Case Study Week 6

Reuben E. Perry

INFO321 Database Management Systems

Andrew Fitzgerald

18 March 2018

Case Study Week 6: Database Redesign

There are many changes that I would make to the spreadsheet that is presented to me. The first thing that stands out to me is that there is no organization to the spreadsheet. I would either start by organizing the spreadsheet in order of invoice number or in order by company name. Next, I would assign a unique company ID to each individual organization on the spreadsheet. A unique company ID will alleviate confusion if there were to be different organizations with similar names. The unique company ID will also allow for a simple query and establish relationships with various tables. Additionally, the company ID will establish a primary key in future tables through the normalization process. I would also assign a unique part ID to each part. I would assign a unique part ID for the same reasons I would assign organizations unique company IDs. In addition to the added fields, I would split the spreadsheet into separate tables for reducing data redundancy and for ease of keeping track of various factors on the spreadsheet. Splitting the spreadsheet into separate tables is essential for the normalization process to reduce possibilities of insertion anomalies, deletion anomalies, and modification anomalies.

As mentioned, I would split the spreadsheet into multiple tables. The first table will resemble the invoice portion of the original spreadsheet. There won’t be part 1 or part 2 for the sake of the process of normalization. NF1, NF2, and NF3 work to reduce data redundancy of which removing part 2 will be essential for normalization. Multiple parts may be entered but the additional part will reflect on the next row. The design of the invoice table will be as followed:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| InvoiceID | InvoiceNumber50 | CompanyID | PartID | Qty |
|  |  |  |  |  |

The next table will reflect each company. This table will reflect company ID, company name, account balance, and account limit.

|  |  |  |  |
| --- | --- | --- | --- |
| CompanyID | CompanyName | AccountBalance | AccountLimit |
|  |  |  |  |

The final table will resemble parts. This table will simply reflect the part ID, part name, and unit price. I changed the original name of price1 or price2 to UnitPrice as I feel that UnitPrice is specific to the definition of the field.

|  |  |  |
| --- | --- | --- |
| PartID | PartName | UnitPrice |
|  |  |  |

First Normal Form (1NF) will require the current UNF to not have any multiple values and only contain atomic values. Therefore, the original UNF in 1NF will look like the following. Please note that I a column titled “InvoiceID”. This will allow me to assign a unique number to each action of the invoice and assign a primary key.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| InvoiceID  (Primary Key) | InvoiceNumber | CompanyName | AcctBalance | AcctLimit | Part | Qty | Price |
| 871 | 87 | Mary’s IT Services | 5000 | 20000 | Bolt | 50 | 10 |
| 872 | 87 | Mary’s IT Services | 5000 | 20000 | Nut | 40 | 20 |
| 721 | 72 | Mel’s Hotdogs | 7200 | 7200 | Bolt | 100 | 10 |
| 722 | 72 | Mel’s Hotdogs | 7200 | 7200 | Screw | 45 | 20 |
| 571 | 57 | Brookings | 500 | 12000 | Screw | 60 | 30 |
| 572 | 57 | Brookings | 500 | 12000 | Bolt | 65 | 10 |
| 671 | 67 | Parker’s | 1000 | 5000 | Nut | 70 | 20 |
| 31 | 3 | Action Computer’s | 1000 | 10000 | Screw | 40 | 30 |
| 71 | 7 | Kline’s | 8000 | 9999 | Bolt | 30 | 10 |
| 91 | 9 | John’s | 10000 | 30000 | Nail | 25 | 50 |
| 171 | 17 | Lee’s | 700 | 1000 | Bolt | 75 | 10 |
| 172 | 17 | Lee’s | 700 | 1000 | Nail | 120 | 50 |
| 191 | 19 | Al’s | 7000 | 7000 | Screw | 110 | 30 |
| 201 | 20 | Al’s | 7000 | 7000 | Nut | 90 | 20 |

The functional dependency analysis is as followed:

InvoiceID Company Name, Part, Qty

CompanyName Acct Balance, Acct Limit

Part Price

In order for the following table to be in 2NF, it must be in 1NF and not contain any non-primary attributes. Therefore, the following table will represent the UNF in 2NF. Please note that an ID as been assigned to the parts and the companies to establish a primary key.

