Content

Word Count: 3024

1. Executive Summary

McDonnel Clothing is a highly profitable company which is considering expanding its business. As it universally acknowledged that data speaks louder than words, data warehouse is a perfect resolution for Maynard Naes, the owner of the company, to analyse the business and make decisions. This document briefly explains what is data warehouse and provides a star schema data warehousing model for the McDonnel Group. This will make it straightforwardly for stakeholders to analyse their business efficiently. Using this data warehouse, stakeholders may easily access the profits, and understand the relationship among customers, products, and stores. In that case, the owners are able to determine their decisions in order to manage the future growth of their company.

Including this summary, this document consists of five parts illustrating the data warehouse for the McDonnel Clothing. Executive summary is the first part, which introducing the purpose and the structure of this document. Following is the data warehousing part, detailed discussing the definition of data warehouse and the achievements of using it. The third part is the design of the data warehouse. This part briefly introduces the star schema data warehousing model which is designed for the McDonnel Group. The forth part discusses how this data warehouse helping the McDonnel Group managing their business. This part detailed explained the way this data warehouse works and how the data presented to support decision-making. The last part is the appendix, including the data dictionary for all fact and dimension tables, the SQL statements for creating tables, and the breakdown for the assignment work.

2. Introduction to Data Warehousing

2.1 Definition of Data Warehousing

As an indispensable element of Business Intelligence, data warehousing often refers to a system which provides strategic information to management level to support business analysis and decision-making. Distinguished from operational database, data warehousing focuses more on making business strategies from macroscopic level rather than daily operational activities.

Data warehousing could be regarded as a warehouse which stores data from different databases. Bill Inmon (1992) defines four characteristics of data warehousing, which are: subject-oriented, integrated, time-variant and non-volatile.

Subject-oriented means that the analysis target of data warehousing could be one specific subject, for example, the inventory system. And integrated means that data stored in one data warehousing system could be from distinguished sources while representing same entity. Time-variant means that data warehousing needs to keep all versions of data for purpose of retrieving historical data and making comparison with transactional systems. And non-volatile means that historical data stored in data warehousing should not be changed.

2.2 Necessity of Data Warehousing

As the development of digital business technology, data amount collected by one organisation grows rapidly, which increases the difficulties in data management. Besides, it is harder to update data timely and manage data in various forms from different databases and operating systems. Especially, redundant data is an inevitable problem in business data management. However, the occurrence of data warehousing improves the data management and influences the performance of organisations further.

