Academic year 2019/2020

**Autumn Term**

**Data Modelling and OLAP Techniques for Data Analytics**

**CASE STUDY SCENARIO: Blue Sky Online Consumer Electronics Retailer**

**Student ID 15027012**

**10TH/01/2020**

Table of Contents

[1. Introduction 3](#_Toc29584873)

[2. Analysis and design of a dimension data model 3](#_Toc29584874)

[3.1. Identification of grain and hierarchy explanations 4](#_Toc29584875)

[3.2. Fact table measures and dimensions 4](#_Toc29584876)

[3.3. Identification of fact table 6](#_Toc29584877)

[3.4. Definition of the chosen schema and its relations as ER diagram 8](#_Toc29584878)

[4.1. Implementation of Star schema 10](#_Toc29584879)

[4.1.1. Designing the Relational database 10](#_Toc29584880)

[4.2. Creation of a database 11](#_Toc29584881)

[4.3. Creating dimensions 11](#_Toc29584882)

[4.3.1. CustomerDim table 12](#_Toc29584883)

[4.3.2. The Datedim table 13](#_Toc29584884)

[4.3.3. SellingChannelDim table 13](#_Toc29584885)

[4.3.4. PaymentDim table. 14](#_Toc29584886)

[4.4. Fact table 15](#_Toc29584887)

[4.5. Primary and foreign constraints 16](#_Toc29584888)

[6. Populating and implementing the created data warehouse 16](#_Toc29584889)

[6.1. CustomerDim table 18](#_Toc29584890)

[6.2. DateDim table 21](#_Toc29584891)

[6.3. Paymentdim table 22](#_Toc29584892)

[6.4. SellingChannelDim table 24](#_Toc29584893)

[7. Implementation of ETL processes to populate a fact table 26](#_Toc29584894)

[8. SQL statements if dimensions and fact tables are populated correctly 29](#_Toc29584895)

[8.1. Customerdim table 30](#_Toc29584896)

[8.2. Datedim table 30](#_Toc29584897)

[8.3. Paymentdim table 31](#_Toc29584898)

[8.4. SellingChanneldim table 32](#_Toc29584899)

[8.5. ORDERFACTS table 32](#_Toc29584900)

[9. The design, implementation and analysis of the OLAP Cube 33](#_Toc29584901)

[10. Critical reflection 33](#_Toc29584902)

[11. Conclusion 34](#_Toc29584903)

[12. Reference: 35](#_Toc29584904)

# 1. Introduction

Founded in 1990, Blue Sky Online is an online consumer electronic retailer conducting online business across UK and Europe through several outlets including its own website, Amazon, eBay, and Tesco. Presently, the company offers a range of products in more than 500 brands nearly prearranged in ten categories.

However, the dynamic and highly competitive nature of consumer electronics have led to different decision-making challenges to pricing, targeting high value customers, consumer satisfaction and product offerings. Due to a high customer base across the UK and Europe, marketing and customer relationship managers couldn’t utilise the customer base since it is not maintained properly with important information missing, and customer details and transactions managed by different applications such as spreadsheet and google docs.

The company aims to manage and analyse customer interactions to improve its business relationships with customers, facilitate customer retention and drive sale growth.

To meet the company’s aims, a dimensional data model will be designed, and a data warehouse built where we execute SQL queries to create and populate fact and dimension tables.

However, the implemented data warehouse should combine customer information and sale transactions into a central corporate repository so that managers can produce reports and do analysis that would enable them to give useful information for their decision-making activities.

# 2. Analysis and design of a dimension data model

Dimensional modeling is an established technique for making databases simple (Kimball and Ross, 2013).

A dimensional model is a data construction method enhanced for data warehousing tools. The idea of dimensional modelling developed by Kimball and consists of fact and dimensional tables. The elements of dimensional model include fact, dimension, attributes, fact table and dimensional table.

The accuracy in generating dimensional modelling determines the successful implementation of the data warehouse. However, to meet the creation of a dimension model, the following steps will be taken.

