ISYS90086
Data Warehousing
Data Warehouse Design-Overhill Winery

Hangyu Pan 1050937 hjpan@student.unimelb.edu.au
Zinan Cheng 1004909 zinanc@student.unimelb.edu.au

Content

1. Executive Summary

Overhill Winery is a winery which specializes in producing cool climate wines. The overall business contains growing 3 grape varieties as the raw materials which are pinot noir, merlot and pinot grigio, manufacturing several brands of white wine and red wine and selling wines to the distributors or wine merchants. At current stages, the business operation has been fully supported through two information systems. The production system is to record the data of manufacturing processes and the merchant systems is widely used for storing each sale and the corresponding detail.

With the expectation of business growth, it is necessary for them to implement effective and efficient decision procedures based on the data. Therefore, data warehousing techniques will play a significant role in this improvement. Business intelligence dashboard developed through data warehouse tools does contribute to operation, reporting and strategic decision.

This report is to deliver the data warehouse introduction, establish a data warehouse for the business and provide solutions for the current concerned business issues within the next four parts.

2. Data Warehousing

2.1. Data Warehouse and Data Warehousing

A Data warehouse is a kind of database for decision support. It is independent from the transactional database. A data warehousing can be regarded as a process for collecting data from various sources and developing business insight as well as producing strategic information. Joseph (2013) puts forward that the current data warehouse can be considered as a system with multiple functions of extracting, cleaning, confirming and sourcing data into a dimension data store. It also supports and implements querying and analysis within the information system. In this report, the data warehouse for Overhill Winery is going to be developed on the initial stage with the basic structure.

C.Date (2003) defines that “data warehouse is a repository of enterprise or business databases which provides a clear picture of current and historical operations of organizations.” With this definition, it is more likely to be considered as an overall requirement of your data warehouse design. To be more specific, Inmon (1996) points that data warehouse is supposed to have four characteristics which are subject oriented, integrated, time variant and non-volatile. To achieve them, the dimension modelling has to be implemented. A subject-oriented data warehouse is to define a specific target to analyze. With integrated function, data warehouse is able to combine data varieties with one entity. In terms of time variant, data warehouse can retain data within various time frames in order to reflect historical difference from the data. Non-volatile does mean that data stored in data warehouse is relatively stable. Generally, it is stored without change for a long time.

2.2. Why data warehouses are necessary?

With the business expansion in the current trend, most of organizations puts the big data as one of the main features in the development. Although as for the architecture of micro-service, there have been multiple independent transactional database or data marts behind business application, it is difficult to make sustainable procedure of analysis and data mining within the independent transactional databases. Data warehouse is one of the solutions to improve the value of data from multiple sources.

The necessity of data warehouse is for the sake of business analysis. Business analysis does affect the competitiveness of each group. With various strategic information requirements, data warehouse possesses more flexibility and independency. Firstly, data warehouse resolves issues caused by the multiple independent

transactional database. It is able to collect data from various databases through data integration. Not only does it reduce the complexity of querying but also it optimizes data quality and present the data consistency. With concepts of data warehousing, it also simplifies the establishment of dashboard which is necessary for the end users. Therefore, data warehouse cannot be ignored in most of analyzing tools.

