

PROJECT REPORT

**Building and Analyzing a DW Prototype for
Metro Store in Islamabad**

Submitted to:

Sir Asif Naeem

Submitted by:

Muhammad Waleed Usman

19i-2140

INTRODUCTION

METRO Cash & Carry is a leading international company in self-service wholesale and operates more than 600 outlets in 29 countries. It is 3rd largest trader in world, 2nd largest in Europe and the Largest in Germany with a turnover of 59.9 billion in 2006. METRO is operating around 2,378 locations in 29 countries, with employees numbering over 263,000.

Since there are thousands of transactions performed by customer in a day so, there is a need for collecting raw data coming from different customers, products, suppliers. Storing the data and make it meaning full. Here there is a need of Data ware house. A data warehousing is defined as a technique for collecting and managing data from varied sources to provide meaningful business insights. It is a blend of technologies and components which aids the strategic use of data. It is electronic storage of a large amount of information by a business which is designed for query and analysis instead of transaction processing. It is a process of transforming data into information and making it available to users in a timely manner to make a difference. Also metro is the biggest superstore so there will be the continuous transaction 24 a day, so to store data continuously we need to make some queries regarding that.

Tasks done in the project:

I have completed the following tasks in this project:

1. Identification of appropriate dimension tables and fact table. The appropriate primary and foreign keys in the dimensions and fact table. The following star schema is made in the project.