# 3.1. Identification of grain and hierarchy explanations

The grain is a detail level of data recorded. The best practice is to define grain at the lowest level possible because any business queries at the higher grain level can be assessed by aggregation of lower level data. Considering the Blue Sky business case, the lowest available grain for time from the collected data is a daily record of customers purchases. Generally, this would be sufficient to determine longer term trends as well as, for example, seasonality in sales but insufficient to find out the most active consumption hours within a day. However, this is not a part of the business requirements and beyond the data derived from assessible data sources.

The grain for Blue Sky business data available.

|  |
| --- |
|  |
| Date Key  Customer Code  Payment ID  Selling Channels  Commission Rate  Total Cost  Total Retail Price |

Figure 1: Grain summary

The grain underlines the keys and dimensions of the fact table subsequently used in the schema. Hierarchies can be present in Date dimension if the most detailed level used. Transactions are recorded at particular day, which corresponds to the week, then month, quarter and year of the purchase. Therefore, if required for the analysis, data can be aggregated to weekly, monthly, quarterly or annual.

# 3.2. Fact table measures and dimensions

Facts are numerical measurements (values) that characterise a specific business activity. For example, sales figures are numerical measurements that signify product and sales services (Coronel, Morris and Rob, 2013).

A fact table is the central table in a star schema of a data warehouse. It stores quantitative information for analysis and is often denormalized. A fact table holds data to be analysed and a dimension table stores data about the ways in which the data in the fact table can be analysed.

The attributes of a fact table are a set of foreign keys intersecting dimension tables plus numeric measurements. Fact tables provide the (usually) additive values that act as independent variables by which dimensional attributes are analysed. Fact tables are often defined by their granularity.

The fact table comprises of facts that are linked through their dimensions (Coronel, Morris and Rob, 2013). Facts are usually stored in a fact table that is the centre of the star schema. A fact table that does not contain any measure is a [fact-less fact table](http://dwgeek.com/data-warehouse-fact-less-fact-examples.html/).

Dimensions are qualifying features that offer additional standpoint to a particular fact. For instance, customer sales transactions might be compared by selling channels from region to region and from one period to the next. In consequence, dimensions are the starting point through which you study facts (Coronel, Morris and Rob, 2013).

Attributes are frequently used to search and filter facts. Dimensions offer descriptive characteristics about the facts through their attributes. Therefore, the data warehouse design will define common business attributes that will be used by managers to narrow a search, group information, or describe dimensions. A transaction fact table is a fact table that contains measures, keys to dimension tables, and degenerate dimensions, if applicable. It consists a primary key which is usually identified by a single key field.

When using this fact table to examine the current state of something going through the business process, all rows should be summed through the latest time period that is being examined. The date key, payment ID, Invoice and customer code are primary keys. A Primary key (PK), an identifier composed of one or more attributes that uniquely identifies a row (Coronel, Morris and Rob, 2013). The Foreign keys are attributes in one table whose values must match the primary key in another table or whose values must be null (Coronel, Morris and Rob, 2013).

Further explanation of the dimension name and reasons why the attributes were chosen and the corresponding attributes. However, the measures in the fact table will be the commission rate, total cost and total retail price.

Date ID: This describes anything that specifies the periods, month, quarter and year from the best and most profitable customers. The attributes under the Date ID are the Date Key, Month of Date, Calendar Year, Calendar Quarter.

Selling Channels: This simply contain the name of the online store with its code and the commission rate. The attributes are the selling channel key, Name and Commission Rate.

Payment Data: Contain the name of card the customer used to make a transaction. The attributes are the Payment data key, Store Name, Store Code and Commission Rate.

Customer Details: This contain customer’s identification details when making a purchase. From the data given there are two customer details, and this was merged together as one dimension the following attributes are Customer Code, Invoice Number key, Selling Channel, Total Cost, Total Retail Price, Date, Income

# 3.3. Identification of fact table

The idea of a dimension is an important and differentiating notion for multidimensional databases (Jensen, Bach Pedersen and Thomsen, 2010). Therefore, dimension tables are fundamental companions to the fact table and comprise of the textual setting with questions such as “who, where, when, how, what and why” related to the business process measurement event (Kimball and Ross, 2013). We, therefore, identify dimensions in adequate detail to describe things such as the clients and properties at the correct grain (Connolly and Beg, 2010).

Facts must be numerical and additive. Also, additional facts can be added to the fact table anytime.

