Nick Carrozza

Lab 7 – Normalization One

3 November 2016

1. I would respond by saying that the data is evidently very unstructured. For readability, efficiency and effectiveness, I would advise a restructuring of the data to better reflect the rules of normalization. The way the data is structured now is certainly an issue within the scope of relational databases, especially being that the data is not yet in 1NF. Granted, the information is presented in a way that allows one to transform it into a more organized form.

2.

PackageID	TagNumber	InstallDate	SoftwareCostUSD
AC01	32808	9/13/05	754.95
DB32	32808	12/3/05	380
DB32	37691	6/15/05	380
DB33	57772	5/27/05	412.77
WP08	32808	1/12/06	185
WP08	37691	6/15/05	227.5
WP08	57222	5/27/05	170.24
WP09	59836	10/30/05	35
WP09	77740	5/27/05	35

3. The primary key of the above table in 1NF is the composite key (PackageID, TagNumber).

4.

PackageID	TagNumber	InstallDate	SoftwareCostUSD	SoftPackageName	CompModel
AC01	32808	9/13/05	754.95	Zork	IBM
DB32	32808	12/3/05	380	Portal	IBM
DB32	37691	6/15/05	380	Portal	Apple
DB33	57772	5/27/05	412.77	Mozilla	Compaq
WP08	32808	1/12/06	185	Chrome	IBM
WP08	37691	6/15/05	227.5	Chrome	Apple
WP08	57222	5/27/05	170.24	Chrome	Gateway
WP09	59836	10/30/05	35	APT	Asus
WP09	77740	5/27/05	35	APT	Sony

5. TagNumber → CompModel

   PackageID → SoftPackageName

   (PackageID, TagNumber) → InstallDate

   (PackageID, TagNumber) → SoftwareCostUSD

6. The table is not in third normal form because functional dependencies exist between non-key elements and elements exist that do not depend strictly on their respective primary key and only their respective primary key. For instance, SoftPackageName depends on PackageID not TagNumber and CompModel depends on TagNumber not PackageID, even though the composite key (PackageID, TagNumber) is the primary key of the table. This should be disaggregated or broken down such that there are no transitive dependencies between non-key elements in any of the tables, meaning, that every non-key attribute depends entirely on the primary key of the table and nothing but the primary key.

7. Systems:

PackageID	TagNumber	InstallDate	SoftwareCostUSD
AC01	32808	9/13/05	754.95
DB32	32808	12/3/05	380
DB32	37691	6/15/05	380
DB33	57772	5/27/05	412.77
WP08	32808	1/12/06	185
WP08	37691	6/15/05	227.5
WP08	57222	5/27/05	170.24
WP09	59836	10/30/05	35
WP09	77740	5/27/05	35

Tags:

TagNumber	CompModel
32808	IBM
37691	Apple
57772	Compaq
57222	Gateway
59836	Asus
77740	Sony

Packages:

PackageID	SoftPackageName
AC01	Zork
DB32	Portal
DB33	Mozilla
WP08	Chrome
WP09	APT

Primary Key for the Systems Table: (PackageID, TagNumber)

Primary Key for the Tags Table: TagNumber

Primary Key for the Packages Table: PackageID

8.

TagNumber → CompModel

PackageID → SoftPackageName

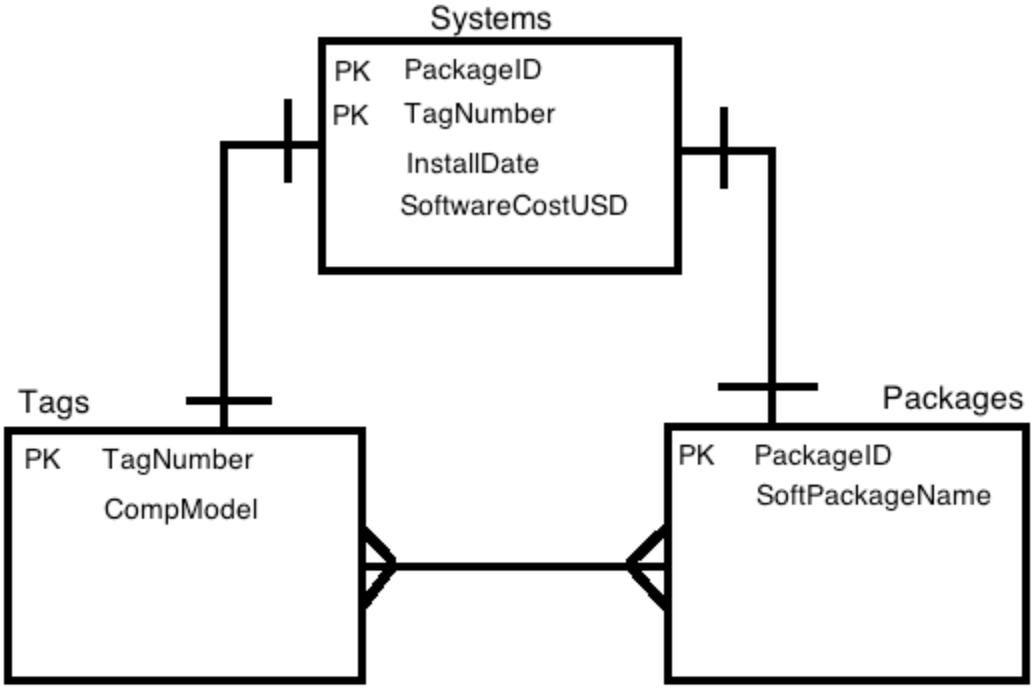
(PackageID, TagNumber) → InstallDate, SoftwareCostUSD

9. The new tables exhibit third normal form for the following reasons. First, the tables are in second normal form since no non-key attributes depend on only none or one of the two components that make up the primary key, but rather depend on both of them in unison. InstallDate depends on both the PackageID and TagNumber together, not one of them in particular. The same is true for SoftwareCostUSD. Further, the structure of the tables exhibits third normal form since each non-key attribute in each table depends on nothing but the primary key in their respective table. This is true since InstallDate and SoftwareCostUSD do not depend on each other whatsoever, and depend on the composite key (PackageID, TagNumber) in full. Also, CompModel depends strictly on TagNumber, and SoftPackageName depends

strictly on PackageID. Because of the functional dependencies between these attributes, the tables are in third normal form.

10)

E – R Diagram:


Systems
PK PackageID
PK TagNumber
InstallDate
SoftwareCostUSD
Tags
PK TagNumber
CompModel
Packages
PK PackageID
SoftPackageName