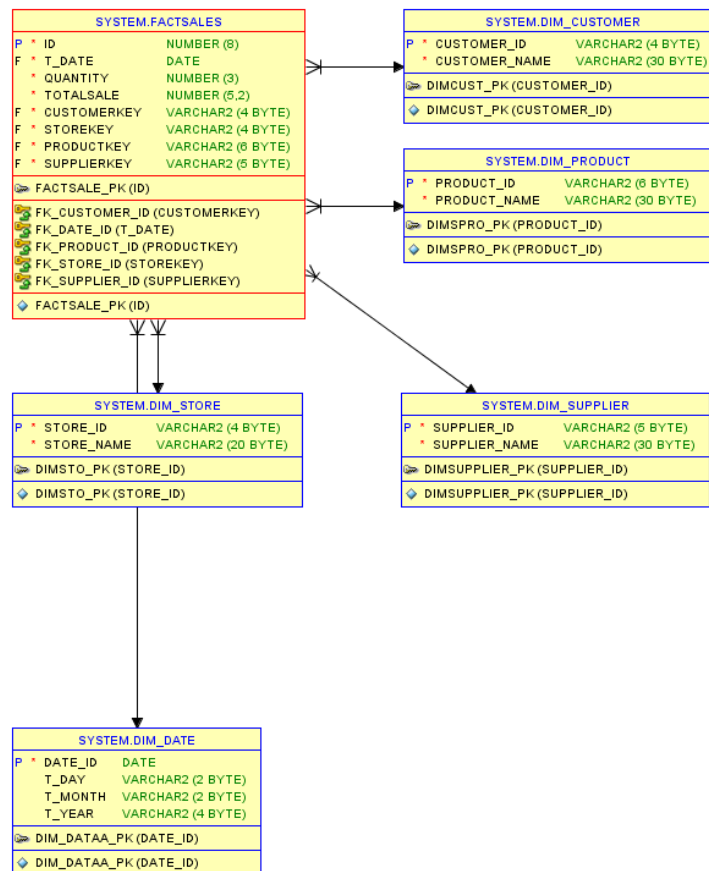
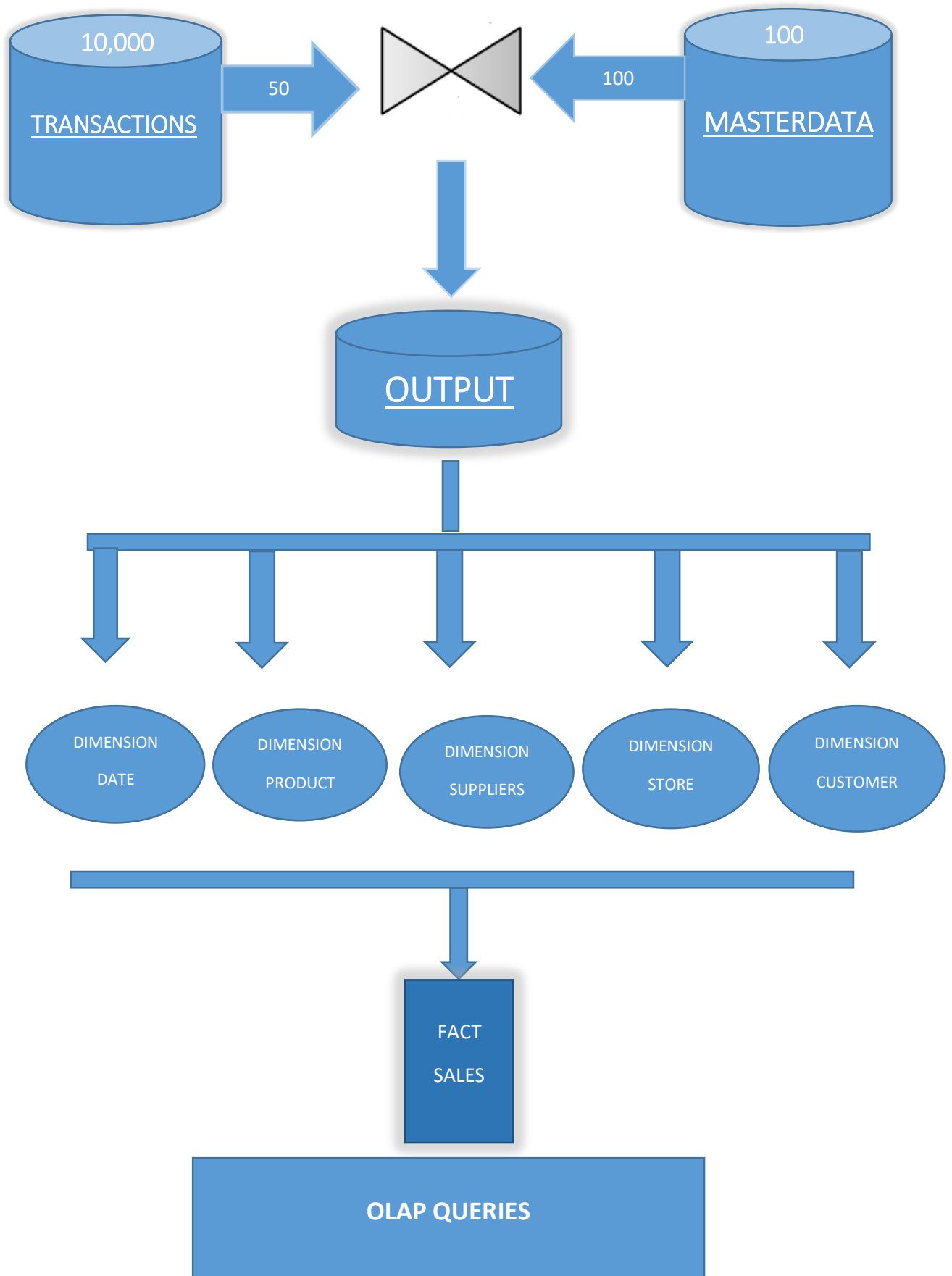
