Melissa Iori
Lab 7

Part One:

1.) While the names are good, the intersections of rows and columns are not atomic which can lead to inconsistency as the table grows. With normalization we can manage this information in a more consistent way.

2.)

PackID	TagNum	InstallDate	SoftwareCost
AC01	32808	09-13-1995	754.95
DB32	32808	12-03-1995	380.00
DB32	37691	06-15-1995	380.00
DB33	57772	05-27-1995	412.77
WP08	32808	01-12-1996	185.00
WP08	37691	06-15-1995	227.50
WP08	57222	05-27-1995	170.24
WP09	59836	10-30-1995	35.00
WP09	77740	05-27-1995	35.00

3.) There are no unique fields here, so the primary key must be the minimal super key. (TagNum, SoftwareCost), though unique in this snapshot, would be a bad choice for primary key, as packages of the same cost could be installed on the same machine.

A good candidate would be a composite key of (PackID, TagNum) although it assumes that a software package would only be installed onto a machine once, not taking into account uninstalls/reinstalls. In terms of the business logic, (PackID, TagNum) should be suitable, because the company is probably only paying for the actual software license on that particular machine just once. If the software became corrupted and had to be re-installed, it would likely be at no cost, depending on the warranty, and wouldn't really count as an install for the purposes of this table. Anyway, (PackID, TagNum) is the primary key.

Part Two:

4.) (table is split over two pages)

PackID	TagNum	InstallDate	SoftwareCost	PackName	ComputerModel
AC01	32808	09-13-1995	754.95	Photoshop	HP
DB32	32808	12-03-1995	380.00	DB2	HP
DB32	37691	06-15-1995	380.00	DB2	Apple
DB33	57772	05-27-1995	412.77	SQL Server	Lenovo

WP08	32808	01-12-1996	185.00	Zork	HP
WP08	37691	06-15-1995	227.50	Zork	Apple
WP08	57222	05-27-1995	170.24	Zork	Toshiba
WP09	59836	10-30-1995	35.00	Zork	Dell
WP09	77740	05-27-1995	35.00	Zork	Apple

5.) (PackID, TagNum) → SoftwareCost
(PackID, TagNum) → InstallDate
PackID → PackName: Given a PackID, you get back one PackName for that PackID.
TagNum → ComputerModel: Given a TagNum, you get back one ComputerModel for that machine.

(PackName is not a unique identifier, so there is no PackName → PackID. PackName could be Zork for both WP08 and WP09, but WP08 could be version 1.0 and WP09 could be version 1.1. The version isn't stated in the PackName.)

6.) A table in third normal form must also satisfy second normal form, and a table in second normal form has no partial key dependencies. In this table, PackName depends only on PackID, a single part of the primary key, so it has no need to be in this table. ComputerModel depends only on TagNum, and it also does not need to be here.

Part Three:

Installations:

PackID	**TagNum**	**InstallDate**	**SoftwareCost**
AC01	32808	09-13-1995	754.95
DB32	32808	12-03-1995	380.00
DB32	37691	06-15-1995	380.00
DB33	57772	05-27-1995	412.77
WP08	32808	01-12-1996	185.00
WP08	37691	06-15-1995	227.50
WP08	57222	05-27-1995	170.24
WP09	59836	10-30-1995	35.00
WP09	77740	05-27-1995	35.00

SoftwarePackages: (table split over two pages)

PackID	**PackName**
AC01	Photoshop

DB32	DB2
DB33	SQL Server
WP08	Zork
WP09	Zork

Computers:

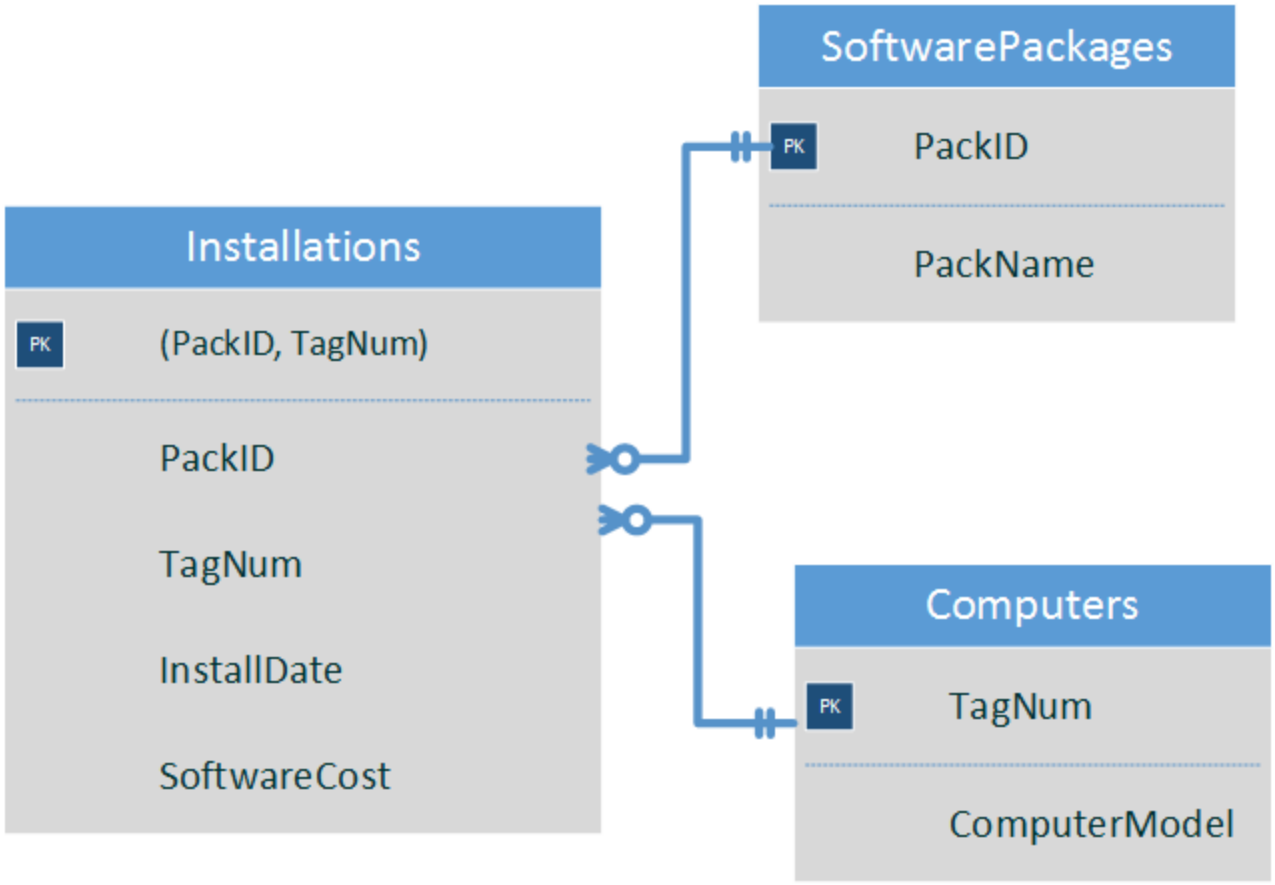
TagNum	**ComputerModel**
32808	HP
37691	Apple
57772	Lenovo
57222	Toshiba
59836	Dell
77740	Apple

7.) Primary keys:
Installations: (PackID, TagNum)
SoftwarePackages: PackID
Computers: TagNum

8.) Functional Dependencies:
Installations: (PackID, TagNum) → SoftwareCost
(PackID, TagNum) → InstallDate
SoftwarePackages: PackID → PackName
Computers: TagNum → ComputerModel

9.) The tables are in third normal form. Firstly, they are entirely atomic, satisfying first normal form. Secondly, there are no partial key dependencies, satisfying second normal form. And finally, there are no multiple key dependencies. All dependencies are on the primary key for each table.

10.) E/R diagram is below on the next page...


SoftwarePackages
PK
PackID
PackName
Installations
PK
(PackID, TagNum)
PackID
TagNum
InstallDate
SoftwareCost
Computers
PK
TagNum
ComputerModel