The dimensions in the fact table connect the facts to the related dimension tables. Some dimensions are hierarchical, such as postcode, city and country, location and geography respectively (Ibm.com, 2019).

Dimensions of the fact table can be identified by answering the following questions:

* -who is involved: customer (age group, income group, marital status and postcode/City/Country), seller?
* -what is involved: product (name, category, brand, price),
* -where does it occur: sale channel (intermediary store)?
* -when does it occur; month, quarter and year required
* -how does it occur: payment method, invoice number?
* -why does it occur: customer interactions (first time/ recurrent)?

From Blue Sky business perspective, the date dimensions (Date\_key) as a foreign key (FK) are crucial for every fact tables as facts are a sequence of observations.

The date dimensions answer the first question: When does it occur: month, quarter and calendar year?

Who is involved: customer (age group, income group, marital status and postcode/City/Country), seller? The Customer\_code (FK) will represent this information in the fact table.

How does it occur: payment method? The PaymentID (FK) will contain all the information of the cards used to pay for the order.

Where does it occur: sale channel (online store)? The SellingChannel (FK) will indicate where orders are taken from.

What is involved: name of products, category, brand, price, invoice number, total cost and total retail price

Fact table.

|  |
| --- |
| ORDERFACTS |
| Date\_key {FK}  Customer\_code {FK}  PaymentID {FK}  SellingChannel {FK}  InvoiceNumber  TotalCost  TotalRetailPrice |

Figure: 2 Summary of facts table

A single key field distinctively identifies dimension rows. These dimensions contain primary keys assigned with sequential integers starting from one. These dimensions are linked to the fact tables by a type of primary key called the surrogate key.

From the information above, the Date dimension with a Date\_key(PK) linked to the Date\_key(FK) in the fact table, the Customer details dimensions with a Customer\_code (PK) linked to the Customer\_code(FK) in the fact table and the Payments Data dimension with the PaymentID(PK) is linked to the PaymentIDy (FK) in the fact table, the Selling Channels dimensions with the SellingChannel (PK) linked to SellingChannel (FK).

# 3.4. Definition of the chosen schema and its relations as ER diagram

Star Schema

In data warehousing, a star schema is the simplest form of a dimensional model, in which data is organized into facts and dimensions. It is called a star schema because the diagram resembles a star, with points radiating from a centre. The centre of the star consists of fact table and the points of the star are the dimension tables

Star schemas are optimized for querying large data sets and are used in data warehouses and data marts to support [OLAP](https://searchdatamanagement.techtarget.com/definition/OLAP) cubes.

The main characteristics of star schema are:

* Simple structure, making it easy to understand schema
* Great query effectives with it having small number of tables to join
* Relatively long time of loading data into dimension tables as de-normalization of redundancy data caused that size of the table could be large.
* The most commonly used in the data warehouse implementations and is widely supported by a large number of business intelligence tools.

**BSORE Star Schema**

|  |
| --- |
| **ORDERFACTS** |
| Customer\_code (FK)  Date\_key (FK)  PaymentID (FK)  SellingChannel (FK)  InvoiceNumber  TotalCost  TotalRetailPrice |

|  |
| --- |
| **DateDim** |
| Date\_key (PK)  CalendarYear  CalendarQuarter  CalenderOFMonth |

|  |
| --- |
| **PaymentDim** |
| PaymentID (PK)  Payment\_card |

|  |
| --- |
| **CustomerDim** |
| Customer\_code (PK)  FirstName  LastName  BirthDate  MaritalStatus  Gender  Postcode  City  Income |

|  |
| --- |
| **SellingChannelsDim** |
| SellingChannel (PK)  SellingChannel\_name  CommissionRate |

Figure 3: BSORE star schema

Entity relationship diagram (ER) displays the relationships of entity set stored in a database. The main facts of ER diagram model states that they allow you to draw Database Design which is easy to use graphical tool for modelling data Widely used in Database Design. ER diagram is a graphic user interface (GUI) representation of the logical structure of a Database. It is used to help identifies the entities which exist in a system and the relationships between those entities.