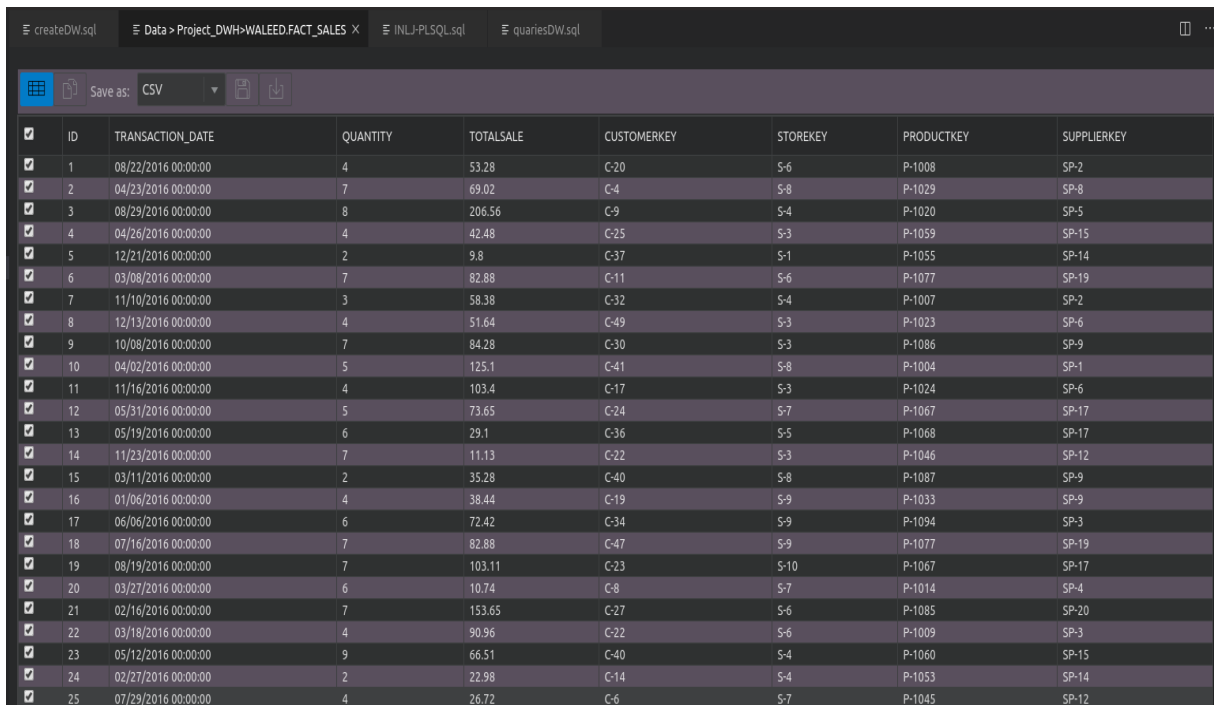


Figure 1: STAR SCHEMA

MODEL OF THE PROJECT



- Implementation of the INLJ algorithm and loading of transactional data into DW after joining it with MD, This is done using the cursors and traversing it through a tuple of range 50 and comparing it with the list of products present in the MASTERDATA. Using nested loop to join the two tables by comparing the PRODUCT_ID using the index of the cursor. After the execution, the FACTSALES and other dimensions are populated. As shown below:



The screenshot shows a SQL IDE window with a table of fact sales data. The table has 9 columns: ID, TRANSACTION_DATE, QUANTITY, TOTALSALE, CUSTOMERKEY, STOREKEY, PRODUCTKEY, and SUPPLIERKEY. The data is displayed in a grid with alternating light and dark rows. Each row is preceded by a checkbox, all of which are checked. The table contains 25 rows of data, with the last row having an ID of 25.

ID	TRANSACTION_DATE	QUANTITY	TOTALSALE	CUSTOMERKEY	STOREKEY	PRODUCTKEY	SUPPLIERKEY
1	08/22/2016 00:00:00	4	53.28	C-20	S-6	P-1008	SP-2
2	04/23/2016 00:00:00	7	69.02	C-4	S-8	P-1029	SP-8
3	08/29/2016 00:00:00	8	206.56	C-9	S-4	P-1020	SP-5
4	04/26/2016 00:00:00	4	42.48	C-25	S-3	P-1059	SP-15
5	12/21/2016 00:00:00	2	9.8	C-37	S-1	P-1055	SP-14
6	03/08/2016 00:00:00	7	82.88	C-11	S-6	P-1077	SP-19
7	11/10/2016 00:00:00	3	58.38	C-32	S-4	P-1007	SP-2
8	12/13/2016 00:00:00	4	51.64	C-49	S-3	P-1023	SP-6
9	10/08/2016 00:00:00	7	84.28	C-30	S-3	P-1086	SP-9
10	04/02/2016 00:00:00	5	125.1	C-41	S-8	P-1004	SP-1
11	11/16/2016 00:00:00	4	103.4	C-17	S-3	P-1024	SP-6
12	05/31/2016 00:00:00	5	73.65	C-24	S-7	P-1067	SP-17
13	05/19/2016 00:00:00	6	29.1	C-36	S-5	P-1068	SP-17
14	11/23/2016 00:00:00	7	11.13	C-22	S-3	P-1046	SP-12
15	03/11/2016 00:00:00	2	35.28	C-40	S-8	P-1087	SP-9
16	01/06/2016 00:00:00	4	38.44	C-19	S-9	P-1033	SP-9
17	06/06/2016 00:00:00	6	72.42	C-34	S-9	P-1094	SP-3
18	07/16/2016 00:00:00	7	82.88	C-47	S-9	P-1077	SP-19
19	08/19/2016 00:00:00	7	103.11	C-23	S-10	P-1067	SP-17
20	03/27/2016 00:00:00	6	10.74	C-8	S-7	P-1014	SP-4
21	02/16/2016 00:00:00	7	153.65	C-27	S-6	P-1085	SP-20
22	03/18/2016 00:00:00	4	90.96	C-22	S-6	P-1009	SP-3
23	05/12/2016 00:00:00	9	66.51	C-40	S-4	P-1060	SP-15
24	02/27/2016 00:00:00	2	22.98	C-14	S-4	P-1053	SP-14
25	07/29/2016 00:00:00	4	26.72	C-6	S-7	P-1045	SP-12

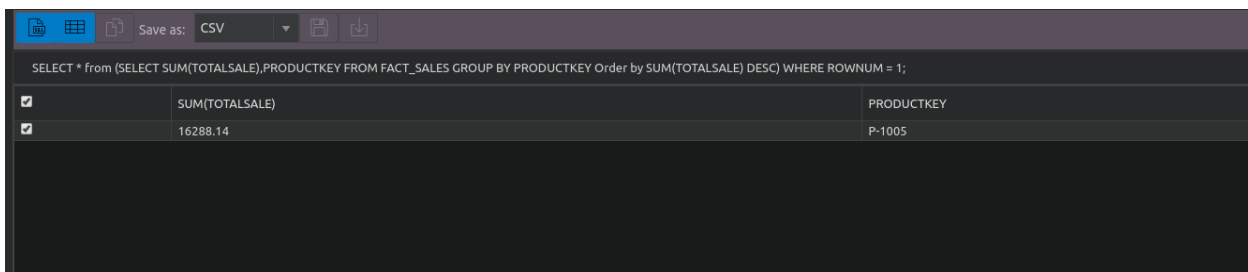
Figure 2: Fact sales data populated

3. Applying OLAP queries:

For analysis purpose we apply OLAP queries to calculate or measure sales of the DW.

Q1 Which product produced highest sales in the whole year?

In this query we have to calculate the highest sale in the whole year, for that purpose we just select sum of total sales for each product in the year and then writing that in descending order then I just put up the last row in the sum(TOTALSALE) which is in fact the highest sale.

