Part One

1) (In a nice way) I would tell him in needs work, and that all of the normal form rules are violated. So, we need to clean it up so that we can fix this. I would also say that it is smarter to sub-divide the PackageID's into seperate rows, instead of the chunk format that he has. Lastly, I would suggest that we figure out how we would create multiple tables, so that we can have a relational DB structure, as it is just the best.

(using # symbol to fit table properly)

2)

PackagID	Tag#	Install Date	SoftwareCost USD
ACO1	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

3) PackageID, Tag# → Composite PK

Part Two

4)

PackagID	Tag#	Install Date	SoftwareCostUSD	PkgName	CMPTModel
AC01	32808	09-13-2005	754.95	SPSSModler	IBM
DB32	32808	12-03-2005	380.00	PostGreSQL	IBM
DB32	37691	06-15-2005	380.00	PostGreSQL	Apple
DB33	57772	05-27-2005	412.77	WordPro	Dell
WP08	32808	01-12-2006	185.00	OracleDB	IBM
WP08	37691	06-15-2005	227.50	OracleDB	Apple
WP08	57222	05-27-2005	170.24	OracleDB	ASUS
WP09	59836	10-30-2005	35.00	TensorFlow	Sony
WP09	77740	05-27-2005	35.00	TensorFlow	HP

5) PackageID → PkgName

Tag# → CMPTModel

PackageID, Tag# → InstallDate, SoftwareCostUSD

6) This new table is not in 3NF as transitive dependencies exist, as well as partial key dependencies → Since there are partial key dependencies, 2NF is violated. Thus the table is not in 3NF for more than just one reason.

For example, you only need the PackageID to determ

ine PkgName, but since we determined that a composite key exists (as shown in Q5), PkgName only depends on part of this key, not the entirety of it. This is the partial dependency I was referring to, so 2NF is already violated and thus so is 3NF. Now let's dive into the existing transitive dependency.

Using the other part of the composite key this time, note that CMPTModel depends on Tag# which is part of (PackageID, Tag#) (the PK of this poor table). So, to draw it out, we have:

(PackageID, Tag #) → Tag# → CMPTModel

a clear transitive dependency.

Part Three:

Package

PackageID	PkgName
AC01	SPSSModler
DB32	PostGreSQL
DB33	WordPro
WP08	OracleDB
WP09	TensorFlow

Computer

Tag#	CMPTModel
32808	IBM
37691	Apple
57772	Dell
57222	ASUS
59836	Sony
77740	HP

Installations

PackageID	Tag#	InstallDate	SoftwareCostUSD
ACO1	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

7) Primary Keys (Determinants):

Package Table → PackageID

Computer Table → Tag Number

Installations Table → (PackageID, Tag #)

8) Functional Dependencies

Package Table: PackageID → PkgName

Computer Table: Tag# → CMPTModel

Installations Table: (PackageID, Tag#) → InstallDate, SoftwareCostUSD

9) These new tables are now in 3NF as all of our Partial key and transitive dependencies have been properly dealt w/. Now that we have 3 seperate tables instead of one terrible table to rule them all, redundancy will not be an issue, and our non-key attributes depend on nothing but the key (their respective PK's).