The prime reasons why ER diagrams are used are:

* Helps you to define terms related to entity relationship modelling
* Provide a preview of how all your tables should connect, what fields are going to be on each table
* Helps to describe entities, attributes, relationships
* ER diagrams are translatable into relational tables which allows you to build databases quickly
* ER diagrams can be used by database designers as a blueprint for implementing data in specific software applications
* The database designer gains a better understanding of the information to be contained in the database with the help of ERP diagram
* ERD is allowed you to communicate with the logical structure of the database to users

ER Diagram models is based on three major components which are entities, attributes and relationships.

# 4.1. Implementation of Star schema

A database as a logically coherent gathering of data with some integral meaning. (Mata-Toledo and Cushman, 2000, pp.1–83). Data warehousing is the process of gathering data with an aim of storing it in a managed database in which the data are subject-oriented and integrated time variant, and non-volatile for the backing of decision-making (Inmon, 2011, pp.123–133). Data from several operations of a company are reconciled and stored in a central repository referred to as data warehouse from where analysts extract information that allows better decision making (Cho and Ngai, 2003).

# 4.1.1. Designing the Relational database

After identifying the dimensions and measures, now we have to use the star schema to relate this dimension and fact table. To achieve that, we will use MySQL Workbench software. MySQL Workbench will enable to design, model and manage the database for Blue Skyline online consumer retailer.

# 4.2. Creation of a database

First, we open MySQL Workbench and connect Database engine. We use the MySQL Workbench to create a new database called “BSORE” programmatically by opening a new SQL file clicking on the first icon in the tool bar.

A new database is created using the following statements.

A screenshot of a cell phone

Description automatically generated

Figure 4: BSORE database output

BSORE database has been created and is listed under SCHEMAS tab on the left pane as indicated above. The command “USE” sets BSORE as a default database.

The BSORE database is an empty database. The tables will be created programmatically.

To create a new table in BSORE database, we use a command “CREATE TABLE”. Can also apply a condition "IF EXISTS" or "IF NOT EXISTS". To successfully create a table, you need to define all its columns, by providing the columns’ name, type and attributes.

# 4.3. Creating dimensions

A star schema was chosen as indicated in the first part of this task. The dimensions in the schema included, Customer details, Date dimension, Selling channels, and Payment details. Each dimension table has different attributes with a primary key (PK).

# 4.3.1. CustomerDim table

A Customer details dimension table “CustomerDim” is created using the following command and will contain all customer personal details

A screenshot of a cell phone

Description automatically generated

Figure 5: CustomerDim table output

The figure above indicates CustomerDim table has been created successfully and is shown by output and under the schemas panel. The table has 9 columns describing each customer by their first name, last name, gender (M or F), birth date, marital status (single or married), postcode, city they live in and their income.

# 4.3.2. The Datedim table

The DateDim table is created using the following command,

A screenshot of a social media post

Description automatically generated

***Figure 6: DateDim table output.***

The figure above indicates that DateDim with all its attributes was successfully created and is indicated by the output and in the schemas panel. The table has 4 columns with Date Key as the primary key, the calendar year, calendar quarter and month of year indicating when orders are made for each customer.

# 4.3.3. SellingChannelDim table

The Selling channels dimension named as SellingChannelsDim table is created using the following command.

‌A screenshot of a social media post

Description automatically generated

***Figure 7: SellingChannelDim table output***

The figure above indicates that the SellingChannelDim was successfully created as shown by the output and under the schemas panel. The SellingChannelDim table has three columns, the SellingChannel, the sellingChanenel\_code and the CommissionRate given to each customer for satisfaction.

# 4.3.4. PaymentDim table.

The Payment details dimensions named as PaymentDim table is created using the following commandA screenshot of a social media post

Description automatically generated

***Figure 8: PaymentDim table output***

The figure above indicates that the PaymentDim table is created successfully and the results are shown in both the output and the schemas panel. The PaymentDim has two columns one indicating the Payment key and Payment card used by the customer when an order is made.

# 4.4. Fact table

Finally, the ORDERFACTS table is created holding all the transactional entries of previous sales with the necessary foreign key columns referred to as primary key columns in the dimension tables and measurements or facts. The following command is be used to create the ORDERFACTS table.

A screenshot of a social media post

Description automatically generated

***Figure 9: ORDERFACTS table output.***

The above figure indicates that ORDERFACTS table is created successfully as shown by the output and under the schemas panel. It contains seven columns with the first indicating four primary keys indicated as foreign keys after populating the ORDERFACTS table. The other attributes, InvoiceNumber, TotalCost and TotalRetailPrice indicates other specifics in orders.

