Jackson O'Brien

3/26/2024

1. The table provided is relatively ready to be implemented, however there is no solid primary key. There is also multiple duplicate data and that PackageID also holds multiple rows under one row, which is not compatible with creating a database. The way to fix this could be to separate each row and have a duplicate PackageID in the column, but not the row. If you select everything from that PackageID or TagNumber it will show all information related to them.
2.

PackageID	TagNumber	InstallDate	SoftwareCostUSD
AC01	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	11-12-2006	185.00
WP08	37961	06-15-2005	227.50
WP09	57222	05-27-2005	170.24
WP09	57740	05-27-2005	35.00

3. There is no good primary key, but for the time being PackageID or TagNumber would be the best candidate for one. I would create a new column for a new type of ID of integers that is unique to every row to fix this issue.

4.

PackageID	TagNumber	InstallDate	SoftwareCost USD	SoftwarePackageName	ComputerModel
AC01	32808	09-13-2005	754.95	Portal	IBM
DB32	32808	12-03-2005	380.00	Half Life	IBM
DB32	37691	06-15-2005	380.00	Half Life	Apple
DB33	57772	05-27-2005	412.77	Counter Strike	Lenovo
WP08	32808	01-12-2006	185.00	Garry's Mod	IBM
WP08	37691	06-15-2005	227.50	Garry's Mod	Apple
WP08	57222	05-27-2005	170.24	Garry's Mod	Lenovo
WP09	57740	10-30-2005	35.00	Team Fortress	Acer
WP09	77740	05-27-2005	35.00	Team Fortress	Dell

5. The SoftwarePackageName relies on the PackageID to determine the value, same with the ComputerModel value. ComputerModel relies on TagNumber to determine the name of the value. The SoftwareCostUSD is also reliant on the ComputerModel which is reliant on the PackageID. The functional dependency would look like:
   PackageID → SoftwarePackageName → SoftwareCostUSD
   TagNumber → ComputerModel
6. This table is not in the third normal form because it has transitive dependency. The SoftwareCostUSD is dependent on the SoftwarePackageName which is dependent on the PackageID. A non-prime attribute is an attribute that is not a part of a candidate key, the third normal rule table should have no translation dependency for non-prime attributes which is stopped in the second normal form. The way to fix this would be to split it up amongst separate tables.

Part Three:

InstallInfoTable

PackageID	TagNumber	InstallDate	SoftwareCostUSD
AC01	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	57740	10-30-2005	35.00
WP09	77740	05-27-2005	35.00

PackageNameTable

PackageID	SoftwarePackageName
AC01	Portal
DB32	Half Life
DB33	Counter Strike
WP08	Garry’s Mod
WP09	Team Fortress

ComputerNameTable

TagNumber	ComputerModel
32808	IBM
37691	Apple
57772	Lenovo

57222	Lenovo
57740	Acer
77740	Dell

7. The PackageID is the primary key of the InstallInfoTable, the PackageNameTable primary key is the PackageID. The ComputerNameTable primary key is the TagNumber.

8. The function dependencies for the InstallInfoTable would be PackageID, TagNumber → InstallDate, SoftwareCostUSD. The PackageNameTable dependencies for PackageID are the SoftwarePackageName. The only dependency for ComputerNameTable would be the ComputerModel column.

9. These tables are in the third normal form rule because there are no transitive dependencies, non prime attributes are not depending on the other non prime attributes. All the non prime attributes depend on the candidate keys only.
10.