2.3. Benefits of Data Warehouse

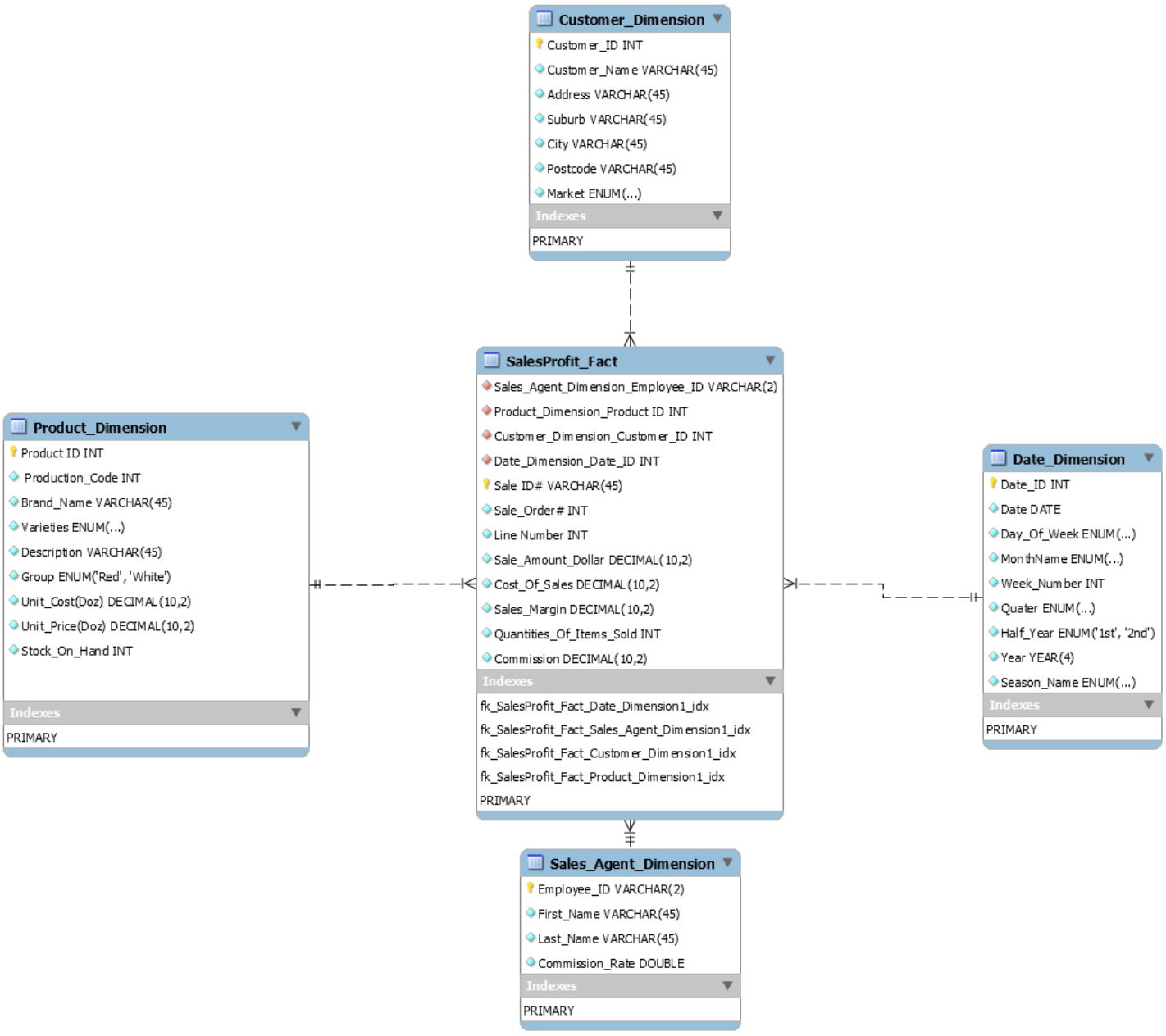
With the implementation of data warehouse, there are several significant benefits.

In terms of the independency of data warehouse, it does enhance the efficiency data retrieve, as it does not slow down the operating system and can compute through its own resources. At the same time, the transactional management could be controlled simply to avoid impact of analysis. For instance, when the group has strategies to increase overall retail price level, it can directly change it on the transactional database and cover the old price. In the data warehouse, new dataset of price is also produced but not replace old dataset of price. For the purpose of comparison, it definitely adds value of data within time frame when considering the strategic information. The independency of data warehouse does also help data access control, as data warehouse may contain some information which cannot be disclosed. Data warehouse has certain impact to ensure the security and privacy of data.