However, we have to add a relation between fact table foreign keys to primary keys of dimensions. This is indicated by foreign key constraints.  A foreign key constraint is a database object that assists in keeping your foreign key data consistent. Creation of a foreign key constraint maintains referential integrity and used in telling MySQL to enforce certain rules such as deleting or updating over data. Certain rules such as deleting or updating over data.

# 4.5. Primary and foreign constraints

A screenshot of a social media post

Description automatically generated

***Figure 10: Primary and foreign keys constraints output.***

The above diagram indicates a successful table creation with columns of four foreign keys. For instance, the foreign key SellingChannels in the fact table references the primary key SellingChannels\_Key in the SellingChannelsDimen table. However, the above tables have no data and data will be imported in the next step.

# 6. Populating and implementing the created data warehouse

To populate and implement BSORE data warehouse, MySQL Workbench will be used. MySQL Workbench is opened and connected to database engine.

The Extract, transformation, and load (ETL) system of the Data warehouse (DW) or BI environment consists of a work area, instantiated data structures, and a set of processes (Kimball and Ross, 2013).

The ETL system is everything between the operational source systems and the DW/BI presentation area.

Extraction is the first step in the process of getting data into the data warehouse environment. After the data is extracted to the ETL system, there are several potential transformations, such as cleansing the data (correcting misspellings, resolving domain conflicts, dealing with missing elements, or parsing into standard formats), combining data from multiple sources, and de-duplicating data (Kimball and Ross, 2013)

The final step of the ETL process is the physical structuring and loading of data into the presentation area’s target dimensional models. Because the primary aim of the ETL system is to hand off the dimension and fact tables in the delivery step, these subsystems are crucial (Kimball and Ross, 2013).

These subsystems will focus on dimension table processing, such as surrogate key assignments, code lookups to provide appropriate descriptions, splitting, or combining columns to present the appropriate data values (Kimball and Ross, 2013).

The figure below indicates the steps taken to import a csv file into a table. This will be applied to all tables in BSORE database.

A screenshot of a cell phone

Description automatically generated

***Figure 11: Data import wizard steps output***

# 6.1. CustomerDim table

The file provided contained variables that required cleaning to be imported as csv file into BSORE database. The Customer Details-1 csv file, with the first two numbers representing “CC” in excel format, birthdate changed to ‘yyyy-mm-dd’, and Gender changed M for male and F for female. A sample of the cleaned csv file is shown below.

A picture containing screenshot

Description automatically generated

***Figure 12: Sample of CustomerDetails1.csv file***

To populate CustomerDim table, we perform the following steps as indicated by the outputs below. We right click on the CustomerDim table then click on select rows. Select the path to the file with the transformed data as indicated below.

A screenshot of a cell phone

Description automatically generated

***Figure 13: MySQL Import path output***

Next is the destination table is selected.

A screenshot of a cell phone

Description automatically generated

***Figure 14: MySQL destination table output***

However, each csv file has different variables which have to be configured. The encoded is already set at utf-8. The source column and destination column have to match to successfully go to the next step as indicated below.

A screenshot of a cell phone

Description automatically generated

***Figure 15: MySQL configuration settings output***

The output below indicates the import was successful.

A screenshot of a cell phone

Description automatically generated

***Figure 16: MySQL import data output.***

The above output indicates customer details was implemented and populated successfully. The same steps used as explained above will be applied for the rest of the dimension tables.

# 6.2. DateDim table

The given date.csv file contained many variables which were not required for the task. Therefore, the csv file was transformed and renamed DateIDX.csv as indicated below.

A picture containing screenshot

Description automatically generated

***Figure 17: Sample of DateID.csv file.***

Similar steps as explained above will be used to populate DateDim table.

Selection of DateIDX.csv path file to be imported

A screenshot of a cell phone

Description automatically generated

***Figure 18: MySQL selection output***

Then selection of destination table.

A screenshot of a cell phone

Description automatically generated

***Figure 19: MySQL select destination output.***

Then configuration of import settings.

A screenshot of a cell phone

Description automatically generated

***Figure 20: MySQL import settings output.***

Then, Datedim import results.

