Mid-term Project Report

Retail Data Warehouse-KTSY Store

Prepared by: Group 2

Ketaki Joshi

Tanvi Sandeep Bhagat

Swetha Daddala

Yuanying Li

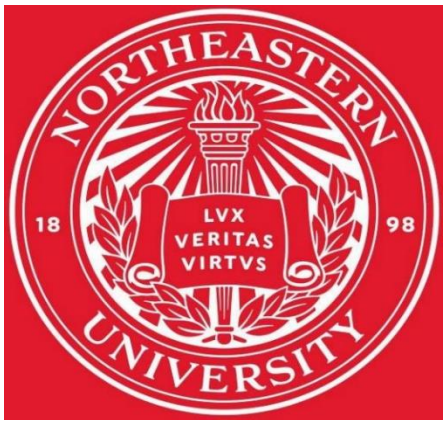
Course: ALY 6030

Under the Guidance Of:

Prof. Fidel Rodriguez

Northeastern University

Oct 12, 2020



Contents

1. Introduction

Identify a publicly traded business

15 years back, Information Technology gave a gift to this world in the form of E-commerce. Since then, every small and big business has used it to improve its outreach, customer count, sales, profit, and every possible aspect. But this was not sufficient. As data grew from MB to GB to TB, smart businesses felt the need to store the data efficiently and to utilize it for improving the various aspects of the business.

Digital transformation is disrupting every industry and creating massive new opportunities, with the retail industry being no exception where the customers and products are the key aspects. Which products are in demand or which customer would be interested in buying what products can be understood with the help of the technology? Data Warehouse plays a crucial role in this.

As much has been written about the importance of grocery stores in today's world, the "Grocery-Store Management" caught our attention from among several publicly-traded businesses available.

Irrespective of whether the store is a small house-hold business or a franchise of a larger firm, heading a grocery store successfully requires substantial responsibility. The store managers should make sure that the store functions smoothly, the items are not overpriced or underpriced, and that customers are satisfied with the quality of the products. Launching new stores in new locations is also challenging as it involves careful consideration of the customer's needs. There has been significant analysis that highlights the potential troubles that store managers face inefficient store management and development. Several inter-related complications must be overcome to attract a new grocery store and to have it be successful.

About KTSY Store: We have built a grocery store management data warehouse for a fictional "KTSY Store". Our design model helps the store managers effectively perform their day-to-day tasks by having a primary focus on the following business functions:

1. Customer Support

2. Sales Analysis

3. Vendor Management

Our Data Warehouse design methodology: The idea behind the design of our data warehouse is to create an optimized schema to promote efficient decision support processing. We have used "Star Schema" also referred as the "Dimensional Model" for our database design. This model of ours satisfies the following requirements of a dimensional model:

1.The data obtained for our design is accurate and scrubbed.

2.The schema created performs quick query processing.

3.The data schema is simple and easy to understand.

We have ensured to incorporate the following key steps into our design methodology:

1. Choosing the data mart.

2. Choosing the grain of the fact table.

3. Choosing the dimensions appropriate for the grain

4. Choosing the facts

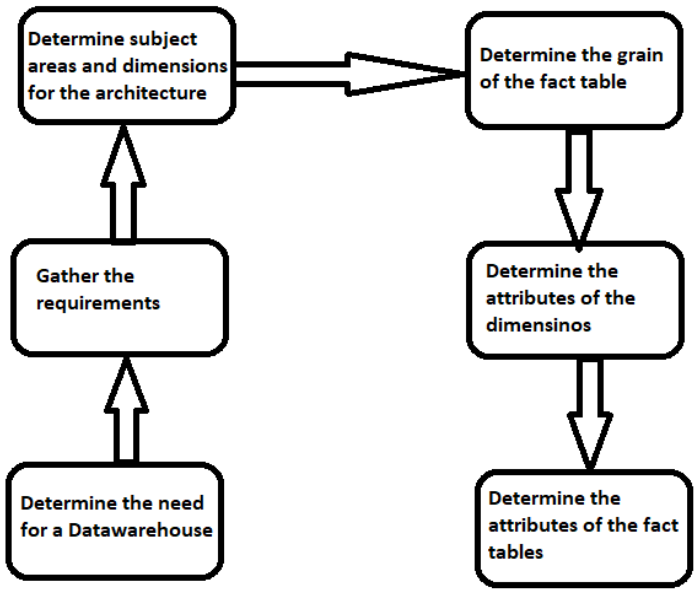


Figure #1: Dimensional Modeling process for KTSY Data Warehouse

2. Requirement Analysis for Data Warehouses in KTSY_Store

Our way of gathering the requirements for KTSY store is based on an extensive collection of OLAP (Online Analytical Processing) queries and replicated business scenarios.

