

Data Warehousing

(AIK232)

LAB 3

REPORT

Chinonso Williams Ubani

# Introduction

In this project, I have used a dataset from a Grocery business that I used during Lab 1 for the Data Warehousing course. Furthermore, we have studied the nature and character of the data if it can fulfill all the requirements of lab-3 tasks. They identified the business process and the grains.

# Aim / Tasks

The main aim of this project is to design a simple star schema for the grocery business we have chosen. To design a star schema the following four main steps were considered:

* Identifying the business process
* Identifying the grains
* Choosing the dimensions
* Choosing the measures

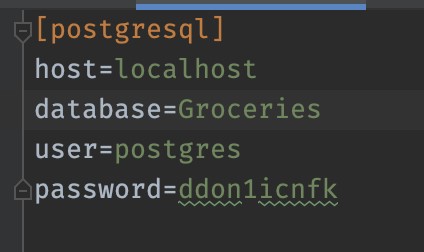
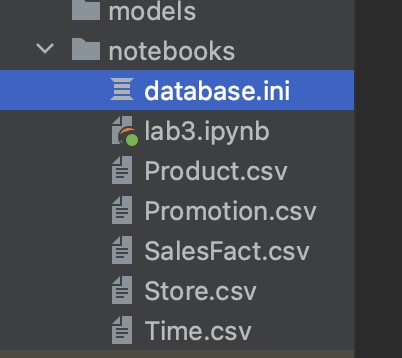
Furthermore, several tasks will be implemented at a later stage. All tasks will be presented in the implementation part of this report.

# Methodology

To model the database for this Grocery Store schema, we extensively utilized Python and PostgreSQL. We used python to write the database on PostgreSQL. In python, several libraries were employed depending on the usage. To mention some:

* psycopg2
* datetime
* Pandas

Setting the Environment for this project



**To connect to the database:**

1. Install PostgreSQL locally
2. Create a database called Groceries (you can rename the database anything)
3. Update the database name in the database.ini to match the database created locally
4. Update the user and password to match the local machine credentials
5. Run the Python queries/codes for the G & VG requirements

# Implementation

In this section, I will provide the solution to each task and explain it in detail. To start with I designed the data model (ER Diagram) as a star schema, and we used StarUML to create it. I have defined the relationships by putting constraints on the primary key and foreign keys. Mentioned each different data type (varchar, integer, bigint, timestamp) and with these criteria, I have created the tables in PostgreSQL.