With integration of data warehousing, it can establish a general data model for different areas without restriction of data sources. As well as that, the data warehousing tool support us to transform data or pre-process data before loading the data into the warehouse. It simplifies reporting, analyzing and visualizing process. Apart from that, the simple design of data warehouse architect such as star schema or snowflake schema has strong abilities of extension and expansion to cope with complex searching on a sale with higher dimensions.

Overall, data warehouse does enhance performance of systems and simple procedures of business analysis.

3. Design of the data warehouse



3.1 The main idea of design

Overhill Winery is going to address five problems related to the fields of profit and sales. With its business operation, there are two main transaction systems. In this data warehouse design, it is going to integrate the data from those main databases with dimension modelling. As well as that, the main measures in the fact is profit or margin. To identify the profitability of product, date, customer and market as well as sales agents, it will introduce four dimensions which are products, customers, date and sales agents. The overall design is presented with star schema. It also implements surrogate keys in order to cope with various types of changing data during the analysis as well as enhance the querying performance.

3.2 Fact Table – SalesProfit

Within the design of fact table, each of sales row of SalesProfit_Fact presents a line or a product with its quantities in a sale order. It also represents the level of grain. It is critical to notice that the fact is composed of 4 dimensions

(product id, customer id, date id and sales agent id) basically instead of sale order. However, with purpose of storing various filters, it implements sale id as surrogate keys as the primary key for the 4 dimensions.

There are several significant facts or measures. Firstly, Sale Amount Dollar is the actual price per product for a sale order. This attribute does use frequently to compare the advised price and cost and to calculate the corresponding measure. Secondly, cost of good sale represents each cost for selling the product. This attribute is the actual cost in a sale, as the actual cost for selling the product does not only consider the cost of production on the product. Then, Sales Margin is calculated by the difference between sale amount dollar and cost of good sale. The attribute does contribute to improve performance of loading the query and simplify querying. Sale Amount Dollar multiplying the commission rate is to reflect the commission for selling this product in a sale order. However, regarding the change of commission rate, the commission here is necessary to be calculated with commission rate under the sale date. It is not calculated based on the commission rate from the commission rate in the sales agent table. Finally, the Quantities of Items Sold reflects the number of a product sold in a sale order.

Sale Order Number is the higher level of grain for the sale order detail. It is to support querying as aggerate reference.

3.3 Date Dimension

Date is one of dimensions in the design to store time data. Date ID as surrogate key is the primary key for this entity. Basically, the date stores the detail of date in the format of “dd-mm-yyyy”. With requirements of some features of date, the table also contains day of week including Monday to Sunday, Month name from January to December, 4 quarters and 4 seasons as well as half year. Although there are multiple attributes in the date dimension, it is necessary to add multiple restrictions within initial date setting. The relationship of one-to- many between date and sale fact table also refers that a sale can only have one specific date feature but a date has many sales or not.

With those attributes for the date dimension, it does simplify the querying when it is going to analyze the facts from various date point. For instance, when it is going to compare the yearly profit. It can be grouped the profit data through the year dimension instead of coding much case to identify the year of date in each sale order.

3.4 Customer Dimension

In the entity of customer, the primary key is Customer ID. There are 6 attributes in the table of customer. Considering the unique customer with unique market, the customer ID here is supposed to be the same as the customer number from merchant system of Overhill Winery. Therefore, customer ID is not surrogate key in the data warehouse but it is surrogate key in the transaction database.

With more grains of address in the customer table, it is able to aggregate customers into various group such as suburb, city and market. Base on the customer ID, the customer detail would be updated during each integration with transaction database. It is significant to notice that Markets is following with Customer ID uniquely. Therefore, it is not necessary to consider the circumstance that a customer from multiple markets.

In the table of facts from sales, customer ID is regarded as the foreign key. The relationship of one-to-many between customer and SalesProfit fact table implies that the customer can put many sales but one sale can only have customer.