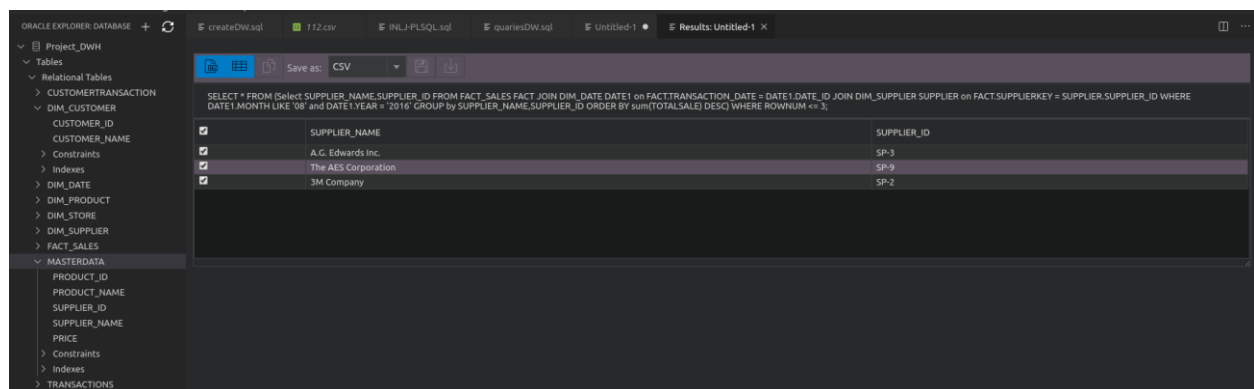


SUM(TOTALSALE)	PRODUCTKEY
16288.14	P-1005

Figure 3: Q1 QUERY OUTPUT

Q2 Determine the top 3 supplier names in Aug 2016 in terms of total sales.

In this query we have to determine the top 3 suppliers names in the month August 2016, for that purpose we just select sum of total sales for each supplier in the Aug 2016 and then writing that in descending order then I just put up the last 3 row from the sum(TOTALSALE) which are in fact the top three suppliers.

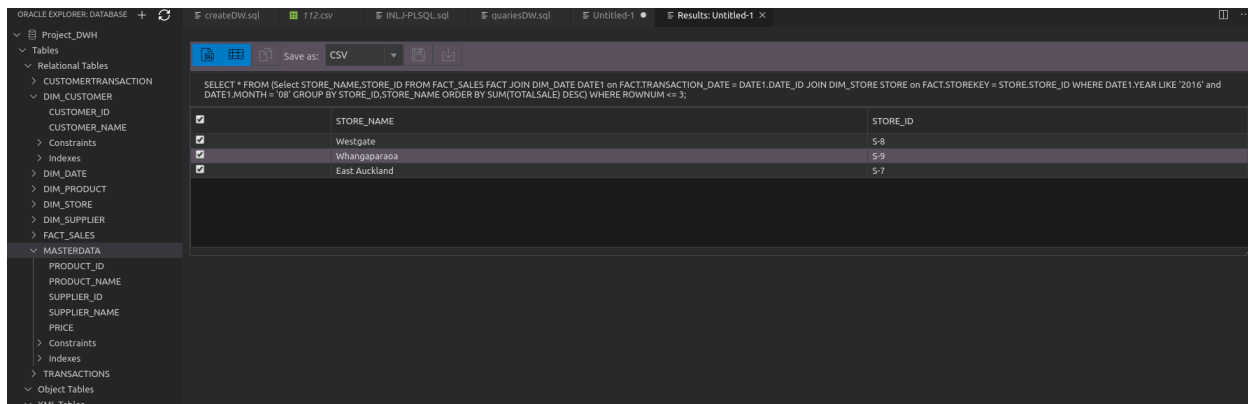


SUPPLIER_NAME	SUPPLIER_ID
A.G. Edwards Inc.	SP-3
The AES Corporation	SP-9
3M Company	SP-2

Figure 4: Q2 QUERY OUTPUT

Q3 Determine the top 3 store names in Aug 2016 in terms of total sales.