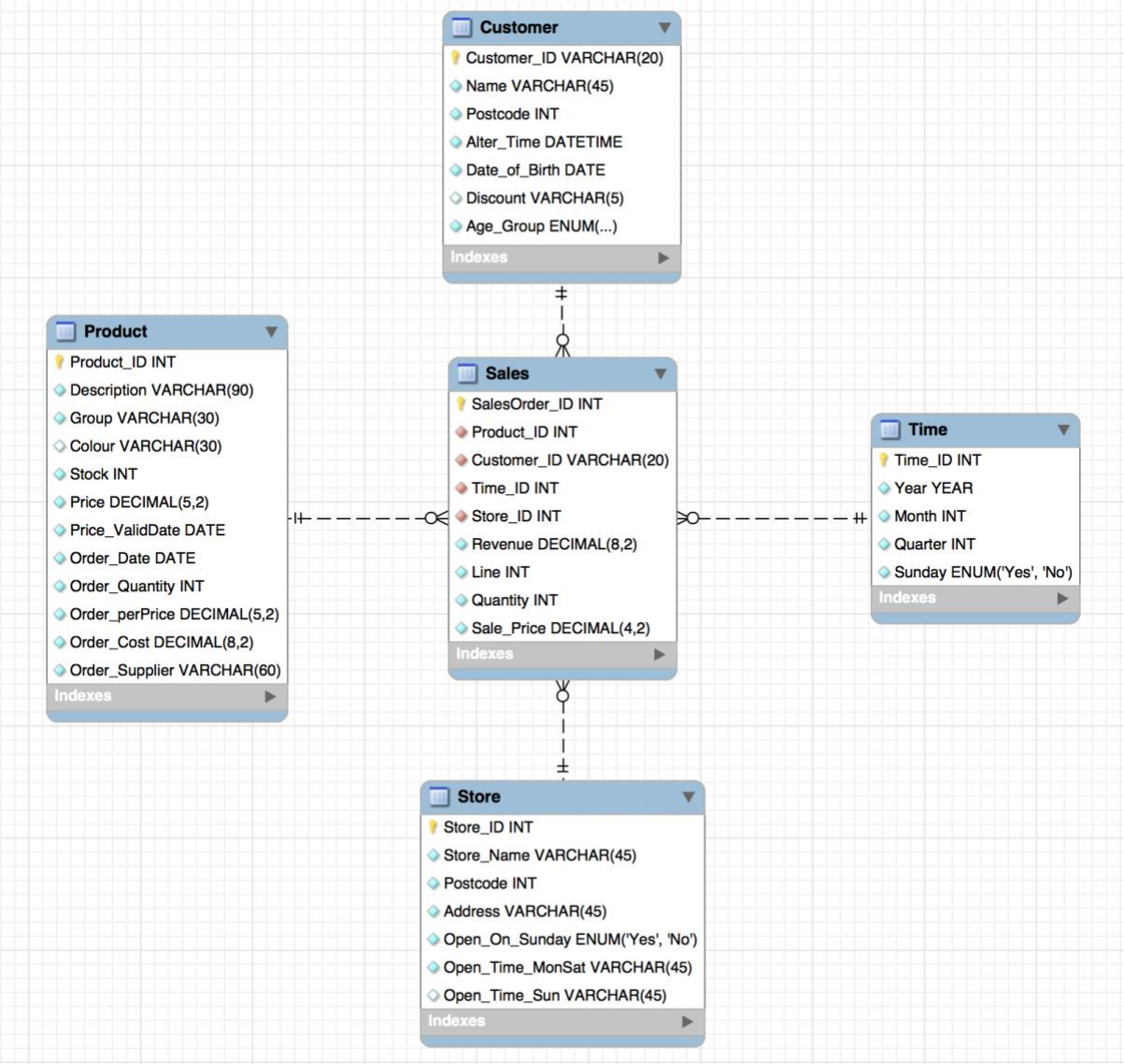
One of the most important points data warehousing provides is the maintenance of historical data. It provides the opportunity for enterprise to backtracking its past data. Based on adequate historical data, analysis of mutually influenced factors and suitable analysis approach, the development attack of one organisation could be figured out, which also guides a better strategy for organisation to plan. It could not be ignored that data warehousing provides an easier way to monitor schedule issues, cost budget and quality control.

2.3 Benefits from Data Warehousing

The first benefit from data warehousing is that it could maintain historical data for sake of business tendency analysis regardless whether historical data is stored in transactional systems or not. This benefit avoids the risk of losing historical data. The second point is that the integrated data from distinguished databases and operating systems provides decision-makers a better overview and understanding cross the organisation to expanding business in the most profitable way. This benefit improves working efficiency of end-users and decreases the barrier caused by users' unfamiliarity with data retrieval.

Another noticeable advantage is that data warehousing improves the quality of data, optimises data quality and decreases data redundancy. In addition, data warehousing presents data in consistent manners. Another merit is that data warehousing tidy data and unify their form to make end-users understand and retrieve data more easily. The strengths discussed above not only help organisation in saving resources and simplifying work process, but also support readable historical data retrieval for end-users in the future.

3. Design of the Data Warehouse



3.1 Overall

As the diagram shows above, this data warehouse model is designed as a star schema in order to help the stakeholders to easily analyse the business. It has one fact table called Sales, and four dimensions tables named as Product, Customer, Time, and Store. The detailed explanations of each tables are as follows.

3.2 Product Table

Product table is one of the four dimensions which stores the relevant data of products. This table has eleven attributes, including one unique primary key called Product_ID, which is usually as a 6-digit number. The attribute Colour can be null if there is none. The Stock attribute is for inventory.

When the Stock is under the predefined thresholds, the reorder decision will be automatically made for the relevant product. Also, the manager may frequently change the order referring to the Stock attribute of the product.

The Product table has three hierarchies: Product Details hierarchy, Price hierarchy, and Order hierarchy. The Product Details hierarchy includes the basis of this product: Product ID, Description, Group, Colour (if any), and Stock. In the Price hierarchy, there are the Price and Price_Valid_Date. The Price attribute is the unit price of the product. While in the Order hierarchy, it contains the Order_Date, Order_Quantity, Order_perPrice, Order_Cost, and the Order_Supplier. The Order_Cost attribute is calculated as Order_perPrice multiplied by Order_Quantity. Since the McDonnel Group is willing to analyse the margin of the product, it is necessary to manage the cost of ordering this product. As a slowly changing dimension, the Order hierarchy may be multiple, since one product may have several supply orders. That is to say, when the inventory manager decides to reorder, there will be 4 new columns added as a new Order hierarchy to this table. The data warehouse need these historical data to be retrieved in order to manage the cost of this product.

