Örebro University

Database Normalization

Assignment 4 - Report

Duy Dinh & Özgun M
2014 - 12 - 18

Table of Contents

Introduction

In this assignment the students were given the task of analysing a demo database of a company. In the database there were different tables for each area of the company, including employees, departments, individual items, the stores, the suppliers etc.

The task was to make this demo database correct and efficient by implementing normal forms for each table and also checking for wrong information in the tables.

In this particular report the tables of the database will have their own sub-chapters where questions will be answered according to instructions.

For this assignment, the program DBVisualizer was utilized for easier displaying and manipulation of databases.

The format of the presentation for each sub-chapter is as follows:

- Possible Candidate keys.
- The chosen primary key.
- Full functional dependencies.
- Excel-table displaying small parts of each table for visualization.

Normal form rules

Conditions for normal-forms were followed by these points.

For **1NF**:

1. There are no repeating groups
2. All data values are atomic
3. Each field has a unique name
4. The table has a primary key

For **2NF**:

1. It is in 1NF
2. All non-key attributes are dependent on ALL PARTS of the primary key

For **3NF**:

1. It is in 2NF
2. All non-key attributes are not dependent on any other non-key attributes

TABLES

DEPT (Department)

Candidate keys: {Number}

Primary Key: Number

F.F.D (Full function dependency):

- {Number -> (Name, Manager, Store, Floor)}

Highest normal form: BCNF

There are no dependencies between non-key attributes since the column Number is the only field that determines the other attributes, which meets the requirement of BCNF.

NUMBER	NAME	MANAGER	STORE	FLOOR
1	Bargain	7	5	0
10	Candy	13	5	1
14	Jewelry	33	8	1
19	Furniture	26	7	4
20	Major Appliances	26	7	4

EMPLOYEE

Candidate keys: {Number}

Primary key: Number

F.F.D:

- {(Number) -> (Name, Salary, Birthdate, Startdate)}

Highest normal form: BCNF

The only thing that needs to be attended to in this table is the column “Name” where it needs to be split into two different columns “Surname” and “Forename” to make it atomic. After correcting this issue, this table meets the requirement of BCNF.

NUMBER	SURNAME	FORENAME	SALARY	MANAGER	BIRTHDATE	STARTDATE
10	Ross	Stanley	15908	199	1927	1945
11	Ross	Stuart	12067	0	1931	1932
13	Edwards	Peter	9000	13	1928	1958
26	Thompson	Bob	13000	199	1930	1970
32	Smythe	Carol	9050	199	1929	1967

ITEM

Candidate keys: {Number}

Primary key: Number

F.F.D:

- {(Number) -> (Name, Dept, Price,QoH, Supplier)}

Highest normal form: BCNF

The table below meets the requirements for BCNF since there are no other candidate keys to depend on. All the non-key attributes are fully dependent on the primary key, which displays the item number.

Items					
NUMBER	**NAME**	**DEPT**	**PRICE**	**QOH**	**SUPPLIER**
11	Wash Cloth	1	75	575	213
19	Bellbottoms	43	450	600	33
21	ABC Blocks	1	198	405	125
23	1 lb Box	10	215	100	42

PARTS

Candidate keys: {PNUM}

Primary key: PNUM

F.F.D:

- {PNUM -> (PNAME, COLOR, WEIGHT, QOH)}

Highest normal form: BCNF

One problem with this table is that there are no ways to split it without assuming that each column can only have one set of values. This makes it hard to split name and colour into its own table since it implies that a particular part can strictly have one colour. In real world, the same part might have different colours.

PNUM	**PNAME**	**COLOR**	**WEIGHT**	**QOH**
1	central processor	pink	10	1
2	memory	gray	20	32
3	disk drive	black	685	2
4	tape drive	black	450	4

SALE

Candidate keys: {Item, Number-Item}

Primary key: Number-Item

F.F.D:

- {(Number, Item) -> Quantity}
- {Number -> (Date, Employee, Credit)}
- {Item -> Dept}
- {Dept -> Store}

Highest normal form: 1NF

This doesn't meet the requirements for higher normal forms at first, mostly because of the several dependencies between the non-key attributes. It didn't meet the 2NF since there were F.F.D's to some of the other candidate keys – this violates the condition for 2NF.

This table was one of the harder ones to normalize because of the large amount of determinants. The table was split into four different tables to fulfil the requirements of 3NF and as well as BCNF:

Items Sold

NUMBER	ITEM	QUANTITY
100581	118	5
100581	120	1
100582	26	1

Receipt

NUMBER	DATE	EMPLOYEE	CREDIT
100581	15/01/1975	157	0
100582	15/01/1975	1110	24356540
100586	16/01/1975	35	54096831

Item Area

ITEM	DEPT
118	26
120	26
26	14

Dept Area

DEPT	STORE
26	7
14	8
49	8

STORE

Candidate keys: {Number, City}

Primary key: Number

F.F.D:

- {Number -> (City, State)}
- {(City) -> (State)}

Highest normal form: 2NF

The reason for this is that "City" and "State" is dependent on each other. This violates the requirement for 3NF.

The field below demonstrates a complete table of BCNF:

Store

NUMBER	CITY
5	San Francisco
7	Oakland
8	El Cerrito

City area

CITY	STATE
San Francisco	Calif
Oakland	Calif
El Cerrito	Calif

SUPPLIER

Candidate keys: {Number, City}

Primary key: Number

F.F.D:

- {City -> State}

There is also a functional key in {(Name, City) -> State}

Highest normal form: 2NF

It doesn't reach 3NF because there is a transitive way from "City" to "State". Below are the tables in BCNF:

Supplier

NUMBER	NAME	CITY
15	White Stag	White Plains
33	Levi-Strauss	San Francisco
42	Whitman's	Denver
89	Fisher-Price	Boston

Supplier loc.

CITY	STATE
White Plains	Neb
San Francisco	Calif
Denver	Colo
Boston	Mass

SUPPLY

Candidate keys: {PNUM-JNUM}

Primary key: PNUM-JNUM

F.F.D: There were no apparent dependencies in this table, mostly because the columns had names which were difficult to interpret and understand.

Highest normal form: BCNF

From looking at the fields, it was noticed that "PNUM" and "JNUM" was determining the other columns. For BCNF to be valid, each determinant needs to be a candidate key. This is true in this case.

PNUM	JNUM	SNUM	SHIPDATE	QUAN
7	1003	122	01/02/1975	144
8	1005	241	01/07/1975	1
9	1004	122	01/02/1975	144
10	1006	999	01/01/1976	144

Ending words

There are reservations for errors, including human errors. There might be things that have been overlooked and some of the data in the tables can be incorrect in correlation to the other tables, if corrections were not implemented correctly. Overall the database tables are normalized as much as possible.