3.5 Sales_Agent Dimension

In the entity of sales agent, employee ID is the primary key. There are 3 attributes for this dimension table. It is significant to notice that the data on this table is updated once the change happens on the transactional database. The employee ID is mainly used to be the reference to load data from the commission report from transaction system. The relationship of one-to-many between SalesProfit fact table declares that there is only one sales agent for each sale but sales agent can have many sales.

3.6 Product Dimension

Product is also one of dimensions in this star schema. The table contains 7 attributes which are production code, brand name, varieties, description, description, group, unit cost, unit price and stock on hand. As well as that, product dimension is extract data from the production system. The relationship of one-to-many between SalesProfit fact table declares that a product can relate to many sales but a sale in fact table has only one product id. The product id is the surrogate key for this table.

There are multiple attributes for the purpose of aggerating as reference. Brand name indicates 3 sperate brands. Group attribute shows 2 types of wine. Varieties is to reflect the 3 types grapes for the productions. Production code is to represent the item in various production line. It does help further analysis if it is going to be after the profitability of item. The code can be used to find more information on the specific product including its production volume and year in the transactional database. However, in the data warehouse, it does help identify the item.

There are two price attributes here. The unit cost does represent cost of production and the unit price reflects advised sale price.

4. How does data warehousing address and solve the business problems?

4.1. Which products are the most profitable?

To address the most profitable products, it is significant to grab the product id with its features which are base product and wine type. The information retrieved is about product, margin per product, cost of good sale per product, quantities of item sold as unit_sales.

Querying procedures:

1. **Choose the time period with date dimension:**
   To select date features in the date dimension, it needs to grab the groups of date id with the same feature as the conditions. For example, with the query-“SELECT date_id from date_dimension WHEN YEAR=2019 and SEASON=’summer’ and month=1”, it will all of date id which is in the summer of 2019.
2. **Link date id with SalesProfit_Fact and group measues**
   According to the result of date id from last step, it can select sales id within this date id restriction follow by the subquery from last steps. At the same time, the product id corresponding to each sale can be found. Then, it is necessary to calculate measures. Regarding to measures of product, the average measure is also great point to address it. To grab the average sale amount of the product id, it can use quantities of items sold to multiply the sale amount dollar to grab the total sale amount. Then, it can divide by the total quantities of item sold within a product id to generate the actual average sale amount for the product within various time period. Similarly, margin and cost of good sale for each product can also be used to address the problem.
3. **Link product id with product dimension**
   According to the last step of sales id from last step, it is easy to grab the unique product id responding to each sale. With joining to product dimension with product id, the details about the product can also put into the table such as product group for wine type and description for the base product.

4. **Order by measures and address the problem**
   Through the order querying, it can find out the profitable product up to time period in terms of various margin measures.

4.2. Who are the key customers?

To address the key customers, it needs to retrieve information about customers, quantities of sales placed, total sales, cost of sales and margin within specific time period.

Querying procedures:

1. **Choose the time period with date dimension:**
   Similar to the 1st step in the querying procedures of the most profitable product. The difference is to delete the month condtion.
2. **Link date id with SalesProfit_Fact and group measues**
   According to the result of date id from last step, it can select sales id within this date id restriction follow by the subquery from last steps. At the same time, the customer id corresponding to each sale can be found. Then, it is necessary to calculate measures. For the total sale amount of the sale id, it can use quantities of items sold to multiply the sale amount dollar to grab it. Also, the margin per sale id can be calculated by the sales margin multiplying the quantities of items sold. Similarly, cost for this sale can be work out. Finally, it is to group by the customer id to get total unit sale (Quantities of Items Sold), total sale amount, total margin and total cost per customer id.
3. **Link customer id with customer dimension**
   According to the last step of sales id from last step, it is easy to grab the unique customer id corresponding to each sale. With joining to customer dimension with customer id, the details about the customer can also put into the table.
4. **Order by measures and address the problem**
   Through the order querying, it can find out the best customer up to time period. In addition, there are multiple features related to key customers. They do contribute to customer relationship and address the key features of key customers.