3.3 Customer Table

Customer table is a dimensional table to keep the customer data. This table contains 6 attributes. The unique primary key is called Customer_ID, usually formed as a combination of letters and digits.

The Postcode attribute refers to the suburb where the customer lives. According to this attribute, the McDonnel Clothing may relate the customer to the nearest store which is on sale. With the time goes by, the customer may occasionally change their house address. As the stakeholders desired to know where the profitable customers lived at the time of particular sale, it is necessary to keep the historical postcodes and the time of alteration. Therefore, when the customer changes his postcode, the user will add two new columns to his table to keep the data.

Age Group is a not null attribute aggregated the customer into three groups by age. This attribute needs to be selected as one of the three predefined age groups: '<=30', '31-50', and '>50', which means '30 years and under', '31 years to 50 years', and 'over 50 years'. The McDonnel Group may need to understand customers' shopping pattern by analysing the reports aggregated by age.

Discount attribute refers to the customer loyalty program. According to the purchase history records or the specific store events, the customer may get a 5% discount on his next purchase. This discount varies from different customers. Hence, it is not mandatory to this table.

3.4 Time Table

Time table is a dimension table storing the time data. Time ID is the unique primary key of this table. In order to figure out which time period is the most profitable, five attributes are designed in this table. Among which, the Sunday attribute is an enum datatype, which can only be chosen as 'Yes' or 'No'. This attribute is set because only the original store is open on Sunday, and the stakeholders are desired to analyse sales reports of Sundays to decide whether opening all stores on Sunday.

The Year, Month, and Quarter attributes are designed for the analysis of profits aggregated by time. The users are able to retrieve Various time periods can be calculated through this table. In that case, the stakeholders may efficiently access to the most profitable product/customer/store of various time period, as well as the most profitable time period of this year.

3.5 Store Table

Store table is one of the four dimensions which keeps the data of stores. As it shows in the diagram, this table consists of seven attributes, with the primary key called the Store_ID. This table along with the Sales table, may help identifying the most profitable products of the store. So that the stakeholders shall place these products in the prominent places of the store.

In this table there are two hierarchies, that is the Store Details hierarchy and the Open Details hierarchy. The Store Details hierarchy contains the basic details of each store, including the Store_ID, Store_Name, Postcode and Address. The Postcode refers to which suburb the store is in. This may be helpful for the analysis of the profitable store location.

The Open Details hierarchy consists of the opening time of each store, that is the Open_Time_MonSat attribute, the Open_Time_Sun attribute, and the Open_On_Sunday attribute. The Open_Time_MonSat attribute refers to the opening hours on the regular days from Monday to Saturday. And the Open_Time_Sun attribute means the opening hours on Sunday. Since not all the stores are opened on Sunday, this attribute would be null in some rows. The Open_On_Sunday attribute can only be chosen as 'Yes' or 'No'. This refers to identify if this store opens on Sundays. Since the McDonnel Group would like to expand their business further by opening all stores on Sunday, this table may make it easy for stakeholders to analyse the profits to support the decision-making.

3.6 Sales Table

Sales table is the fact table of this star schema. It is the intersection table among all tables. The sales table is designed from the business process – sales, which is the key process of the McDonnel Clothing Company.

This table consists of nine attributes, including four foreign keys. The unique primary key is called SalesOrder_ID. The foreign keys are Product_ID, Customer_ID, Time_ID, and Store_ID. This means that each line of the sales order refers to a unique customer, a unique product, a unique time and a unique store. Through this table, the stakeholders are capable of easily understanding the relationship among four dimensions and accessing to the profit analysis efficiently.

The Revenue attribute refers to how much money the customer spent on this order line. It equals to Sale_Price multiplied by Quantity. This attribute helps user to calculate the dollar sales efficiently.

The Sale_Price attribute is usually the same as the Price attribute in Product table of the relevant product. Since the McDonnel Group has a customer loyalty program, when the related customer has a discount attribute in the data warehouse, the Sale_Price would be reduced to the relevant number. This Sale_Price varies to customers and related to products, therefore, this attribute is designed as a measure places at the fact table.

