RDBMS Outcome 2

Connley Farquhar

Dundee & Angus College

HND Software Development Year 2

Table of Contents

[Planning 2](#_Toc183768682)

[Description of Case Study 2](#_Toc183768683)

[Data Dictionary 2](#_Toc183768684)

[Scenario Scope 4](#_Toc183768685)

[Normalisation Process 5](#_Toc183768686)

[ERD Diagram 6](#_Toc183768687)

[Business Rules 6](#_Toc183768688)

[Functional Rules 7](#_Toc183768689)

[Database Documentation 7](#_Toc183768690)

[Translation of Design Model to Physical Model 7](#_Toc183768691)

[Table Sample Data 12](#_Toc183768692)

[Customer Table 12](#_Toc183768693)

[Item Table 13](#_Toc183768694)

[Item Stock Table 13](#_Toc183768695)

[Menu Table 13](#_Toc183768696)

[Orders Table 13](#_Toc183768697)

[Outlet Table 14](#_Toc183768698)

[Shift Table 14](#_Toc183768699)

[Staff Table 14](#_Toc183768700)

## Planning

|  |
| --- |
| **Description** |
| Identity Questions to ask Client to structure Database Design for Implementation. |
| Define Entities that will be used in the Database for the client project. |
| Research on how Entity Relationships would be connected. |
| Ponder Normalisation and how it will be laid out. |
| Consider Unique Identifiers for Entity Groups. |
| Work in line with the Client to ensure all areas are covered before implementation. |
| Determine Scope of SQL Code that would have to be implemented for functionality. |

## Description of Case Study

This case study is based on creating, designing and implementing a relational database management system in short (RDBMS) for the company Fast Burgers. They are a fast-food franchise with well over 100 Outlets within the United Kingdom. This Database must also hold two menus a breakfast menu which will only be available at 11AM and a “savers” Menu which can differentiate depending on the season or month.

The system they are asking for us to design requests ability to:

1. Track Orders
2. Identify and list staff members handling orders
3. Sort and list the most popular orders
4. Make note of payment methods

## Data Dictionary

|  |  |  |
| --- | --- | --- |
| **Database:** Fast Burgers | | |
| **Entity name: Staff** | | |
| **Description: Staff Table will hold data on staff members working at one of many outlets at Fast Burger.** | | |
| **Attribute** | **Datatype** | **Constraints/comments** |
| staffID  staffSurname  staffRole | INT(50)  VARCHAR(50)  VARCHAR(50) | Primary Key |

|  |  |  |
| --- | --- | --- |
| **Database:** Fast Burgers | | |
| **Entity name: Shifts** | | |
| **Description: Shift Table will hold data on staff members shift dates.** | | |
| **Attribute** | **Datatype** | **Constraints/comments** |
| shiftID  shiftDate  shiftStart  shiftEnd  staffID | INT(50)  DATE  TIME  TIME  INT(50) | Primary Key  Foreign Key |
| **Database:** Fast Burgers | | |
| **Entity name: Outlet** | | |
| **Description: Outlet Table will hold data on the outletID and Location of the outlet.** | | |
| **Attribute** | **Datatype** | **Constraints/comments** |
| outletID  outletLocation | INT(50)  VARCHAR(50) | Primary Key |
| **Database:** Fast Burgers | | |
| **Entity name: Orders** | | |
| **Description: Orders Table will hold data on orders placed in the outlet.** | | |
| **Attribute** | **Datatype** | **Constraints/comments** |
| orderID  orderDate  custID  outletID | INT(50)  DATETIME  INT(50)  INT(50) | Primary Key  Foreign Key (Customer Table)  Foreign Key (Outlet Table) |
| **Database:** Fast Burgers | | |
| **Entity name: Customer** | | |
| **Description: Customer Table will hold data on customers that visit the outlet.** | | |
| **Attribute** | **Datatype** | **Constraints/comments** |
| custID  custName  custpayType | INT(50)  VARCHAR(50)  VARCHAR(50) | Primary Key |
| **Database:** Fast Burgers | | |
| **Entity name: Item** | | |
| **Description: Item Table will hold data on Items on each menu sold at the outlets.** | | |
| **Attribute** | **Datatype** | **Constraints/comments** |
| itemID  itemName  menuID | INT(50)  VARCHAR(50)  VARCHAR(50) | Primary Key  Foreign Key (Menu Table) |
| **Database:** Fast Burgers | | |
| **Entity name: ItemStock** | | |
| **Description: ItemStock Table will hold data stock amount of items in the outlet.** | | |
| **Attribute** | **Datatype** | **Constraints/comments** |
| itemID  outletID  itemstockAmount | INT(50)  INT(50)  INT(255) | Foreign Key (Item Table)  Foreign Key (Outlet Table) |
| **Database:** Fast Burgers | | |
| **Entity name: Menu** | | |
| **Description: Menu Table will hold data on menus and what each is for.** | | |
| **Attribute** | **Datatype** | **Constraints/comments** |
| menuID  menuType  menuStart  menuEnd | INT(50)  VARCHAR(50)  DATE  DATE | Primary Key |

