Dan Blossom

Professor Labouseur

CMPT 308 Database

October 28th, 2013

Normalization Homework 1

Part one:

1) I would thank him for the spreadsheet and say how useful it will be to assist in designing a robust database. I would further thank him for the opportunity.

2) The data entered into postgres in one large table below:

	packid character(4)	tagnum integer	installdate date	price numeric
1	AC01	32808	2005-09-13	754.95
2	DB32	32808	2005-12-03	380.00
3	DB32	37691	2005-06-15	380.00
4	DB33	57772	2005-05-27	412.77
5	WP08	32808	2006-01-12	185.00
6	WP08	37691	2005-06-15	227.50
7	WP08	57222	2005-05-27	170.24
8	WP09	59836	2005-10-30	35.00
9	WP09	77740	2005-05-27	35.00

3) The primary key in this scenario is a composite of packid and tagnum. I chose those two because there are repeats of packid and repeats of tagnum but not a repeat of packid and tagnum together.

Part two:

1) Two new columns added for computer model and software package name.

	packid character(4)	tagnum integer	installdate date	price numeric	model text	softpackname text
1	AC01	32808	2005-09-13	754.95	Apple	Poorware
2	DB32	32808	2005-12-03	380.00	Apple	Flux
3	DB32	37691	2005-06-15	380.00	HP	Flux
4	DB33	57772	2005-05-27	412.77	Dell	Fusion
5	WP08	32808	2006-01-12	185.00	Apple	OilBall
6	WP08	37691	2005-06-15	227.50	HP	OilBall
7	WP08	57222	2005-05-27	170.24	ASUS	OilBall
8	WP09	59836	2005-10-30	35.00	Lenovo	Darren
9	WP09	77740	2005-05-27	35.00	Acer	Darren

SQL code to create the above table:

```
-- create big messy table
drop table if exists kramerica_all
create table kramerica_all(
  packID       char(4) not null,
  tagNum       int not null,
  installDate  date not null,
  price        numeric,
  model        text,
  softpackname text,
  primary key(packID, tagNum)
);
```

2) The functional dependencies:

- packID → software package name
- tagNum → model
- packID, tagNum → installDate, price

3) There are a few issues that cause this database to not be in 3rd normal form.

- Delete anomaly, We cannot delete AC01 (aka Poorwware) without deleting its existence
- Update anomaly, Package name is entered numerous times, which could (and will) cause entry errors. There are no errors above because I entered the data VERY carefully (it was actually annoying!).
- Update anomaly, The same issue is with computer name as it is entered numerous times.
- Update anomaly, If Kramerica realizes a large rubber ball will not solve the tanker issue and need to rename that software package, it needs to be renamed in numerous places.
- Insert anomaly, if we want to enter a computer that has not received any packages yet we have to enter some arbitrary data to avoid null errors.

Part three:

```
--so it looks like the original spreadsheet
create view original as
        select packID, tagID, installDate, price
        from   inventories;
```

	packid character(4)	tagid integer	installdate date	price numeric
1	AC01	32808	2005-09-13	754.95
2	DB32	32808	2005-12-03	380.00
3	DB32	37691	2005-06-15	380.00
4	DB33	57772	2005-05-27	412.77
5	WP08	32808	2006-01-12	185.00
6	WP08	37691	2005-06-15	227.50
7	WP08	57222	2005-05-27	170.24
8	WP09	59836	2005-10-30	35.00
9	WP09	77740	2005-05-27	35.00

```
--a view to look like step two
create view detail_info as
        select i.packId, i.tagID, i.installDate, i.price, p.packName, c.model
        from inventories i, packages p, computers c
        where i.packID = p.packID
        and   i.tagID = c.tagID;
```

	packid character(4)	tagid integer	installdate date	price numeric	packname text	model text
1	AC01	32808	2005-09-13	754.95	Poorware	Apple
2	DB32	32808	2005-12-03	380.00	Flux	Apple
3	DB32	37691	2005-06-15	380.00	Flux	HP
4	DB33	57772	2005-05-27	412.77	Fusion	Dell
5	WP08	32808	2006-01-12	185.00	OilBall	Apple
6	WP08	37691	2005-06-15	227.50	OilBall	HP
7	WP08	57222	2005-05-27	170.24	OilBall	ASUS
8	WP09	59836	2005-10-30	35.00	Darren	Lenovo
9	WP09	77740	2005-05-27	35.00	Darren	Acer

1) I have created three tables for this database and the primary keys are as follows:

- Packages: Primary Key = packID
- Computers: Primary Key = tagID
- Inventories: Primary Key = invID

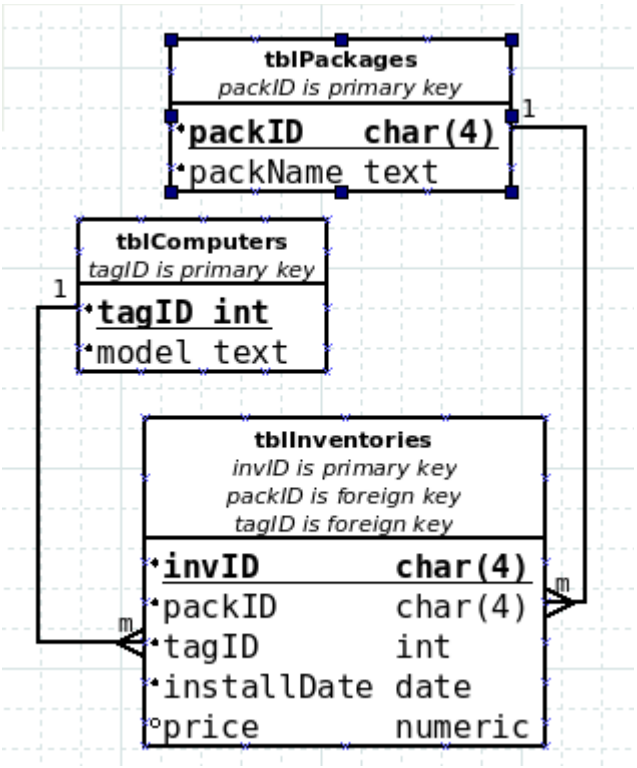
2) The functional dependencies are similar as to step two of part two but spread among the tables:

- Packages: packID → packName
- Computers: tagID → model
- Inventories: invID → packID, tagID, installDate, price

3) The new tables are in 3rd normal form because the anomalies have all been addressed. Also each key is dependent on only the key and nothing but the key.

- We can add a computer to the computers table and assign it a new asset tag without it needing software.
- We can build new packages of software without it attaching to a particular computer.
- We can add new fields to either the packages or computers tables without causing issues with the inventories table. Example if we needed to add serial to computers table, it should not break any of the other views or queries on inventories, unless serial was needed in an existing view.

4) E/R Diagram using Dia software for Linux. Since only primary key is noted by bold and underline I placed comments to explicitly state Primary and Foreign keys for each table.



SQL Code to create the tables:

```
--computer table
drop table if exists computers
create table computers(
  tagID     int not null,
  model     text not null,
  primary key(tagID)
);

--packages table
drop table if exists packages
create table packages(
  packID    char(4) not null,
  packName  text not null,
  primary key(packID)
);

--inventory table
drop table if exists inventories
create table Inventories(
  invID       char(4) not null,
  packID      char(4) not null references packages(packID),
  tagID       int not null references computers(tagID),
  installDate date not null,
  price       numeric,
  primary key(invID)
);
```