4.3. Which market is the most profitable?

To address the profitable market, the procedures is similar to the key customers, as the market is one of features for the customers. The measures implemented here is from total sale and unit sale.

Querying procedures:

1. **Choose the time period with date dimension:**
   Similar to the 1st step key customer querying procedures. The difference is to replace the condition of season with month and the year setting is 2019.
2. **Link date id with SalesProfit_Fact**
   According to the result of date id from last step, it can select sales id within this date id restriction follow by the subquery from last steps. At the same time, the customer id corresponding to each sale can be found.
3. **Link customer id with customer dimension and group the market features**
   According to the last step of sales id from last step, it is easy to grab the unique customer id corresponding to each sale. With joining to customer dimension with customer id, the details about the customer can also put into the table especially market. Then, it is to group by markets to aggerate the total sales and unit sales.
4. **Order by measures and address the problem**
   Through the order querying, it can find out the profitable market up to time period.

4.4. Which time periods are the most profitable?

To address the profitable time periods, the measures implemented are quantities of item sold (unit sales), sale amount dollar and sales margin.

Querying procedures:

1 **Link date id with SalesProfit_Fact**
The first step is to join date id with fact table.

2 **Group the date features**
Within date_dimension, there are multiple features such as week number for weekly sale data, month for monthly sale data and year for yearly sale data etc to group up. Then, the aggerate data of unit sale, dollar sales and margin per defined period can be generated.

3 **Order by measures and address the problem**
Through the order querying, it can find out the profitable time period.

4.5. Who are the Key sales agents?

To address the key sales agents, it implements total commission as the measures addressed within various time periods.

Querying procedures:

1. **Choose the time period with date dimension:**
Similar to the 1st step of profitable products. Here, it is able to customize time period based on the analysis.
2. **Link date id with SalesProfit_Fact and group by sales_agent**
According to last step, the related sale id can be generated within date id from it. Then, the total commission can be sum up through group by employee_id.
3. **Link sales_agent_id**
In this step, the sales_agent_dimension is going to join SalesProfit_Fact on the employee_id.
4. **Order by the measures and address the problem**
Through the order querying, it can find out the key sales agents with largest commission amount.

5. Appendix

5.1. Data Dictionary

Customer dimension

Attribute	Description	source
Customer id	Unique identifier for a customer	Customer id attribute in customer database table
Customer name	The name of each customer	Customer name attribute in customer database table
address	The address detail of each customer	The address attribute in customer database table
suburb	The place where customer live outside of city	The suburb address attribute in customer database table
city	The place where customer live in the city	The city address attribute in customer database table
postcode	A group of number that used as a part of address	The postcode attribute in customer database table
market	A specific target place where the customer stay	The market attribute in customer database table

Product dimension

Attribute	Description	source
Product id	Unique identifier for the product	Product id attribute in product database table
Production code	Another code for the product	Production code attribute in product database table
Brand name	The name of each branch	Product brand name attribute in product database table
varieties	Three different kinds of product	Three varieties attribute in product database table
description	Some specific information of the product	A brief description in product database table
group	Two different color of product	Two kinds of color in product database table
Unit cost	The cost of each unit of product	The amount cost for each unit of product in product database table
Unit price	The price sales of each unit of product	The sales price of each unit product in product database table
Stock on hand	The number of product stored	The number of products on hand in product database table

