Part 1:

1) The first thing I would ask is the compensation for the project. The second thing I would say is that there is only one large table that "rules all." It is easy to mess up the table and cause incorrect and inconsistent data through insert, update, and delete anomalies. Insert anomaly occurs when certain attributes cannot be inserted into the database without the presence of other attributes, causing us to add some placeholder data. In this case, if a new software is not installed on one of the computers, then there would be no installation date, without which, we cannot add the software into the table to track it. Update anomaly causes a change in one conceptual piece of data in more than one place. In this example, if a computer with a tag number 32808 has to be changed, then we would need to change it at least three times as of the current state of the table. This might cause a lot of problems and inconsistency. Delete anomaly means that we lose unrelated data when we delete something that we intended to delete. In this case, if we want to delete software WP08, then we end up with losing all three related tag numbers, their installation dates and cost. In order to avoid all these anomalies, we have to implement normalization – to break one large table into many small ones with reversibility and lossless join property.

2) The data in 1NF:

PackageID	TagNumber	InstallDate	SoftwareCostUSD
AC01	32808	9/13/2005	754.95
DB32	32808	12/3/2005	380.00
DB32	37691	6/15/2005	380.00
DB33	57772	5/27/2005	412.77
WP08	32808	1/12/2006	185.00
WP08	37691	6/15/2005	227.50
WP08	57222	5/27/2005	170.24
WP09	59836	10/30/2005	35.00
WP09	77740	5/27/2005	35.00

3) Primary key in the table would be a combination of PackageID and TagNumber. Individually, both the PackageId and TagNumber repeat (therefore, not unique), but by taking them together, they compose a unique combination for each row. Logically, it also works: a specific software can be installed on a specific computer only once.

Part 2:

1)

PackageID	TagNumber	SoftwareName	ComputerModel	InstallDate	SoftwareCostUSD
AC01	32808	Zork	Apple	9/13/2005	754.95
DB32	32808	HomeBase	Apple	12/3/2005	380.00
DB32	37691	HomeBase	Dell	6/15/2005	380.00
DB33	57772	HomeBaseNew	IBM	5/27/2005	412.77
WP08	32808	eSmartFile	Apple	1/12/2006	185.00
WP08	37691	eSmartFile	Dell	6/15/2005	227.50
WP08	57222	eSmartFile	Lenovo	5/27/2005	170.24
WP09	59836	TWDesktop	Asus	10/30/2005	35.00
WP09	77740	TWDesktop	Toshiba	5/27/2005	35.00

2) (PackageID, TagNumber) ⟶ SoftwareName, ComputerModel, InstallDate, SoftwareCostUSD

3) 3 NF states that there are no multiple key dependencies, that is, there are no attributes that are dependent on more than one key. In the table and dependency above, it is clear that the requirement is not met, because SoftwareName is dependent on PackageID and ComputerModel is dependent on TagNumber. InstallDate and SoftwareCostUSD are both dependent on both PackageID and TagNumber:
PackageID ⟶ SoftwareName
TagNumber ⟶ ComputerModel
(PackageID, TagNumber) ⟶ InstallDate, SoftwareCostUSD

This arrangement shows that we need to create 3 tables to meet the requirements of the 2 and 3 NF rules.

Part 3:

PackageID	SoftwareName
AC01	Zork
DB32	HomeBase
DB32	HomeBase
DB33	HomeBaseNew
WP08	eSmartFile
WP08	eSmartFile
WP08	eSmartFile
WP09	TWDesktop
WP09	TWDesktop

PackageID ⟶ SoftwareName

Primary key in this table is PackageID.

PackageID is the determinant, it functionally determines SoftwareName.

Here, SoftwareName is only dependent on PackageID, there are no multiple dependencies, and therefore, the table is in the 3NF.

It is a strong entity, it has an artificial primary key.

Liliya Yerassilova Normalization Lab 1 03/22/2016

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

TagNumber ⟶ ComputerModel

Primary key in this table is TagNumber.

TagNumber is the determinant, it functionally determines ComputerModel.

Here, ComputerModel is only dependent on TagNumber, there are no multiple dependencies, and therefore, the table is in the 3NF.

It is also a strong entity with an artificial primary key.

PackageID	TagNumber	InstallDate	SoftwareCostUSD
AC01	32808	9/13/2005	754.95
DB32	32808	12/3/2005	380.00
DB32	37691	6/15/2005	380.00
DB33	57772	5/27/2005	412.77
WP08	32808	1/12/2006	185.00
WP08	37691	6/15/2005	227.50
WP08	57222	5/27/2005	170.24
WP09	59836	10/30/2005	35.00
WP09	77740	5/27/2005	35.00

(PackageID, TagNumber) ⟶ InstallDate, SoftwareCostUSD

Primary key in this table is a combination of PackageID and TagNumber, it is a composite primary key.

PackageID and TagNumber is the determinant, it functionally determines InstallDate and SoftwareCostUSD.

Here, both InstallDate and SoftwareCostUSD are dependent on the entire (composite) primary key, there are no multiple dependencies, and therefore, the table is in the 3NF.

This is a weak (associative) entity, and has a composite primary key.

E / R Diagram: