__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. He has all of his data in one location, which is the spreadsheet. The way the data is stored is unnormalized. The disadvantages of unnormalized data are:

- Insert anomalies: If there is any data within the spreadsheet that needs to have a "not null" constraint, it could lead to an insert anomaly. One of the goals of hiring us as a consultant is to keep track of software packages installed on their computers. This would imply that TagNum is an important field since it tells us what software package is installed on a machine identified by the TagNum. This makes TagNum a "not null" field. However, what if there is a situation where we don't know the TagNum for a particular PackID. It can lead to incomplete data being stored like the following example:

PackID	**TagNum**	**InstallDate**	**SoftwareCost**
AC01	unknown	09-13-1995	754.95

- Update anomalies: The data stored in the spreadsheet is not atomic. What this means is that there are multiple lines containing data for the same PackID. If we want to make an update for PackID WP08, there are 3 spots we potentially have to update.

- Delete anomalies: If he decided to delete the first row of the spreadsheet, because TagNum 32308 no longer has PackID AC01 installed, he would completely eliminate the fact that PackID AC01 was installed on their computers at one time. This defeats the purpose of putting together a tracking system which tells them what software was installed on their machines.

In summary, unnormalized data can lead to a lot of inconsistently stored data and in the worst case, lead to loss of data. Normalizing the data would solve this major problem. So the first step would be to take CEO Meservy's spreadsheet data and put it into a normalized relational structure.

2. Put his data in 1NF and display it.

A. **PackageDetails**

PackID
AC01
DB32
DB33
WP08
WP09

ComputerDetails

TagNum	PackID	InstallDate	SoftwareCost
32808	AC01	09-13-1995	754.95
32808	DB32	12-03-1995	380.00
37691	DB32	06-15-1995	380.00
57772	DB33	05-27-1995	412.77
32808	WP08	01-12-1996	185.00
37691	WP08	06-15-1995	227.50
57222	WP08	05-27-1995	170.24
59836	WP09	10-30-1995	35.00
777740	WP09	05-27-1995	35.00

3. What is the primary key ?

A. The PackID field is the primary key

<u>Part Two</u>

Add two columns of new data: one for software package name (e.g., Zork, Portal, etc) and one for computer model (e.g, HP, Apple etc). Be sure that your new data is consistent with the original data. Do not add any additional columns.

Display the new table

PackageDetails

PackID	PackName
AC01	Zork
DB32	Portal
DB33	Not Lotus Notes
WP08	Open Office
WP09	MS SQl Server

ComputerDetails

TagNum	PackID	InstallDate	SoftwareCost	ComputerModel
32808	AC01	09-13-1995	754.95	HP pavilion
32808	DB32	12-03-1995	380.00	HP pavilion
37691	DB32	06-15-1995	380.00	Lenovo
57772	DB33	05-27-1995	412.77	Mac Air
32808	WP08	01-12-1996	185.00	HP pavilion
37691	WP08	06-15-1995	227.50	Lenovo
57222	WP08	05-27-1995	170.24	Mac Air
59836	WP09	10-30-1995	35.00	Samsung
777740	WP09	05-27-1995	35.00	Sony Vaio

Identify and document all the functional dependencies

PackID -> PackName is a functional dependency because one particular PackID only maps to one particular PackName. Also, putting in one value results in the output of one value.

TagNum -> ComputerModel is a functional dependency because one particular TagNum only refers to one particular computer which can only be one particular model. Putting in one value results in the output of one value.

Explain why this new table is not in third normal form

In order to be in third normal form, every non-key attribute must provide a fact about the key, the whole key, and nothing but the key. In the example above, the non-key attributes InstallDate, SoftwareCost and ComputerModel don't really give us any information about the primary key, which is PackID. Also, SoftwareCost might be transitively dependent on the composite key TagNum, PackID based on InstallDate and ComputerModel. However, one can enter mis-information for InstallDate and ComputerModel leading to inconsistencies.

<u>Part Three</u>

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 entities.