4. Data warehousing used in solving Business Problems

4.1 Who are the key customers?

Data retrieved:

In this issue, data needs to be retrieved includes: 'customer ID, product ID, quantity, revenue' from Sales Table, 'customer ID, postcode, alter time, age group' from Customer table, 'time ID, year, quarter, month' from Time table and 'product ID, order price' from Product table.

Retrieval procedure:

Step 1: Select 'Time_ID' from Time table using 'Year' to define specific year field, the result should return several rows which are within the specific year period.

Step 2: Link the returned 'Time_ID' from Time table to Sales table, select the corresponding 'Customer_ID', 'Product_ID', 'Revenue', 'Quantity' and link the above 'Product_ID' to 'Product_ID' in Product Table, select corresponding 'Order_Cost' from Product table, group by Customer_ID.

Step 3: Sum the amount of 'Quantity', ''Revenue' and 'Order_Cost' based on equal customer ID in Sales Table. Using 'margin' = 'revenue' minus 'order cost' to as a new attribute. Sort customer ID by 'margin'.

Step 4: Link the customer ID from step 3 to 'customer ID' in Customer Table, select 'customer_ID', 'postcode', 'alter_time' and 'age group' from Customer Table.

Regarding various time periods, only needs to change 'year' field in step 1 to 'quarter' or 'month'.

Problem addressed:

Following the above steps, decision-maker could clearly figure out customers' contribution to this organisation based on the revenue acquired from them (revenue), quantity of multiple kinds of products they bought(quantity) and profit made from them during various time periods(margin).

For purpose of solving the problem of which suburb the most profitable customers live in, end-users could choose top 10 to top 30 rows returned above of according to the various length of time periods, then add the following step:

Step 5: group by suburb, counting customer ID as 'count'. Order by 'count'.

The row returned shows the most profitable customers live in which suburbs and the exact amount of customers.

4.2 Which products are the most profitable?

Data retrieved:

In this issue, the data which needs to be retrieved includes: 'product ID, time ID, quantity, revenue' from Sales Table, 'time ID, year, quarter, month' from Time Table and 'product ID, order cost' from Product Table.

Retrieval procedure:

Step 1: select 'Time_ID' from Time Table using 'year' to define specific year field, the result should return several rows which are within the specific year period.

Step 2: link the returned 'Time_ID' from Time table to Sales table, select the corresponding 'Product_ID', 'revenue', 'quantity' and link the above 'Product_ID' to 'Product_ID' in Product Table, select corresponding 'order cost' from Product Table, group by product ID.

Step 3: sum the amount of 'quantity', ''revenue' and 'order cost' based on equal product ID in Sales Table.

Step 4: Using 'margin' = 'revenue' minus 'order cost' as a new attribute. Sort product ID by 'margin'.

Problem addressed:

Regarding various time periods, only needs to change 'year' field in step 1 to 'quarter' or 'month'.

Following the above steps, the result (returned rows) could demonstrate the earning performance of each product sold during different time periods. Due to the results are grouped by product ID and sort by margin, decision-maker could easily find the most profitable products.

4.3 Which store location is the most profitable?

Data retrieved:

In this issue, the data which needs to be retrieved includes: 'product ID, time ID, quantity, revenue' from Sales Table, 'time ID, month' from Time Table, 'product ID, order cost' from Product Table and 'store ID, postcode, address' from Store Table.

Retrieval procedure:

Step 1: Select 'Time_ID' from Time Table using 'month' to define specific year field, the result should return several rows which are within the specific month period.

Step 2: Link the returned 'Time_ID' from Time table to Sales table, select the corresponding 'Product_ID', 'revenue', 'quantity' and link the above 'Product_ID' to 'Product_ID' in Product Table, select corresponding 'order cost' from Product Table, and link the above 'store_ID' to 'store_ID' in Store Table, group by store ID.

Step 3: Sum the amount of 'quantity', ''revenue' and 'order cost' based on equal store ID in Sales Table.

Step 4: Using 'margin' = 'revenue' minus 'order cost' as a new attribute. Sort store ID by 'margin'.

Step 5: Link the customer ID from step 3 to 'store_ ID' in Store Table, select 'store_ID', 'postcode', 'address' to be returned.

Problem addressed:

Following the above steps, the result could show the profitability of each store during selected month. Due to the results are sorted by margin, decision-maker could easily find the most profitable store, its postcode and location.

4.4 Which time periods are the most profitable?

Data retrieved:

In this issue, the data which needs to be retrieved is: 'product ID, time ID, quantity, revenue' from Sales Table, 'time ID, month, year, quarter' from Time Table, 'product ID, order cost' from Product Table.