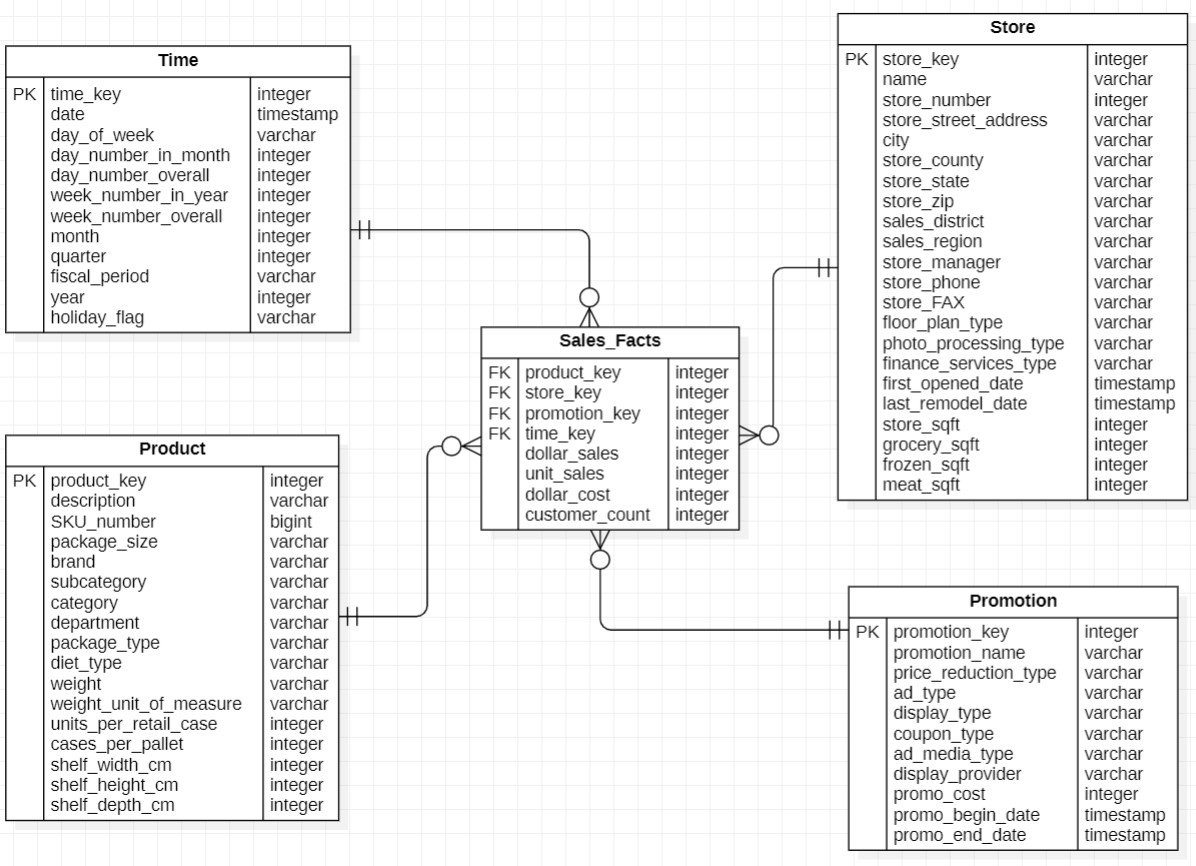


Figure. 1 ER diagram of the Grocery Business.

I have exported the Data from the Access file to separate CSVs for each entity as product, promotion, time, store, and sales facts. These files contain the product, promotion, time, store, and sales fact table for the Grocery business which we have chosen. The product file contains all the product detailed descriptions used in the grocery business; we also included the product key as the primary key for the product table. The promotion table contains all the names of promotions and types used for the business, it also includes the beginning date, end date, display type, and so on. Here the promotion key is the primary key for promotion. The time file contains the data attributes for all the dates in detail in which all products were sold in the business, this includes the day of the week, the day number in the month, the year, and so on; here the time key is used as the primary key for this table. The store file contains all the necessary details for each store location since as a grocery business it has different stores in vast areas of the country, the store is used as the primary key. Lastly, the Sales facts contain all the quantitative transactional data for the business, and all the different dimension tables have been connected as a one-to-many (1:M) relationship with the fact table.

# Tasks

***Question 0: Which transformation needs to be applied to the DW, if the company is interested in knowing the effect of promotions on sales?***

To give an answer to this data the company needs to take into consideration of the following useful variables:

* Promotion name
* Year
* Quarters
* Sum of sales
* Promotion costs
* Product name as the description

So, I created a new connection between the sales and promotion tables in SQL and established the connection with the database to MS Excel. Once the connection is established, we created a pivot table in an excel file for further explorations.

I analyzed the dataset and saw that there was not a significant increase in sales across the products for this store. Also, there was not enough data to conclude the effect of promotions over sales afterward as the dataset only contain the data from Quarter 4 (October, November & December) during the years 1994 and 1995.



Table. 1 Promotion table

When I run the below SQL query on our database I have the following output.

|  |
| --- |
| **SELECT** product.description **as** product\_name, **SUM**(sales\_facts.dollar\_sales) **as** sales,  **SUM**(sales\_facts.dollar\_cost) **as** **cost**, (**SUM**(sales\_facts.dollar\_sales) - **SUM**( sales\_facts.dollar\_cost ))**as** profit, **SUM**(sales\_facts.unit\_sales) **as** units,  **COALESCE**(promotion.promotion\_name, 'No promotion') **as** promotion\_name,  **COALESCE**(**SUM**(promotion.promo\_cost), 0) **as** promotion\_cost, t.**year**, t.**month**, t.quarter  **FROM** sales\_facts  **JOIN** promotion **ON** promotion.promotion\_key = sales\_facts.promotion\_key  **JOIN** product **ON** product.product\_key = sales\_facts.product\_key  **JOIN** **time** t **ON** t.time\_key = sales\_facts.time\_key  **WHERE** product.description = 'Strong Cola'  **GROUP** **BY** product\_name,promotion.promotion\_name,t.**year**, t.**month**, t.quarter **ORDER** **BY** profit **DESC**, t.**month** **desc**, t.**year** **desc;** |

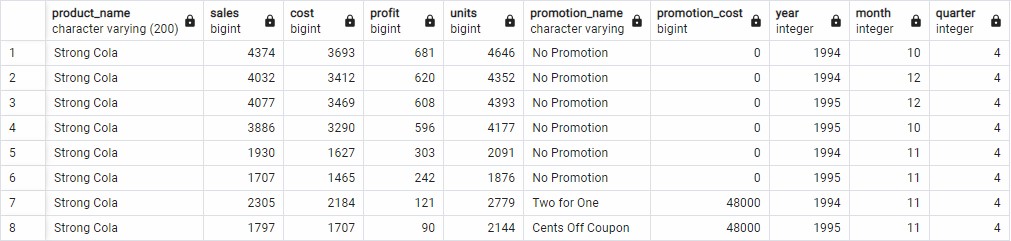


Table. 2 The query result of aggregating by the promotion for one (strong cola) product.