In this query we have to determine the top 3 store names in the month August 2016, for that purpose we just select sum of total sales for each store in the Aug 2016 and then writing that in descending order then I just put up the last 3 row from the sum(TOTALSALE) which are in fact the top three stores.



The screenshot shows the Oracle SQL Developer interface. The left pane displays the database schema for 'Project_DWH', including tables like CUSTOMERTRANSACTION, DIM_CUSTOMER, DIM_DATE, DIM_PRODUCT, DIM_STORE, DIM_SUPPLIER, FACT_SALES, and MASTERDATA. The central pane contains a SQL query that selects the top 3 stores by total sales for August 2016. The right pane shows the query results as a table with two columns: STORE_NAME and STORE_ID. The results list Westgate (5-8), Whangaparaoa (5-9), and East Auckland (5-7) as the top three stores.

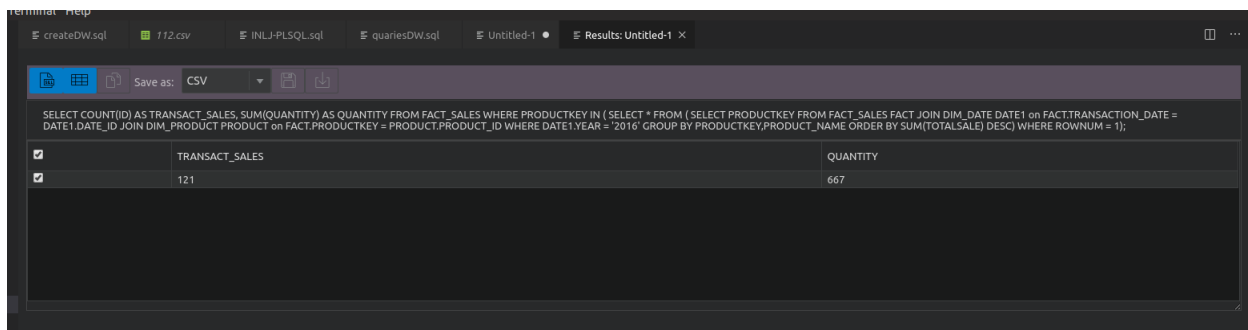
```
SELECT * FROM (select STORE_NAME,STORE_ID FROM FACT_SALES FACT JOIN DIM_DATE DATE1 on FACT.TRANSACTION_DATE = DATE1.DATE_ID JOIN DIM_STORE STORE on FACT.STOREKEY = STORE.STORE_ID WHERE DATE1.YEAR LIKE '2016' and DATE1.MONTH = '08' GROUP BY STORE_ID,STORE_NAME ORDER BY SUM(TOTALSALE) DESC) WHERE ROWNUM <= 3;
```

STORE_NAME	STORE_ID
Westgate	5-8
Whangaparaoa	5-9
East Auckland	5-7

Figure 5: Q3 QUERY OUTPUT

Q4 How many sales transactions were there for the product that generated maximum sales revenue in 2016? Also present the product quantity sold.

In this query we have to determine the product that generated maximum sales in 2016, for that purpose we just select sum of total sales for each product for year 2016 and then finding the total transactions produced by each product, the top will be the winner.



The screenshot shows a SQL query results window with the following SQL query:

```
SELECT COUNT(ID) AS TRANSACT_SALES, SUM(QUANTITY) AS QUANTITY FROM FACT_SALES WHERE PRODUCTKEY IN ( SELECT * FROM ( SELECT PRODUCTKEY FROM FACT_SALES FACT JOIN DIM_DATE DATE1 ON FACT.TRANSACTION_DATE = DATE1.DATE_ID JOIN DIM_PRODUCT PRODUCT ON FACT.PRODUCTKEY = PRODUCT.PRODUCT_ID WHERE DATE1.YEAR = '2016' GROUP BY PRODUCTKEY,PRODUCT_NAME ORDER BY SUM(TOTALSALE) DESC) WHERE ROWNUM = 1);
```

The results table has two columns: TRANSACT_SALES and QUANTITY. The first row shows the results for the product with the maximum sales in 2016.