a) Primary Requirements: Identifying and defining the right set of requirements is an extremely important step in designing a data warehouse especially in the grocery management business.

We have captured certain unique information related to our KTSY_Store. In order to do this, we focused on the data available from transactional systems and navigation data. We stored this data in a schematic structure of “Star schema.”

From our analysis on the requirements of KTSY_Store, we simulated a few business scenarios and developed a few OLAP queries, we then assigned these queries to different categories upon which the data mart is built. As an example, if we consider the queries clustered underneath the Vendor Management business function, the store management has a chance to notice the buying patterns of customers and take the right steps in increasing the sales.

As the data warehouse designers, we noticed emerging patterns as we analyzed the business questions. From the business perspective, we assumed that management wants to know the sales according to ‘products, categories, customers, time, and suppliers”. The management may also like to look at the employee's information department wise from the tables Employees and Departments. We have analyzed that each of the aforesaid tables becomes a dimension table. We have stored the important values associated with product sales in the “Orders” fact table. We have combined the aforesaid dimension tables with the fact tables, to form a sales subject area for our data warehouse.

i) Our Approach:

The Process: For our requirement analysis for data warehousing in our KTSY project, we underwent a series of discussions and plans to develop the OLAP (online analytical processing) queries. We inspected the actual e-commerce sites to thoroughly understand the requirements, simulated many business scenarios to understand the workflow, and developed certain OLAP queries. After we replicated the business questions and OLAP queries, we assigned them to various categories. The fact table in our design model consists of the important values the business wants to analyze related to their product sales. After we combined the aforesaid dimension tables with the newly designed fact table, we were able to build a Sales subject area for the data warehouse.

The Grain of the Fact Table: The Fact tables usually represent the additive values that act as independent variables by which dimensional attributes are analyzed. Fact tables are normally defined by their grain. The grain of a fact table indicates the most atomic level by which the facts are defined. In our KTSY store, the grain of orders fact table can be indicated as "sales volume by each month per product". Each record in this fact table is therefore uniquely defined by an “order_id, customer_id, employee_id, time_key”. Other dimensions might be members of this fact table (such as order_details/customers/products/employees) however, these add nothing to the uniqueness of the fact records. For the sales subject area in our store, we desired to capture the monthly sales and product transactions by the category. From the OLAP queries we simulated during the requirement analysis, we understood that we should pay a keen level of

attention to the details. As an example, one OLAP query we have included is “What brand products do customers most often buy?”. In order to answer the mentioned question, we realized that we should have access to all of the categories and products on all sales because at this level we can associate a specific product with a specific customer.

Slowly Changing Dimensions: Dimensional attributes are often dynamic and will rarely remain static. Likewise, during our data warehouse design, we often thought about the slowly changing dimension attributes. We encountered certain complexity issues that are associated with different dimension tables. For example, the store might introduce new products and might rule out the existing ones. An employee might change his department, or an employee might decide to quit his work at the store. Also, if the store reassigns sales related tasks to new sales representatives, our data warehouse design should be able to track the changes robustly. If a customer's name changes, we should be able to record the change. We designed our database to handle the mentioned changes accurately and efficiently. We have decided to include “**Type 2**” changes in our data warehouse, so we have decided to include the following columns to our table whenever there is an update:

Surrogate Key – the original ID will no longer be sufficient to identify the specific record we require; we therefore need to create a new ID that the fact records can join.

