use my expertise to make it even better.

I’m really looking forward to working in an environment filled with minds capable of forming such a brilliant prototype.”

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

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

II)

TagNumber	Model	PackageId	PackageName	InstallDate	SoftwareCost USD
32808	Alienware	AC01	gcc	09-13-2005	754.95
32808	Alienware	DB32	portal	12-03-2005	380.00
32808	Alienware	WP08	firefox	01-12-2006	185.00
37691	Dell	DB32	portal	06-15-2005	380.00
37691	Dell	WP08	firefox	06-15-2005	227.50
57772	Mac	DB33	postgresql	05-27-2005	412.77
57222	Solaris	WP08	firefox	05-27-2005	170.24
59836	HP	WP09	Word	10-30-2005	35.00
77740	IBM	WP09	Word	05-27-2005	35.00

Functional Dependencies:

TagNumber -> ModelNumber
PackageID -> PackageName
(TagNumber, PackageId) -> InstallDate
(TagNumber, PackageId) -> SoftwareCostUSD

This table can not be said to be in third normal form because in a 3rd normal form table, no non-primary key value may depend on another non primary key value. You could not enter a new entity to this table in practice unless that entity had a valid value for every column field in the table. In practice you might want to add a new computer with a known software package installed with a known cost but an unknown install date. This would result in an insertion anomaly, because the data could not be added to the table.

In a 3rd normal form table, no entity entered with certain fields can rely on the existence of other fields.

III)

Computers

TagNumber	Model
32808	Alienware
37691	Dell
57772	Mac
57222	Solaris
59836	HP MaSc
77740	IBM

Software Packages

PackageId	PackageName
AC01	gcc
DB32	portal
DB33	postgresql
WP08	firefox
WP09	Word

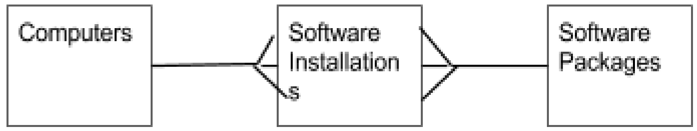
Package Installations

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

1)TagNumber
2)PackageId
3)(TagNumber,PackageId)

TagNumber -> Model
PackageID -> PackageName
(TagNumber, PackageId) -> InstallDate
(TagNumber, packageid) -> SoftwareCostUSD

The resulting tables are in 3rd normal form because you can now enter new entities to the database with only the fields that are provided. No one field depends on any other field other than a primary key. data can now be entered into the database when it lacks data pertaining to certain column in the software installation table. For instance, software packages that were never installed can be entered into the database without a price or installation date. Individual rows can also be deleted. Getting rid of the possibility of an anomaly.


Computers
Software Installation s
Software Packages