Shea DeCaro

Part One:

1. As he shows you the spreadsheet, having just signed your consulting agreement, he asks what you think of it. How do you reply?
    a. I would reply that it is a good start to organizing the tracking information for their software packages, but that it needs a bit of expanding upon since the information isn't easily distinguishable and they're missing the other key information they requested me to add.

2. Put his data in 1NF and display it. Show the table

FIRST NORMAL FORM

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. What is the primary key?
    a. The primary key is a combination of PackageID, TagNumber, and InstallDate. SoftwareCostUSD wouldn't be included because it doesn't really make any of the other information unique when we add it.

Part Two:
Add two columns of new data. One column for software package name (e.g., Zork, Portal, etc.) and one for computer model (e.g., IBM, Apple, etc.). Be sure that your new data is consistent with the old data.

1. Display the new table

PackageID	TagNumber	InstallDate	SoftwareCostUSD	PackageName	ComputerModel
AC01	32808	09-13-2005	754.95	Zork	IBM
DB32	32808	12-03-2005	380	Portal	IBM
DB32	37691	06-15-2005	380	Portal	Apple
DB33	57772	05-27-2005	412.77	Twine	Microsoft
WP08	32808	01-12-2006	185	SugarCube	IBM
WP08	37691	06-15-2005	227.5	SugarCube	Apple
WP08	57222	05-27-2005	170.24	SugarCube	Amiga
WP09	59836	10-30-2005	35	AGS	Dell
WP09	77740	05-27-2005	35	AGS	HP

2. Identify and document all functional dependencies
    a. The functional dependencies are as follows:
        i. Package ID determines Package Name
        ii. Tag Number determines Computer Model
        iii. Primary Key determines the cost

3. Explain why this new table is not in third normal form.
    a. The new table is not in 3NF because the new columns only put it in 2nd normal form. 3rd normal form would mean that there are no partial dependencies and no transitive dependencies. But, the software costs depend on the whole primary key, and our transitive dependencies consist of the software package name being dependent on **only** the packageID**, not the entire primary key.**

Part Three:
Decompose your 1NF table into a set of tables that are in at least third normal form. (BCNF would be better.) Remember that it's wrong to add artificial keys to associative identities. Actually, as I said before, do not add any additional columns.

1. Identify all primary keys (determinants) for all tables.
    a. The primary keys for each table are the PackageID, Tag Number, or a combination of the packageID, TagNumber and Install Date.

2. Identify all functional dependencies of each table.
    a. Table 1:

i. PackageID determines the Software Package Name

PackageID (PK)	PackageName
AC01	Zork
DB32	Portal
DB32	Portal
DB33	Twine
WP08	SugarCube
WP08	SugarCube
WP08	SugarCube
WP09	AGS
WP09	AGS

b. Table 2:
   i. The Tag Number determines the exact Computer Model

TagNumber (PK)	ComputerModel
32808	IBM
32808	IBM
37691	Apple
57772	Microsoft
32808	IBM
37691	Apple
57222	Amiga
59836	Dell
77740	HP

c. Table 3:
   i. The combination of package id, tag number, and install date determines the cost of the installation.

PackageID (PK)	TagNumber (PK)	InstallDate (PK)	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. Explain why the new tables are in third normal form.
    a. The tables are in third normal form because there are no partial dependencies and no transitive dependencies. The first two tables both have single attribute primary keys, and the third table is set up so that the software cost is completely dependent on the rest of the table.

4. Draw beautiful E/R diagram using LucidChart