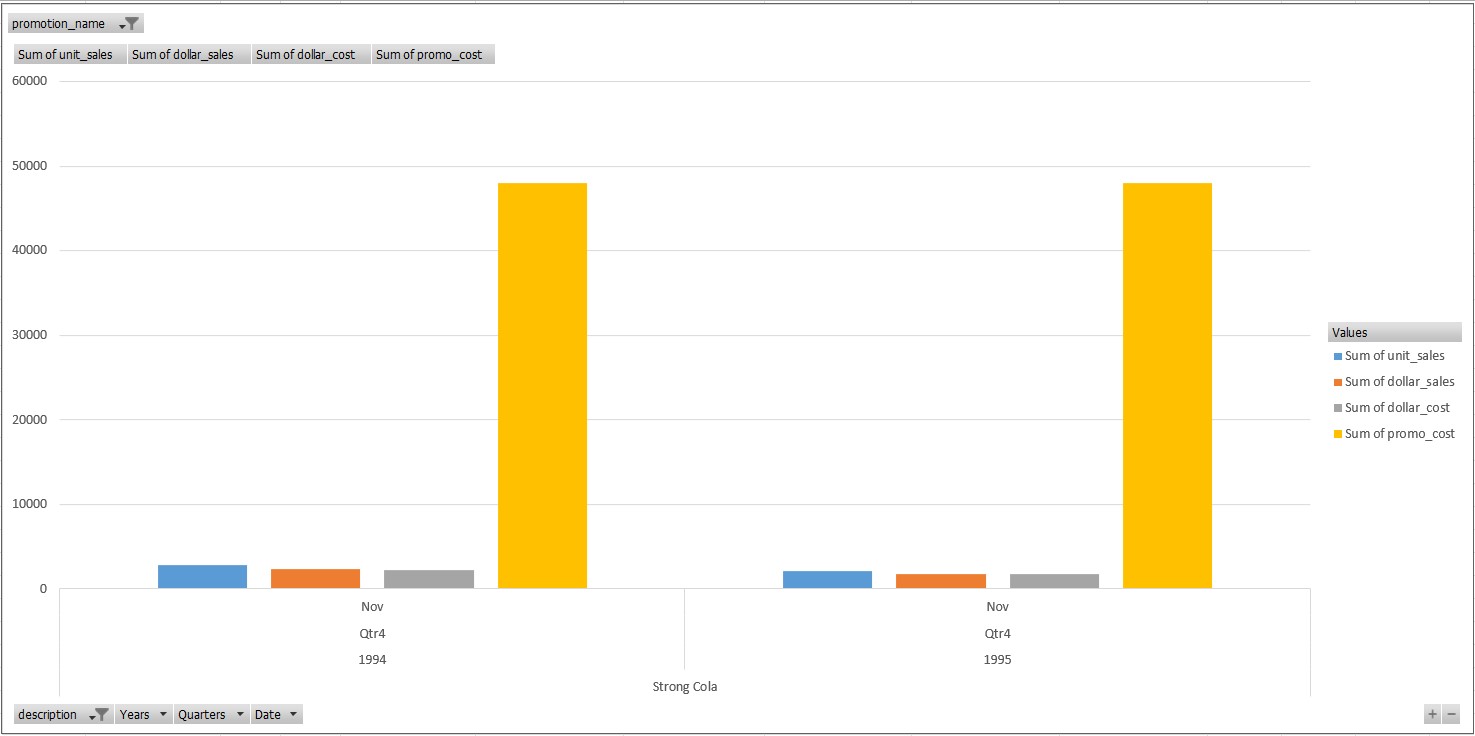


Figure 2. visualization for Strong Cola, based on the sales, cost, and promotion cost.

