**Database Design Process and Issues report**

For Wholey Moley Foods

**Stella Fu**

04 September 2020

**Table of Contents**

[1. Introduction 1](#_Toc50130820)

[2. Design Process and Issues 2](#_Toc50130821)

[2.1 Entities 2](#_Toc50130822)

[2.2 Keys 2](#_Toc50130823)

[2.3 Relationships 3](#_Toc50130824)

[2.4 Bridging Entities 4](#_Toc50130825)

[2.5 Attributes 5](#_Toc50130826)

[2.5.1 Customer Table 6](#_Toc50130827)

[2.5.2 Employee Table 6](#_Toc50130828)

[2.5.3 Branch Table 9](#_Toc50130829)

[2.5.4 Item Table 9](#_Toc50130830)

[2.5.5 Order Table 10](#_Toc50130831)

[2.6 Extent of Normalisation 11](#_Toc50130832)

[2.6.1 First Normal Form 11](#_Toc50130833)

[2.6.2 Second Normal Form 11](#_Toc50130834)

[2.6.3 Third Normal Form 12](#_Toc50130835)

[2.6.4 ERD 12](#_Toc50130836)

[2.7 The Extended ERD 13](#_Toc50130837)

[2.8 Data Testing in SQL Server Management Studio 14](#_Toc50130838)

[3. Conclusion and Recommendation 15](#_Toc50130839)

# 1. Introduction

This report is developed to record the process during designing a database management system for Wholey Moley Foods. The business focuses on health food. The products are sold via three branches, farmers markets and trade shows. Alister Lance is the owner of the business. He would like to use the designed system to manage the sales of finished items, customers and employee.

Microsoft Visio is utilised to design the Entity Relationship Diagram (ERD). SQL Server Management Studio is the tool for ERD implementation and data testing.

The following sections of the report will introduce the design process and issues regarding entities, keys, bridging entities, connectivity, attributes and normalisation.

Research findings of the selected jobs are explained in the following section. To summarise the research, the author selected one of the jobs as her preferred future role and discussed whether the skills required for the job have been acquired and how to gain those skills that have not been acquired. Further, the author gave a personal reflection on this research. Lastly, screenshots of the job advertisements were captured and attached in the section of Appendix.

# 2. Design Process and Issues

This section shows how the designer created a relational schema from the aspects of entities, primary and foreign keys, composite entities, relationships, attributes and extent of normalisation.

## 2.1 Entities

According to the provided business rules, six basic entities were generated at the first stage. They are **Customer**, **Employee**, **Branch**, **Item**, **Account and Order**.

A close up of text on a white background

Description automatically generated

Figure - Six Basic Entities at the First Stage

## 2.2 Keys

At this point of time, each of the six entities has a primary key that is unique. They are **cutomerID**, **orderID**, **employID**, **branchID**, **itemID**, **accountID**.

A close up of text on a white background

Description automatically generated

Figure - Basic Entities with Primary Key

## 2.3 Relationships

Relationships between the six basic entities are showed with **cardinality notation (Crow’s Foot Notation)** in the diagram below. This type of cardinality notation was applied throughout the whole ERD design.

A close up of a map

Description automatically generated

Figure - Six Basic Entities

* A customer may have zero, one or several accounts but an account always matches only one customer
* A customer is served by one employee and an employee can have zero or many customers
* A customer may place many orders, but an order belongs to one and only one customer
* An order has one and only one employee who receive the order, and an employee may have many orders
* An employee is based in one branch and he/she may work at multiple branches, and a branch can have many employees
* A branch can have many orders, but an order belongs to one and only branch
* A branch may have many customers and a customer may shop at many branches
* A branch has one or many items and an item may be unavailable in some branches or available in many branches

Cardinality notation used in the above diagram is explained in the following table for reference.

|  |  |
| --- | --- |
| A close up of a building  Description automatically generated | One and Only One (1:1) |
| A close up of a building  Description automatically generated | One or Many (1:M) |
| A close up of a logo  Description automatically generated | Zero or Many (0:M) |

Figure - Cardinality Notation

## 2.4 Bridging Entities

Three many to many (M:N) relationships were encountered in the *Figure 3* diagram.

1. A customer may visit many branches over years and a branch can have many customers
2. A branch may sell many items or certain items, and an item is available in many branches or a certain branch
3. An employee is based in a branch, but he/she may work at multiple branches, and a branch can have many employees

However, the relational model cannot model many to many (M:N) relationships. To solve the issues, bridges must be created between the entities that display M:N relationship.