A screenshot of a social media post

Description automatically generated

***Figure 21: MySQL import results output.***

The above output indicates DateDime table is implemented and populated successfully with 7671 records imported.

# 6.3. Paymentdim table

The Payments data csv provided was renamed to PaymentDataXXX.csv which will be imported as indicated by the sample below.

A picture containing screenshot

Description automatically generated

***Figure 22: Sample of PaymentsDataXXX.csv.***

Selection of path PaymentsDataXXX.csv file to be imported

A screenshot of a social media post

Description automatically generated

***Figure 23: MySQL PaymentsDataXXX.csv output.***

PaymentsDataXXX.csv file table destination.

A screenshot of a cell phone

Description automatically generated

***Figure 24: MySQL PaymentsDataXXX.csv table destination.***

Configuration of columns in the PaymentsDataXXX.csv file into PaymentDim table.

A screenshot of a cell phone

Description automatically generated

***Figure 25: MySQL PaymentDataXXX.csv configuration output.***

PaymentsDataXXX.csv into PaymentDim

A screenshot of a cell phone

Description automatically generated

***Figure 26: PaymentsDataXXX.csv import results output***

The above output indicates that the PaymentDim is implemented and populated successfully.

# 6.4. SellingChannelDim table

The selling channels file provided was renamed to SellingChannelXX.csv as a csv file to be imported into the SellingChannelDim table. The sample is shown below.

A picture containing screenshot

Description automatically generated

***Figure 27: Sample of SellingChannelsXXX.csv file.***

SellingChannelXXX.csv selected to SellingChannelDim table as its destination.

A screenshot of a cell phone

Description automatically generated

***Figure 28: MySQL SellingChannelXXX.csv destination output.***

SellingChannelsXXX.csv import configuration settings

A screenshot of a cell phone

Description automatically generated

***Figure 29: MySQL SellingChannelsXXX.csv configuration settings output.***

SellingChannelsXXX.csv import execution logs

A screenshot of a social media post

Description automatically generated

***Figure 30: MySQL SellingChannelsXXX.csv import output.***

The above output indicates that SellingChannelDim has been implemented and populated successfully.

# 7. Implementation of ETL processes to populate a fact table

Fact tables are usually loaded from transaction tables such as order tables. The essential of loading fact tables is changing natural keys into surrogate. When loading data into a fact table, the columns on the transaction table (ORDERFACTS) become fact table measures where primary keys on the transaction table become degenerate dimension column on the fact table.

When a fact table is loaded, each of the key columns are converted from natural keys. Hence all dimension tables will be populated first for surrogate keys on the dimension tables to translate the fact table natural keys.

Similar steps while loading dimension tables will be used to load ORDERFACTS table. However, the CustomerSaleTransaction.csv file date transformed to customer\_code as indicated in the figure below.

A picture containing screenshot

Description automatically generated

***Figure 31: Sample of CustomerSaleTransactions1.csv.***

Selection of CustomerSaleTransaction1.csv path to be imported

**A screenshot of a social media post

Description automatically generated**

***Figure 32: MySQL file path output***

CustomerSaleTransaction1.csv selected ORDERFACTS table as its destination

**A screenshot of a social media post

Description automatically generated**

***Figure 33: MySQL select destination output***

Configuration of CustomerSaleTransaction1.csv import settings indication matching source and destination columns.

**A screenshot of a cell phone

Description automatically generated**

***Figure 34: MySQL CustomerSaleTransaction1.csv configuration settings output.***

CustomerSaleTransaction1.csv import logs

**A screenshot of a cell phone

Description automatically generated**

***Figure 35: MySQL import log message output.***

CustomerSaleTransaction1.csv records imported into ORDERFACTS table.

CustomerSaleTransactions1.csv import wizard

A screenshot of a cell phone

Description automatically generated

***Figure 36: MySQL import output***

The above output indicates 100 records were imported successfully in the ORDERFACTS table.

After carrying out the steps as indicated for both dimension tables and fact table, the BSORE database contains data as required.

# 8. SQL statements if dimensions and fact tables are populated correctly

To rest SQL statements testing whether the dimensions and fact tables are populated correctly, we open MySQL workbench and open BSORE database schema. Then open a new SQL script and perform SQL statement for each table as shown below.