TRANSACT_SALES	QUANTITY
121	667

Figure 6: Q4 QUERY OUTPUT

Q5. Present the quarterly sales analysis for all products using drill down query concepts, result is:

In this query we have to determine the product that generated maximum sales in each quarter of the year, for that purpose we just select sum of total sales for each 4 months for year 2016. for first quarter the months will be 1 to 3, for second that will be 4 to 6 then 7 to 9 then 10 to 12.

The screenshot shows the Oracle SQL Developer interface with three SQL queries and their results. The left pane shows the database schema with tables like CUSTOMER_TRANSACTION, DIM_CUSTOMER, DIM_DATE, DIM_PRODUCT, DIM_STORE, DIM_SUPPLIER, FACT_SALES, MASTERDATA, and PRODUCTKEY. The main pane displays three queries and their results.

Query 1: Q1 2016

```
SELECT * FROM (SELECT PRODUCT_NAME, SUM(TOTALSALE) AS Q1_2016 FROM FACT_SALES FACT JOIN DIM_DATE DATE1 ON FACT.TRANSACTION_DATE = DATE1.DATE_ID JOIN DIM_PRODUCT PRODUCT ON FACT.PRODUCTKEY = PRODUCT.PRODUCT_ID WHERE DATE1.MONTH BETWEEN '01' AND '03' GROUP BY PRODUCT_NAME);
```

PRODUCT_NAME	Q1_2016
Soups	4615.84
Hummus	971.56
Jam / Jelly / Preserves	245.1
Relish	1955.03
Paprika	2038.4
Ice cream / Sorbet	2799.5
Pickles	250.2
Mushrooms	2091.24

Query 2: Q2 2016

```
SELECT * FROM (SELECT PRODUCT_NAME, SUM(TOTALSALE) AS Q2_2016 FROM FACT_SALES FACT JOIN DIM_DATE DATE1 ON FACT.TRANSACTION_DATE = DATE1.DATE_ID JOIN DIM_PRODUCT PRODUCT ON FACT.PRODUCTKEY = PRODUCT.PRODUCT_ID WHERE DATE1.MONTH BETWEEN '04' AND '06' GROUP BY PRODUCT_NAME);
```

PRODUCT_NAME	Q2_2016
Jam / Jelly / Preserves	337.55
Paprika	1638
Chip dip	1215.02
Asparagus	840.75
Pickles	247.42
Hummus	1216.72
BBQ sauce	1464.32
Soups	2434.32