A close up of text on a white background

Description automatically generated

**Bridging Entity 3**

**Bridging Entity 2**

**Bridging Entity 1**

Figure - Bridging Entities Being Added

In each of the bridging entities in *Figure 5*, the unique primary key consists of two primary keys from each of the entities to be connected. In other word, the bridging entities used foreign keys as their primary key. At this point of time, we have nine entities and there are none many to many relationships between those entities ( see *Figure 5* ). However, in the following section, more bridging entities or composite entities will be created.

## 2.5 Attributes

Based on the nine established entities, attributes were added into every entity according to the early provided business rules and the later add-on requirements from the Forum on Moodle.

Specifically, a single-valued attribute can have only a single value. A simple attribute cannot be subdivided. A composite attribute can be further subdivided into additional attributes. Therefore, some new attributes were created to subdivide the composite attributes.

A close up of text on a white background

Description automatically generated

Figure - Nine Entities with Attributes

### 2.5.1 Customer Table

|  |  |  |
| --- | --- | --- |
| **Simple attribute(i.e.)** | **Composite attribute(i.e.)** | **Single-valued attribute(i.e.)** |
| Date of birth(DOB) | Name | Customer ID |
| Customer type | Address |  |
| Credit status |  |  |
| Email |  |  |

Figure - Attributes in Customer Table

Composite attributes in the above table can be subdivided to attributes:

* Name – First Name, Last Name
* Address – Street, City, Postcode

### 2.5.2 Employee Table

|  |  |  |  |
| --- | --- | --- | --- |
| **Simple attribute(i.e.)** | **Composite attribute(i.e.)** | **Single-valued attribute(i.e.)** | **Multivalued attribute(i.e.)** |
| Date of birth(DOB) | Name | Employee ID | \*Telephone numbers |
| Date of hire | Address |  | \*Skill sets |
| Job title |  |  | \*Working hours |
| Current pay rate |  |  |  |
| \*ReportTo |  |  |  |
| \*OnBoard |  |  |  |

Figure - Attributes in Employee Table

Employees’ age can be derived from the data of their date of birth and the current date, so ‘Age’ is a derived attribute and it will not be stored in the database.

Composite attributes in the above table can be subdivided to attributes:

* Name – First Name, Last Name
* Address – Street, City, Postcode

According to the client’s requirement of keeping records of:

* Employees that have previously been used (on board status)
* History of employees’ working hours, with starting and finishing time for the purpose of staff lunch break and contact tracing (working hour)
* Skill sets of employees (skill set)
* Both employee phone number and emergency contact phone number (telephone numbers)

Specifically, in the Employee table, stating ‘Yes’ or ‘No’ in the onBoard column instead of deleting the employee from the table if they are leaving would be an easy to keep record of employees that have been used.

A new table named WorkingHour need to be created, because one employee can have many working hours in a branch or branches. These are repeatable information that do not need to record under Employee table.

A screenshot of a cell phone

Description automatically generated

Figure - WorkingHour Table

However, according to the client’s reply to employee working hour on Moodle, the designer decided to record branch name instead of branch ID in this table, because their staff normally talk about the branch name rather than a branch ID. If the owner would like to have a report to see who worked at a certain branch on a certain date, we can make a SQL query by using the ‘%’ before and after a required branch name.

Moreover, the client concerns about the repeatable information of skill sets. Therefore, a new table named Skillset is created to fulfil this requirement. However, the relationship between Employee table and Skillset table is many to many (M:N), so a bridge entity called EmployeeSkill was created to solve this issue. The primary key composes of two foreign keys. They are the primary keys from the connected tables.

A screenshot of a cell phone

Description automatically generated

Figure - EmployeeSkill Entity

Further, an attribute named reportTo was added into the Employee table to enable the client to produce a report that displays employee’s name and their manager’s name in the future operation. This need can be achieved by making SQL query of recursion.

Lastly, according to the client’s further requirement of employee’s phone number, the multivalued attribute: telephone numbers was divided to two attributes as below.