Retrieval procedure:

Step 1: Select 'Time_ID' from Time Table using 'Sunday' = 'Yes' to confine time period as Sundays, the result should be rows whose time ID are in Sundays.

Step 2: Link the returned 'Time_ID' from Time table to Sales table, select the corresponding 'Product_ID', 'revenue', 'quantity' and link the above 'Product_ID' to 'Product_ID' in Product Table, select corresponding 'order cost' from Product Table, group by time ID.

Step 3: Sum the amount of 'quantity', ''revenue' and 'order cost' based on equal time ID in Sales Table.

Step 4: Using 'margin' = 'revenue' minus 'order cost' as a new attribute. Sort time ID by 'margin'.

Regarding various time periods, the above steps remain same except step 1, in which 'Sunday' = 'Yes' should be changed into 'Quarter', 'Month' or 'Year'.

Problem addressed:

Following the above steps, the result (returned rows) could demonstrate the earning performance of each product sold during different time periods. Due to the results are grouped by product ID and sort by margin, decision-maker could easily find the most profitable products.

5. Appendix 1 – Data Dictionary

Sales Fact Table

Attribute	Description	Source
SalesOrder_ID	Unique identifier for an order.	Sales Order attribute in Sales Order database table.
Product_ID	Unique identifier for a product.	Code attribute in Product database table.
Customer_ID	Unique identifier for a customer.	Customer ID attribute in Customer database table.
Time_ID	Unique identifier for a time.	
Store_ID	Unique identifier for a store.	StoreID attribute in Sales Order database table.
Revenue	The revenue for each product in each sale.	The Revenue in Sales table is aggregated for each type of products for each order.
Line	One transaction identifies one customer bought different product.	Line attribute in Sales Order database table.
Quantity	The number of each product in each sale.	Qty attribute in Sales Order database table.
Sale_Price	The price of single product in each sale.	Sale Price in Sales Order database table.

Customer Dimension Table

Attribute	Description	Source
Customer_ID	Unique identifier for a customer.	Customer ID attribute in Customer database table.
Name	The name of each customer.	Customer ID attribute in Customer database table.
Alter_Time	The last time of each customer updating his information.	A time generated for data warehouse.

Date_of_Birth	The birthday of each customer.	Date of Birth attribute in Customer database table.
Postcode	The postcode of each customer's home.	Postcode attribute in Customer database table.
Discount	The discount of each customer for each purchase.	A number generated for data warehouse.
Age_Group	The age group of each customer.	The Age_group in the Customer table is aggregated for each customer's age group.

Time Dimension Table

Attribute	Description	Source
Time_ID	Unique identifier for a time.	A unique number generated for data warehouse.
Year	The year for each transaction.	The Year in the Time table is aggregated for the year for each transaction.
Month	The month for each transaction.	The Month in the Time table is aggregated for the month for each transaction.
Quarter	The quarter for each transaction.	The Quarter in the Time table is aggregated for the quarter for each transaction.
Sunday	Whether the transaction happening on Sundays.	The Sunday in the Time table is aggregated for judging whether the transaction happening on Sundays.

Product Dimension Table

Attribute	Description	Source
Product_ID	Unique identifier for a product.	Code attribute in Product database table.
Description	The description of each product.	Description attribute in Product database table.
Group	The group of each product.	Group attribute in Product database table.
Colour	The colour of each product.	The Colour is generated for data warehouse.
Stock	The stock of each product for each inventory.	The Stock in the Product table is aggregated for the stock of each product for each inventory.
Price	The selling price of each product.	Unit Price attribute in Product Price List database table.
Price_ValidDate	The valid date of each product for sale.	Valid until Date attribute in Product Price List database table.
Order_Date	The date of each inventory.	Date attribute in Product Order database table.
Order_Quantity	The number of each product for each inventory.	Quantity attribute in Product Order database table.
Order_perPrice	The cost of each product for each inventory.	Cost per item attribute in Product Order database table.
Order_Cost	The total cost of each product.	The Order_Cost in the Product table is aggregated for the total cost of each product.
Order_Supplier	The supplier of each product.	The Order_Supplier is generated for data warehouse.