# 8.1. Customerdim table

A screenshot of a social media post

Description automatically generated

***Figure 37: MySQL CustomerDim output.***

The output above indicate the CustomerDim table was successfully populated shown by both the result grid and action output.

# 8.2. Datedim table

A screenshot of a social media post

Description automatically generated

***Figure 38: MySQL DateDim table output.***

The output above indicate DateDim table was successfully populated by both the result grid and action output.

# 8.3. Paymentdim table

A screenshot of a social media post

Description automatically generated

***Figure 39: MySQL PaymentDim output***

The output above indicate PaymentDim dimension table was successfully populated as shown by the result grid and action output.

# 8.4. SellingChanneldim table

A screenshot of a social media post

Description automatically generated

***Figure 40: MySQL SellingChannelDim output.***

The output above indicates SellingChannelDim table was successfully populated as shown by the result grid and action output.

# 8.5. ORDERFACTS table

A screenshot of a social media post

Description automatically generated

***Figure 41: MySQL ORDERFACTS table output.***

The output above indicate ORDERFACTS table was successfully populated as shown by the result grid and action output.

# 9. The design, implementation and analysis of the OLAP Cube

Dimensions and measures are the essential mechanisms of a cube. A dimension in cube is also known as a cube dimension in reference to DateDim, CustomerDim, PaymentDim and SellingChannelDim. Measures in same instance are the facts described by the mentioned dimensions which are grouped into logical containers called measure groups. The measure clusters connect together dimensions within the form of a cube.

# 10. Critical reflection

Critical reflection can be defined as; “the means of thinking back on what has been completed and evaluating on it and learning lessons from what did or did not work” (Conway 1994). From this first part of coursework, I will be reflecting on my personal experience of working in a group of four in trying to complete part A of this coursework.

This task has introduced me to apply the physical design of the dimensional model, the Star schema of Blue sky consumer electronics retailer. To do this task, I used MySQL workbench, a great platform to write Sql commands and to create models. After an intensive research on how to use MySQL workbench, I finally got grips on how to use it. I learned to create a schema both using programming and using a diagram.

 I attempted both methods to create a new database and found out that using a diagram is easier than writing commands. However, I wanted to get more practice in writing commands which is more useful in the future. I finally used the programming method to create the BSORE database.

Working alone in attempting this task gave me challenges in understanding dimensional databases which I would not have gained if worked in a group. However, there are benefits in working in groups.

# 11. Conclusion

The task focused on designing and implementation of a Data warehouse for an online retail industry using Blue Sky consumer electronics retailer as a case study. To do the required tasks throughout, consideration of all the principles of Data warehouse and business intelligence. Performing this task has shown how to create a new database called BSORE, dimensions and fact tables. The steps taken for data in csv format for integration into a single repository known as data warehouse for Blue Sky consumer electronics so that managers can produce reports and do analysis that would enable them to give useful information for their decision-making activities.

# 12. Reference:

Cho, V. and Ngai, E.W.T. (2003). Data mining for selection of insurance sales agents. *Expert Systems*, 20(3), pp.123–132.

Connolly, T. and Beg, C. (2010). *Database systems*. 5th ed. Boston: Pearson Education Inc, pp.1177-1188.

Coronel, C., Morris, S. and Rob, P. (2013). *Database systems*. Boston: Cengage Learning, pp.559-562.

Ibm.com. (2019). *IBM Knowledge Center*. [online] Available at: https://www.ibm.com/support/knowledgecenter/en/SSHEB3\_3.4.2/com.ibm.tap.doc/abp\_performance/c\_fact\_dimension\_tables.html [Accessed 4 Nov. 2019].

Jensen, C., Bach Pedersen, T. and Thomsen, C. (2010). *Multidimensional databases and data warehousing*. [San Rafael, Calif.]: Morgan & Claypool, pp.9-95.

Kimball, R. and Ross, M. (2013). *The data warehouse toolkit*. 3rd ed. Indianapolis, Ind: Wiley, pp.1-35.

Mata-Toledo, R.A. and Cushman, P.K. (2000). *Fundamentals of relational databases*. New York: Mcgraw-Hill, pp.1–83.

‌

‌