If we see in the highlighted section for both tables 1 & 2, we get the doubt of how one particular product has two promotions costing 48000 individually. This does not make sense as the total cost for all the promotions cost 29,100 in total to be precise, in Table. 1. And this doesn’t only happen to only one product but to all of the products.

|  |
| --- |
| **SELECT** product.description **as** product\_name , **SUM**(sales\_facts.dollar\_sales) **as** sales, **SUM**(sales\_facts.dollar\_cost) **as** **cost**,(**SUM**(sales\_facts.dollar\_sales) - **SUM**( sales\_facts.dollar\_cost ))**as** profit, **SUM**(sales\_facts.unit\_sales) **as** units,  **COALESCE**(promotion.promotion\_name, 'No promotion') **as** promotion\_name,  **COALESCE**(**SUM**(promotion.promo\_cost), 0) **as** promotion\_cost ,t.**year**, t.**month**, t.quarter  **FROM** sales\_facts  **FULL** **OUTER** **JOIN** promotion **ON** promotion.promotion\_key = sales\_facts.promotion\_key  **FULL** **OUTER** **JOIN** product **ON** product.product\_key = sales\_facts.product\_key  **FULL** **OUTER** **JOIN** **time** t **ON** t.time\_key = sales\_facts.time\_key  **GROUP** **BY** product\_name,promotion.promotion\_name, t.**year**, t.**month**, t.quarter **ORDER** **BY** profit **DESC;** |
|  |

Table. 3 aggregated values for the products with no promotion.

So, depending on these above findings, we can conclude that the way of storing the values of promotion cost has caused such misinformation. In order to fix this the database needs a transformation by changing the method for storing the promotion cost differently. One possible way could be to split the promotion cost per unit so that when we aggregate the distribution of promotion cost, the sum would be 29100 instead of some wrong calculation. And this will help the company analyze the effect of promotion on sales.

***Question 1: How many sales did the company have?***

**SELECT** **SUM**(unit\_sales) **AS** unit\_sales

**FROM** sales\_facts;



***Question 2: How many products did we sell in a store last month?***

|  |  |
| --- | --- |
| **SELECT** **SUM**(unit\_sales) **AS** unit\_sales **FROM** sales\_facts  **WHERE** time\_key **in** (**SELECT** time\_key **FROM** **time** **WHERE** **date** **BETWEEN** '1995-12-01' **AND** '1995-12-31') **AND** store\_key **in** (**SELECT** store\_key **FROM** **store** **WHERE** store\_number = '10'); | |
|  |  |

***Question 3: Which one of our stores has the highest amount of sales?***

|  |  |
| --- | --- |
| **SELECT** store\_number, total\_sales  **FROM** (  **SELECT** **s**.store\_number, **SUM**(sf.dollar\_sales) **as** total\_sales  **FROM** sales\_facts sf  **JOIN** **store** **s** **ON** **s**.store\_key = sf.store\_key  **GROUP** **BY** **s**.store\_key  **ORDER** **BY** total\_sales **DESC**  **LIMIT** 1  ) t; | |
|  |  |

***Question 4: Which products are the most lucrative?***

|  |  |
| --- | --- |
| **SELECT** product.description **as** product\_name , **SUM**(sales\_facts.dollar\_sales) -  **SUM**(sales\_facts.dollar\_cost) **as** profit  **FROM** sales\_facts  **JOIN** product **ON** product.product\_key = sales\_facts.product\_key  **GROUP** **BY** product.description  **ORDER** **BY** profit **desc**; | |
|  |  |

***Question 5:*** *What was the most lucrative day, month, or year?*

*Year*:

|  |  |
| --- | --- |
| **SELECT** **time**.**year**, **SUM**(sales\_facts.dollar\_sales) - **SUM**(sales\_facts.dollar\_cost) **as** profit  **FROM** sales\_facts  **JOIN** **time** **ON** **time**.time\_key = sales\_facts.time\_key  **GROUP** **BY** **time**.**year**  **ORDER** **BY** profit **desc**; | |
|  |  |

*Day of the week:*

|  |  |
| --- | --- |
| **SELECT** **time**.day\_of\_week, **SUM**(sales\_facts.dollar\_sales) - **SUM**(sales\_facts.dollar\_cost) **as** profit  **FROM** sales\_facts  **JOIN** **time** **ON** **time**.time\_key = sales\_facts.time\_key  **GROUP** **BY** **time**.day\_of\_week  **ORDER** **BY** profit **desc**; | |
|  |  |

*Month:*

|  |  |
| --- | --- |
| **SELECT** **time**.**month**, **SUM**(sales\_facts.dollar\_sales) - **SUM**(sales\_facts.dollar\_cost) **as** profit  **FROM** sales\_facts  **JOIN** **time** **ON** **time**.time\_key = sales\_facts.time\_key  **GROUP** **BY** **time**.**month**  **ORDER** **BY** profit **desc**; | |
|  |  |