Sales profits fact

Attribute	Description	source
Sales agent dimension-employee id	Unique identifier for employee in sales agent dimension	Employee id attribute in sales agent database table
Product dimension-product id	Unique identifier for product in product dimension	Product id attribute in product database table
Customer dimension-customer id	Unique identifier for customer in customer dimension	Customer id attribute in customer database table
Date dimension-date id	Unique identifier for date in date dimension	Date id attribute in date database table
Sale ID	Unique identifier in fact table	Surrogate key
Sale order	The order number of each sales	The sale order attribute in sales profits fact table
Line number	The numbers of line of each sale	The line number in sales profits fact table aggregated for total line of sales
Sale amount dollar	The actual sale amount of product per sale	The sale amount in dollar in sales profits fact table aggregated for the total sales
Cost of good sale	The actual cost amount of product per sale	The cost of sales in sales profits fact table aggregated for the total cost of each sale
Sales margin	The money obtained excluding the cost by each sale	The Sales margin in sales profits fact table aggregated for the total net earnings
Quantities of items	The number of items in each sale	The quantities of items in sales profits fact table aggregated for the number of items sold
commission	The commission fee cost in each sale	The commission in sales profits fact table aggregated for the total fee paid for agents

Sales agent dimension

Attribute	Description	source
Employee id	Unique identifier for employee	Employee id attribute in sales agent database table
First name	The first name of employee	First name attribute in sales agent database table
Last name	The last name of employee	Last name attribute in sales agent database table
Commission rate	The rate of commission the sales agent can obtain	The commission rate attribute for each sales agent in sales agent table

Date dimension

Attribute	Description	source
Date id	Unique identifier for date	Date id attribute in date database table
date	The specific date for each trade	Specific date attribute in date database table
Day of week	The day of 7 days in a week for each trade	Day of week attribute in date database table
Month name	The month of one year for each trade	Month attribute in date database table
Week Number	The number of week of one year (1-52)	Week attribute in date database table
quarter	One of fourth year for each trade	Quarter attribute in date database table
Half year	One of second year for each trade	Half year attribute in date database table
year	The detail of a year for each trade	Year attribute in date database table
Season name	The season for each trade	Season attribute in date database table