## Scenario Scope

This Section covers the scope of the project, very simple/basic planning information and what will have to be worked on to achieve project success:

|  |
| --- |
| **Description** |
| Two Menus, Breakfast and Savers. |
| Order Data to be noted like time of order, outlet ordered in, and items ordered. |
| Each order will have a unique staffID Linked to track what staff issued order. |
| Keep track on customer payment type. |

# Normalisation Process

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **UNF** | **1NF** | **2NF** | **3NF** | **Entities** |
| staffID  staffSurname  staffRole  shiftID  shiftDate  shiftStart  shiftEnd  outletID  outletLocation  orderID  orderDate  custID  custName  custpayType  itemID  itemName  itemStock  itemstockAmount  menuID  menuType  menuStart  menuEnd | staffID (PK)  staffSurname  staffRole  shiftID (PK)  shiftDate  shiftStart  shiftEnd  staffID (FK)  outletID (PK)  outletLocation  orderID (PK)  orderDate  custID (FK)  outletID (FK)  custID (PK)  custName  custpayType  itemID (PK)  itemName  itemStock  itemstockAmount  menuID (FK)  menuID (PK)  menuType  menuStart  menuEnd | staffID (PK)  staffSurname  staffRole  shiftID (PK)  shiftDate  shiftStart  shiftEnd  staffID (FK)  outletID (PK)  outletLocation  orderID (PK)  orderDate  custID (FK)  outletID (FK)  custID (PK)  custName  custpayType  itemID (PK)  itemName  menuID (FK)  itemID (PK)  itemStock  itemstockAmount  outletID (FK)  menuID (PK)  menuType  menuStart  menuEnd | staffID (PK)  staffSurname  staffRole  shiftID (PK)  shiftDate  shiftStart  shiftEnd  staffID (FK)  outletID (PK)  outletLocation  orderID (PK)  orderDate  custID (FK)  outletID (FK)  custID (PK)  custName  custpayType  itemID (PK)  itemName  menuID (FK)  itemID (PK)  outletID (FK)  itemstockAmount  menuID (PK)  menuType  menuStart  menuEnd | **Staff**  **Shift**  **Outlet**  **Order**  **Customer**  **Menu Items**  **Stock**  **Menu** |

## ERD Diagram

A diagram of a customer

Description automatically generated

## Business Rules

Specific Business Rules required based on client Brief / Case Study are as follows:

Database MUST have two separate Menus:

1. Breakfast Menu, only available until 11AM (menuID, menuType)
2. Savers Menu, Differentiates depending on the month or season (menuID, menuType)

Database MUST be able to cover and take in data from the 100+ Outlets across the UK, this can be done by displaying which outlet the data is from by assigning it an outletID.

Orders Placed Must have Details attached to it such as:

1. Items Ordered (i.e What product was bought from a menu)
2. Time the order was placed (orderDate)
3. The Outlet it was placed from (outletID)

Order Rules that relate to the menu Items:

1. Must have the Item Name (itemID, itemName)
2. Must have the Item Price (itemID, itemPrice)
3. Items on menus must depend on availability times (menuID, menuType, menuStart, menuEnd)

Orders MUST also hold information from Customers relating to their payment method (outletID, custpayType)

Orders will also have to be marked with the staff Member who dealt with the transaction (staffID)

## Functional Rules

Specific Functional Rules required based on client Brief / Case Study are as follows:

1. Database MUST hold information on customer payment methods used on checkout (custID, custpayType)
2. Must Ensure that Orders placed are Linked with the specific Staff member who placed it (staffID, staffSurname)
3. Database System should be taking in and tracking order information on both menus to see what item is ordered the most. (itemID, itemName, menuID, menuType)

# Database Documentation

### Translation of Design Model to Physical Model

All Database table inputs and entities are based on my Third Normal Form Process.