As explained above, in order to be in third normal form every non-key attribute must provide a fact about the key, the whole key, and nothing but the key. Given this, putting the abvove 1NF tables into a set of 3NF tables would lead to the following structure:

PackageDetails

PackID	**PackName**
AC01	Zork
DB32	Portal
DB33	Not Lotus Notes
WP08	Open Office
WP09	MS SQl Server

ComputerDetails

TagNum	**ComputerModel**
32808	HP pavilion
37691	Lenovo
57772	Mac Air
59836	Samsung
777740	Sony Vaio

InstallDetails

InstallID	**TagNum**	**PackID**	**InstallDate**	**SoftwareCost**
1	32808	AC01	09-13-1995	754.95
2	32808	DB32	12-03-1995	380.00
3	37691	DB32	06-15-1995	380.00
4	57772	DB33	05-27-1995	412.77
5	32808	WP08	01-12-1996	185.00
6	37691	WP08	06-15-1995	227.50
7	57222	WP08	05-27-1995	170.24
8	59836	WP09	10-30-1995	35.00

9	777740	WP09	05-27-1995	35.00

Identify all primary keys (determinants) for all tables

PackageDetails:	PackID
ComputerDetails:	TagNum
InstallDetails:	InstallID

Identify all functional dependencies for all tables

Given a package ID we can determine the name of a software package that is being installed

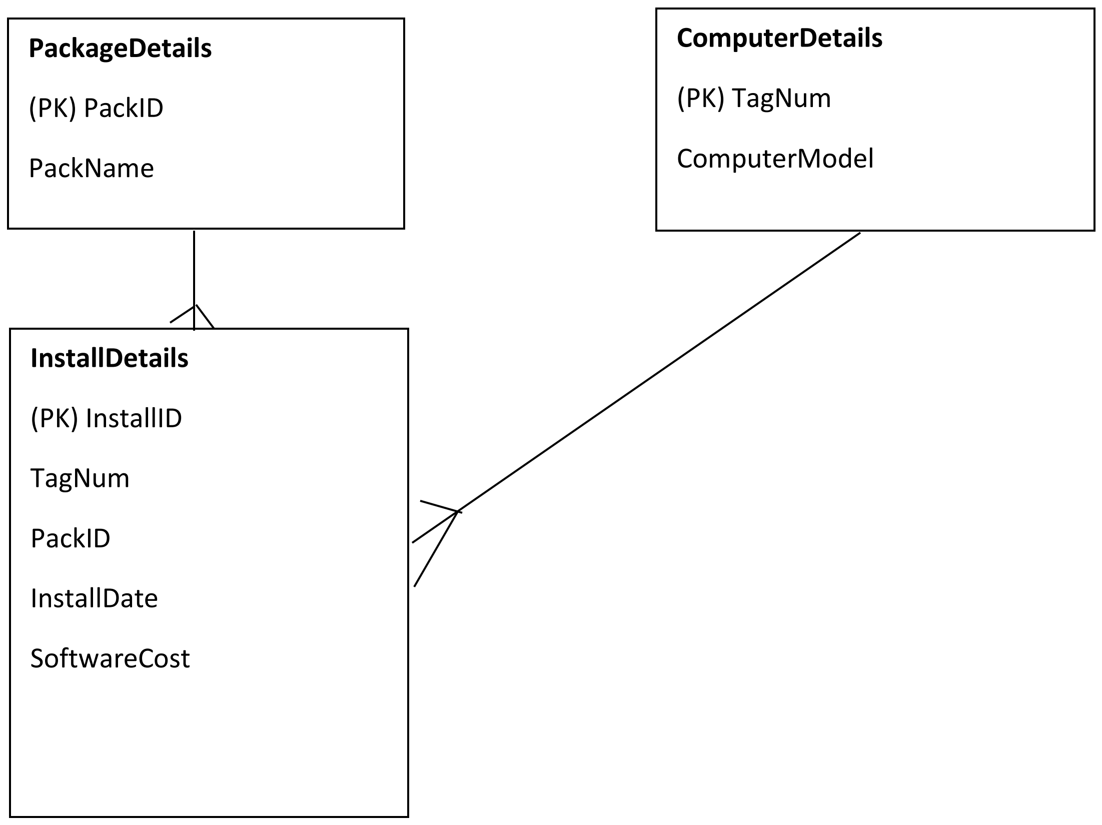
Given a tag number, we can determine details about the machine on which an install was done

Given an install ID, we can determine details about the install including the machine it was installed in, date of install, the software package that was installed and cost

Explain why the tables are in 3NF

In 3 NF, information in the tables is strictly dependent on the key and nothing but the key (so help me Codd). In this case for each table identified above, row data is unique and determined based on the key for that row.

Draw an E/R diagram


PackageDetails
(PK) PackID
PackName
ComputerDetails
(PK) TagNum
ComputerModel
InstallDetails
(PK) InstallID
TagNum
PackID
InstallDate
SoftwareCost