5.2. SQL

```
-- -----------------------------------------------------
-- Table `Project1OverhillWneryDW`.`Date_Dimension`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `Project1OverhillWneryDW`.`Date_Dimension` (
  `Date_ID` INT NOT NULL AUTO_INCREMENT,
  `Date` DATE NOT NULL,
  `Day_Of_Week` ENUM('Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday') NOT
NULL,
  `MonthName` ENUM('Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec') NOT NULL,
  `Week_Number` INT NOT NULL,
  `Quater` ENUM('1st', '2nd', '3rd', '4th') NOT NULL,
  `Half_Year` ENUM('1st', '2nd') NOT NULL,
  `Year` YEAR(4) NOT NULL,
  `Season_Name` ENUM('Spring', 'Summer', 'Autumn', 'Winter') NOT NULL,
  PRIMARY KEY (`Date_ID`))
ENGINE = InnoDB;


-- -----------------------------------------------------
```

```
-- Table `Project1OverhillWneryDW`.`Sales_Agent_Dimension`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `Project1OverhillWneryDW`.`Sales_Agent_Dimension` (
  `Employee_ID` VARCHAR(2) NOT NULL,
  `First_Name` VARCHAR(45) NOT NULL,
  `Last_Name` VARCHAR(45) NOT NULL,
  `Commission_Rate` DOUBLE NOT NULL,
  PRIMARY KEY (`Employee_ID`))
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `Project1OverhillWneryDW`.`Customer_Dimension`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `Project1OverhillWneryDW`.`Customer_Dimension` (
  `Customer_ID` INT NOT NULL AUTO_INCREMENT,
  `Customer_Name` VARCHAR(45) NOT NULL,
  `Address` VARCHAR(45) NOT NULL,
  `Suburb` VARCHAR(45) NOT NULL,
  `City` VARCHAR(45) NOT NULL,
  `Postcode` VARCHAR(45) NOT NULL,
  `Market` ENUM('Vic', 'Mkt', 'RoAu') NOT NULL,
  PRIMARY KEY (`Customer_ID`))
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `Project1OverhillWneryDW`.`Product_Dimension`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `Project1OverhillWneryDW`.`Product_Dimension` (
  `Product ID` INT NOT NULL AUTO_INCREMENT,
  ` Production_Code` INT NOT NULL,
  `Brand_Name` VARCHAR(45) NOT NULL,
  `Varieties` ENUM('Pinot_noir', 'Merlot', 'Pinot_Grigio') NOT NULL,
  `Description` VARCHAR(45) NOT NULL,
  `Group` ENUM('Red', 'White') NOT NULL,
  `Unit_Cost(Doz)` DECIMAL(10,2) NOT NULL,
  `Unit_Price(Doz)` DECIMAL(10,2) NOT NULL,
  `Stock_On_Hand` INT NOT NULL,
  PRIMARY KEY (`Product ID`))
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `Project1OverhillWneryDW`.`SalesProfit_Fact`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `Project1OverhillWneryDW`.`SalesProfit_Fact` (
  `Sales_Agent_Dimension_Employee_ID` VARCHAR(2) NOT NULL,
  `Product_Dimension_Product ID` INT NOT NULL,
```

```
  `Customer_Dimension_Customer_ID` INT NOT NULL,
  `Date_Dimension_Date_ID` INT NOT NULL,
  `Sale ID#` VARCHAR(45) NOT NULL,
  `Sale_Order#` INT NOT NULL AUTO_INCREMENT,
  `Line Number` INT NOT NULL,
  `Sale_Amount_Dollar` DECIMAL(10,2) NOT NULL,
  `Cost_Of_Sales` DECIMAL(10,2) NOT NULL,
  `Sales_Margin` DECIMAL(10,2) NOT NULL,
  `Quantities_Of_Items_Sold` INT NOT NULL,
  `Commission` DECIMAL(10,2) NOT NULL,
  INDEX `fk_SalesProfit_Fact_Date_Dimension1_idx` (`Date_Dimension_Date_ID` ASC) VISIBLE,
  INDEX `fk_SalesProfit_Fact_Sales_Agent_Dimension1_idx` (`Sales_Agent_Dimension_Employee_ID` ASC)
VISIBLE,
  INDEX `fk_SalesProfit_Fact_Customer_Dimension1_idx` (`Customer_Dimension_Customer_ID` ASC)
VISIBLE,
  INDEX `fk_SalesProfit_Fact_Product_Dimension1_idx` (`Product_Dimension_Product ID` ASC) VISIBLE,
  PRIMARY KEY (`Sale ID#`),
  CONSTRAINT `fk_SalesProfit_Fact_Date_Dimension1`
    FOREIGN KEY (`Date_Dimension_Date_ID`)
    REFERENCES `Project1OverhillWneryDW`.`Date_Dimension` (`Date_ID`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_SalesProfit_Fact_Sales_Agent_Dimension1`
    FOREIGN KEY (`Sales_Agent_Dimension_Employee_ID`)
    REFERENCES `Project1OverhillWneryDW`.`Sales_Agent_Dimension` (`Employee_ID`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_SalesProfit_Fact_Customer_Dimension1`
    FOREIGN KEY (`Customer_Dimension_Customer_ID`)
    REFERENCES `Project1OverhillWneryDW`.`Customer_Dimension` (`Customer_ID`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_SalesProfit_Fact_Product_Dimension1`
    FOREIGN KEY (`Product_Dimension_Product ID`)
    REFERENCES `Project1OverhillWneryDW`.`Product_Dimension` (`Product ID`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

5.3. Work Breakdown

Hangyu Pan: Executive Summary; Data warehousing; Design of the data warehousing; Address the business problem; SQL; Work Breakdown; Reference

Zinan Chen: Data warehousing; Design of the data warehousing; Data Dictionary; SQL; Work Breakdown

6. Reference

C. Date, (2003), “Introduction to Database Systems”, 8th ed., Upper Saddle River, N.J.: Pearson Addison Wesley.

Inmon W.H., “Building the Data Warehouse”, Second Edition, JWiley and Sons, New York, 1996.

Joseph, M. V. (2013). Significance of data warehousing and data mining in business applications. *International Journal of Soft Computing and Engineering (IJSCE) ISSN*, 2231-2307.