Query 3: Q3 2016

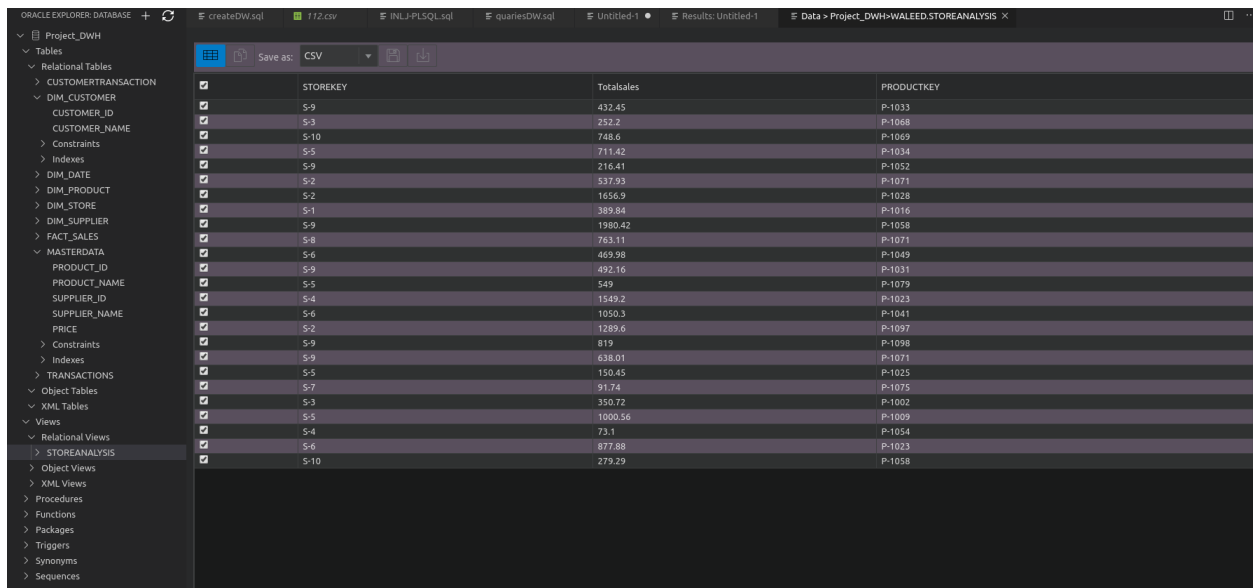
```
SELECT * FROM (SELECT PRODUCT_NAME, SUM(TOTALSALE) AS Q3_2016 FROM FACT_SALES FACT JOIN DIM_DATE DATE1 ON FACT.TRANSACTION_DATE = DATE1.DATE_ID JOIN DIM_PRODUCT PRODUCT ON FACT.PRODUCTKEY = PRODUCT.PRODUCT_ID WHERE DATE1.MONTH BETWEEN '07' AND '09' GROUP BY PRODUCT_NAME);
```

PRODUCT_NAME	Q3_2016
Mushrooms	2131.2
Grapefruit	3614.8
Ice cream / Sorbet	3461.2
Relish	4062.4
Jam / Jelly / Preserves	298.85
Hummus	1625.32
BBQ sauce	1228.8

Figure 7: Q5 QUERY OUTPUT

Q6.Create a materialized view with name “STOREANALYSIS_MV” that presents the product-wise sales analysis for each store.

In this query we have to determine the product wise sale of each shop that, for that purpose we just select sum of total sales for each product for each distinct shop.



STOREKEY	Totalsales	PRODUCTKEY
S-9	432.45	P-1033
S-3	252.2	P-1068
S-10	748.6	P-1069
S-5	711.42	P-1034
S-9	216.41	P-1052
S-2	537.93	P-1071
S-2	1656.9	P-1028
S-1	389.84	P-1016
S-9	1980.42	P-1058
S-8	763.11	P-1071
S-6	469.98	P-1049
S-9	492.16	P-1031
S-5	549	P-1079
S-4	1549.2	P-1023
S-6	1050.3	P-1041
S-2	1289.6	P-1097
S-9	819	P-1098
S-9	638.01	P-1071
S-5	150.45	P-1025
S-7	91.74	P-1075
S-3	350.72	P-1002
S-5	1000.56	P-1009
S-4	73.1	P-1054
S-6	877.88	P-1033
S-10	279.29	P-1058

Figure 8: Q6 QUERY OUTPUT

What I have learned from the project:

As a student I have learned a lot of things in this project. Learned how to tackle the problems while installing the software. Before this project I was blank regarding the SQL development. I didn't know much about the SQL language and databases. But using the knowledge I get from the DWH classes and then implementing that knowledge In practical is the main thing. I learned from the problems I faced while doing this project. I have practiced writing queries for different problem statement. Also I have learned what are the benefits of the data warehouse. In the start I was not able to install the oracle software on my computer, but after practicing it, I learned many things about the oracle database. Oracle was the software I have never used before. In the beginning I faced a lot of problems during installation, after doing some research I figured out the problems. then the next difficult phase was the queries, it took literally a lot of time in understanding and implementing it, but at the end it went good.

Thank you.