|  |  |  |  |
| --- | --- | --- | --- |
| CompanyID (primary key) | CompanyName | AccountBalance | AccountLimit |
| 101 | Mary’s IT Services | 5000 | 20000 |
| 102 | Mel’s Hotdogs | 7200 | 7200 |
| 103 | Brookings | 500 | 12000 |
| 104 | Parker’s | 1000 | 5000 |
| 105 | Action Computer’s | 1000 | 10000 |
| 106 | Kline’s | 8000 | 9999 |
| 107 | John’s | 10000 | 30000 |
| 108 | Lee’s | 700 | 1000 |
| 109 | Al’s | 7000 | 7000 |

The functional dependency is as followed:

CompanyID CompanyName, AccountBalance, AccountLimit

The next table will contain details in regards to the part. PartID is an added field for the sake of implementing a primary key.

|  |  |  |
| --- | --- | --- |
| PartID  (Primary Key) | PartName | UnitPrice |
| 1001 | Bolt | 10 |
| 1002 | Screw | 30 |
| 1003 | Nut | 20 |
| 1004 | Nail | 50 |

The functional dependency is as followed:

PartID PartName, UnitPrice

The third table will include the details in regards to the invoice itself.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| InvoiceID  (Primary Key) | InvoiceNumber | CompanyID | CompanyName | PartID | Qty |
| 871 | 87 | 101 | Mary’s IT Services | 1001 | 50 |
| 872 | 87 | 101 | Mary’s IT Services | 1003 | 40 |
| 721 | 72 | 102 | Mel’s Hotdogs | 1001 | 100 |
| 722 | 72 | 102 | Mel’s Hotdogs | 1002 | 45 |
| 571 | 57 | 103 | Brookings | 1002 | 60 |
| 572 | 57 | 103 | Brookings | 1001 | 65 |
| 671 | 67 | 104 | Parker’s | 1003 | 70 |
| 32 | 3 | 105 | Action Computer’s | 1002 | 40 |
| 71 | 7 | 106 | Kline’s | 1001 | 30 |
| 91 | 9 | 107 | John’s | 1004 | 25 |
| 171 | 17 | 108 | Lee’s | 1001 | 75 |
| 172 | 17 | 108 | Lee’s | 1004 | 120 |
| 191 | 19 | 109 | Al’s | 1002 | 110 |
| 201 | 20 | 109 | Al’s | 1003 | 90 |

The functional dependency is as followed:

InvoiceID CompanyID, CompanyName, PartID, Qty

CompanyID CompanyName

In order to turn the following data into 3NF, we will have to remove any transitive dependencies that exist. Both the company and part data tables do not contain any transitive dependencies. However, the invoice table contains a transitive dependency of which is CompanyID CompanyName. We will need to remove the CompanyName column for the table to truly be 3NF.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| InvoiceID  (Primary Key) | InvoiceNumber | CompanyID | PartID | Qty |
| 871 | 87 | 101 | 1001 | 50 |
| 872 | 87 | 101 | 1003 | 40 |
| 721 | 72 | 102 | 1001 | 100 |
| 722 | 72 | 102 | 1002 | 45 |
| 571 | 57 | 103 | 1002 | 60 |
| 572 | 57 | 103 | 1001 | 65 |
| 671 | 67 | 104 | 1003 | 70 |
| 31 | 3 | 105 | 1002 | 40 |
| 71 | 7 | 106 | 1001 | 30 |
| 91 | 9 | 107 | 1004 | 25 |
| 171 | 17 | 108 | 1001 | 75 |
| 172 | 17 | 108 | 1004 | 120 |
| 191 | 19 | 109 | 1002 | 110 |
| 201 | 20 | 109 | 1003 | 90 |

The functional dependency is as followed:

InvoiceID InvoiceNumber, CompanyID, PartID, Qty

The normalization process shown above is essential for the reduction of data redundancy. It’s critical to reduce data redundancy to reduce the possibilities of anomalies previously mentioned. As seen, each step in the normalization process reduces data redundancy and allow for easier maintenance on a database and the reduction of inconsistencies. A database that has followed the normalization process will allow Home Solutions Inc to keep better track of their inventory system, customers and sales.

Reference

Connolly, T. M., & Begg, C. E. (2015). Database systems: a practical approach to design,

implementation and management (6th ed.). Boston: Pearson.

Singh, C., M., Luse, R., G., S., D., . . . Y. (2015, May 02). Normalization in DBMS: 1NF, 2NF,

3NF and BCNF in Database. Retrieved March 18, 2018, from https://beginnersbook.com/2015/05/normalization-in-dbms/