Store Dimension Table

Attribute	Description	Source
Store_ID	Unique identifier for a store.	StoreID attribute in Sales Order database table.
Store_Name	The name for a store.	The Store_Name is generated for data warehouse.
Postcode	The postcode of a store.	The Postcode is generated for data warehouse.
Address	The address of a store.	The Address is generated for data warehouse.
Open_On_Sunday	Whether a store opens on Sundays.	The Open_On_Sunday in the Store table is aggregated for judging whether a store opens on Sundays.
Open_Time_MonSat	The open time of a store from Monday to Saturday.	The Open_Time_MonSat in the Store table is aggregated for the open time of a store from Monday to Saturday.
Open_Time_Sun	The open time of a store on Sundays.	The Open_Time_Sun in the Store table is aggregated for the open time of a store on Sundays.

6. Appendix 2 – SQL

```
create table Product(
   `Product_ID` INT NOT NULL,
   `Description` VARCHAR(90) NOT NULL,
   `Group` VARCHAR(30) NOT NULL,
   `Colour` VARCHAR(30),
   `Price` DECIMAL(5,2) NOT NULL,
   `Price_ValidDate` DATE NOT NULL,
   `Order_Date` DATE NOT NULL,
   `Order_quantity` INT NOT NULL,
   `Order_perPrice` DECIMAL(5,2) NOT NULL,
   `Order_Cost` DECIMAL(8,2) NOT NULL,
   `Order_Supplier` VARCHAR(60),
   `Stock` INT NOT NULL
   PRIMARY KEY (`Product_ID`)
 );

create table Customer(
   `Customer_ID` VARCHAR(20)  NOT NULL,
   `Name` VARCHAR(45) NOT NULL,
   `Alter_Time` DATETIME NOT NULL,
   `Date_of_Birth` DATE NOT NULL,
   `Postcode` INT NOT NULL,
   `Discount` VARCHAR(5),
   `Age_Group` ENUM('<=30', '31-50', '>50'),
   PRIMARY KEY (`Customer_ID`)
);

create table Time(
   `Time_ID` INT NOT NULL,
```

```
    `Year` YEAR NOT NULL,
    `Month` INT NOT NULL,
    `Quarter` INT NOT NULL,
    `Sunday` ENUM('Yes','No') NOT NULL,
    PRIMARY KEY (`Time_ID`)
 );

create table Store(
    `Store_ID` INT NOT NULL,
    `Store_Name` VARCHAR(45) NOT NULL,
    `Postcode` INT NOT NULL,
    `Address` VARCHAR(45) NOT NULL,
    `Open_On_Sunday` ENUM('Yes','No') NOT NULL,
    `Open_Time_MonSat` VARCHAR(45) NOT NULL,
    `Open_Time_Sun` VARCHAR(45) NOT NULL,
    PRIMARY KEY (`Store_ID`)
);

create table Sales(
    `SalesOrder_ID` INT NOT NULL,
    `Product_ID` INT NOT NULL,
    `Customer_ID` VARCHAR(20) NOT NULL,
    `Time_ID` INT NOT NULL,
    `Store_ID` INT NOT NULL,
    `Revenue` DECIMAL(8,2),
    `Line` INT,
    `Quantity` INT,
    `Sale_ Price` DECIMAL(4,2),
    PRIMARY KEY (`SalesOrder_ID`),
    CONSTRAINT Product_Product_ID_fk
    FOREIGN KEY (`Product_ID`)
```

```
    REFERENCES Product(`Product_ID`),
    CONSTRAINT Customer_Customer_ID_fk
    FOREIGN KEY (`Customer_ID`)
    REFERENCES Customer(`Customer_ID`),
    CONSTRAINT Time_Time_ID_fk
    FOREIGN KEY (`Time_ID`)
    REFERENCES Time(`Time_ID`),
    CONSTRAINT Store_Store_ID_fk
    FOREIGN KEY (`Store_ID`)
    REFERENCES Store(`Store_ID`)
);
```

7. Appendix 3 – Work Breakdown

Name	Student ID	Contribution
Ailin Zhang	874810	1.Executive summary 3.Design of data warehouse 6.Appendix 2-AQL
Nan Wang	853883	2. Introduction to data warehousing 4. Using data warehousing solve business problems 5.Appendix 1- data dictionary

Both members discussed the design of data warehousing.

In detail, Ailin (874810) draws the model through MySQL workbench, and accomplishes part 1,3 and 6, which are: executive summary, design of data warehouse, SQL statements in Appendix 2 and form editing for this assignment respectively.

While Nan (853883) accomplished part 2,4,5 and 7, which are: introduction to data warehousing, using data warehousing addresses business problems, data dictionary and work breakdown in Appendix 1 and 3 respectively.