* Employee phone number
* Emergency contact phone number

### 2.5.3 Branch Table

|  |  |  |  |
| --- | --- | --- | --- |
| **Simple attribute(i.e.)** | **Composite attribute(i.e.)** | **Single-valued attribute(i.e.)** | **Multivalued attribute(i.e.)** |
| Branch name | Address | Branch ID | \*Monthly total sales amount |
| Area |  |  |  |

Figure - Attributes in Branch Table

The composite attribute: Address can be subdivided to attributes:

* Street, City and Postcode

According to the information provided by the client, they do have branch ID that consist of letter and number. For example, WASH1 is the branch ID for Washdyke Farmer’s Market. Therefore, letter + number became the format of Branch ID, and the testing data will follow this naming rule for Branch ID.

Also, the business is operated in the following area at this point of time:

* Canterbury, Hanan Shields, Timaru, Oamaru

To avoid redundant information in the table, monthly total sales amount will not become an attribute. As it can be gained via making SQL query by using relevant attributes from other tables.

### 2.5.4 Item Table

|  |  |  |
| --- | --- | --- |
| **Simple attribute(i.e.)** | **Single-valued attribute(i.e.)** | **Multivalued attribute(i.e.)** |
| Item description | Item ID | Allergen |
| Size |  |  |
| \*SubTo |  |  |

Figure - Attributes in Item Table

An item may have one or more allergen, so another table called Allergen was created. However, the relationship between Item table and Allergen table is many to many (M:N), so a bridge entity called ItemAllergen was created to solve this problem. Two foreign keys from the connected tables became the unique primary key in ItemAllergen Table.

A close up of text on a white background

Description automatically generated

Figure - ItemAllergen Entity

Particularly, an item can be composed of several items. For example, item 2001 may include item 256 and item 259. In this case, item 256 and item 259 all sub to item 2001. Avoid creating another table to record repeatable information, an attribute name subTo was added into the Item Table.

### 2.5.5 Order Table

|  |  |
| --- | --- |
| **Simple attribute(i.e.)** | **Single-valued attribute(i.e.)** |
| Order date | Order ID |
| PickUp date | Branch ID |
| Credit authorisation status | Employee ID |

Figure - Attributes in Order Table

However, an order can have one or many items, and an item can be sold through many orders. This is a many to many (M:N) relationship, so a composite entity named OrderLine was created to solve this issue.

A close up of text on a white background

Description automatically generated

Figure - OrderLine Entity

## 2.6 Extent of Normalisation

This section shows one example of normalisation during this design work. Primary keys are those blocks with blue background.

### 2.6.1 First Normal Form

In 1 NF, repeatable groups should be removed, primary keys need to be defined and attributes need to be single valued. Therefore, ‘name’ was subdivided to first name and last name. ‘address’ was subdivided to street, city and postcode.

A screenshot of a cell phone

Description automatically generated

Figure - An Example of First Normal Form

Partial dependencies that reply on part of the primary key was marked in *Figure 16*.

Also, skill code determines skill description, but this is a transitive dependency, because the skill code attribute is a non-primary key.

### 2.6.2 Second Normal Form

A close up of a piece of paper

Description automatically generated

Figure - An Example of Second Normal Form

In 2 NF, the table should be in 1 NF and the partial dependencies need to be removed.

### 2.6.3 Third Normal Form

A circuit board

Description automatically generated

Figure - An Example of Third Normal Form

In 3 NF, the table is in 2 NF and includes no transitive dependency.

### 2.6.4 ERD

A close up of text on a white background

Description automatically generated

Figure - An Example of ERD

After applying third normal form, an ERD was obtained and displayed in *Figure 19*.

## 2.7 The Extended ERD

The following relational schema was created by Microsoft Visio.

A close up of text on a white background

Description automatically generated

Figure – The Extended ERD with Entities, Relationships, Cardinality, Attributes and Primary Keys

## 2.8 Data Testing in SQL Server Management Studio

A screenshot of a cell phone

Description automatically generated

Figure - Database Diagram in SQL Server

# 3. Conclusion and Recommendation

In summary, this report recorded the process and the issues encountered as well as how to solve those issues during designing the database management system for the business of Wholey Moley Foods.

Eventually, 14 tables were created with relevant attributes to achieve the client’s requirements. But at this stage, with the limit scope and time, the designed database only includes the business’s selling part. The manufacturing and purchasing part are out of scope of this design at the moment.

By using the Microsoft Visio and SQL Server Management Studio, the extended ERD and database diagram was created and attached in this report.