A close up of text

Description automatically generated

Set staffID as Primary Key for the Staff table, varchar was used as it allows for integers and strings as an input.

A screenshot of a computer screen

Description automatically generated

Inserting data into the Staff Table, holds surnames of staff their id and their role within the store.

A computer code with text

Description automatically generated

Set shiftID as the Primary Key for the shift table, this table is used to store staff shift information, table uses staffID from the staff table as a foreign key, ”REFERENCES” is used in SQL when displaying a foreign key relationship between tables.

A screenshot of a computer code

Description automatically generated

Inserting data to the shift table, with predicted data based on staff and their roles.

A group of words on a white background

Description automatically generated

Set custID as a primary key for this table, table is used to store customer order information, after I add the order table, I will be updating this with an OrderID foreign key.

A screenshot of a computer code

Description automatically generated

Inputting data into the customer table, holds a unique customer ID, their name and what payment type they used.

A screen shot of a computer program

Description automatically generated

Menu table for the database, will store what menus that get used within the store. menuID is primary key for this table.

A close-up of numbers

Description automatically generated

Placed menu data into the menu table holding the information of a table ID, name and date that it can be accessed.

A close up of text

Description automatically generated

Item table works with the Menu table but instead of holding what menus are which, it contains what items are on the menu. ItemID is a primary key in this table, and it references menuID from the menu table as a foreign key.

A close up of a sign

Description automatically generated

Within the item table I had forgotten to add an itemPrice value into the table, I used the ALTER TABLE statement to add it.

A screenshot of a computer program

Description automatically generated

Using the INSERT Statement I used the sample data given to me from the client to fill the menu items with their corresponding prices. It may be hard to read but the first Int holds the ItemID, then the Item Name then the price and finally the ID to the menu that the item belongs to.

A computer code with text

Description automatically generated

Item stock table, used to hold information about items, their outlet they belong to and the overall stock amount. For this table I couldn’t get both itemID and outletID as a primary key, so I had to research into W3 Schools on how to do this, got it working by adding them in the bottom In one line.

A white background with black text

Description automatically generated

This Query is to insert stock information to the itemStock table.

A screenshot of a computer code

Description automatically generated

Outlet table contains an outlet id and the outlet location. Its good to store this information due to the sheer number of outlets scattered across the United Kingdom.



Inserting an outlet Location into the outlet Table. After talking to the client, at the moment this is just a prototype database so only one location is necessary.

A computer screen shot of text

Description automatically generated

Order table contains all order information, orderID is set as the primary key, table uses custID AND outletID as foreign keys referencing their respective table for functionality of this table.

A number of numbers on a white background

Description automatically generated

Order Date placed inside of the orders table to contain an id, date ordered what customer ordered it and what outlet it was ordered in.

This Query is used to find the Most ordered item across all the menus from the outlet.

It will take in the item table and use the itemName & ItemID columns, it will then count how many times an item is displayed in the orders table by also pairing it with its corresponding ID, whilst doing it, it will group together each item ordered and sort it from highest to lowest, in this instance I set the limit to 1 which means it will **only** show the most ordered item.

A screenshot of a computer code

Description automatically generated

The output:

A close up of a logo

Description automatically generated

## Table Sample Data

### Customer Table

A screenshot of a computer

Description automatically generated

### Item Table

A screenshot of a menu

Description automatically generated

### Item Stock Table

A screenshot of a computer

Description automatically generated

### Menu Table

A screenshot of a menu

Description automatically generated

### Orders Table

A screenshot of a computer

Description automatically generated

### Outlet Table

A close up of a computer screen

Description automatically generated

### Shift Table

A screenshot of a computer

Description automatically generated

### Staff Table

A screenshot of a computer

Description automatically generated