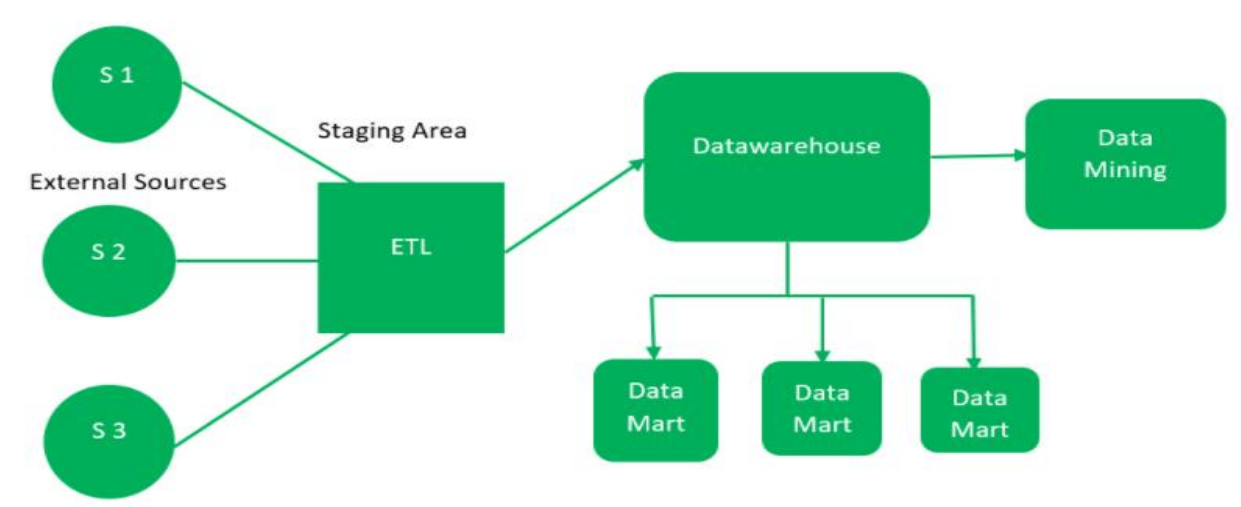
Current Flag – A quick method of returning only the current version of each record

Start Date – The date from which the specific historical version is active

End Date – The date to which the specific historical version record is active with the elements in place.

ii) Architectural Patterns: A data-warehouse is a heterogeneous collection of different data sources organized under a unified schema. From among Top-Down approach and Bottom-Down approach, we used ***a top-down approach*** in our warehouse architecture. Our data warehouse bus architecture is a matrix that shows dimensions in columns and data marts as rows.

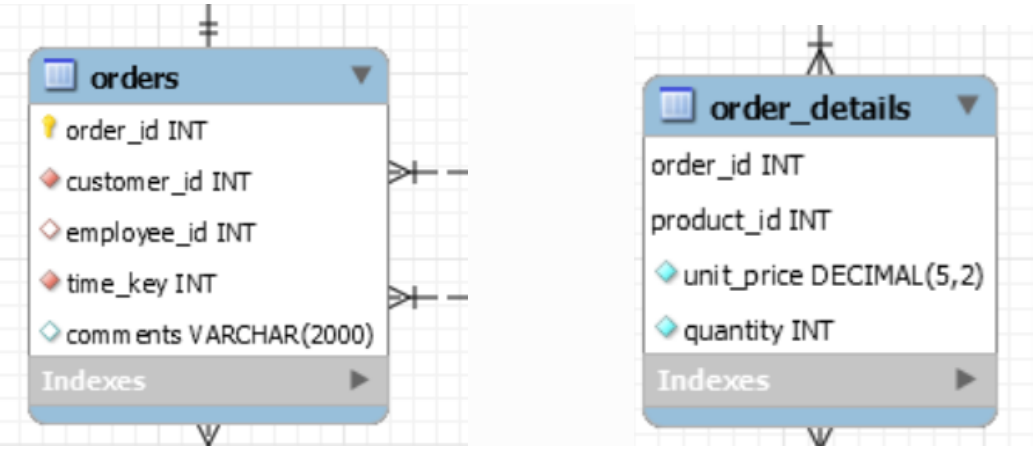
Top-Down Approach:



Fact Tables

The **orders** table and **order_details** table are the fact tables in our data warehouse. For the orders table, **order_id** is the primary key and we have **customer_id, employee_id, time_key** as foreign keys. Whereas in the **order_details** table, there are two foreign keys namely **order_id** and **product_id.**

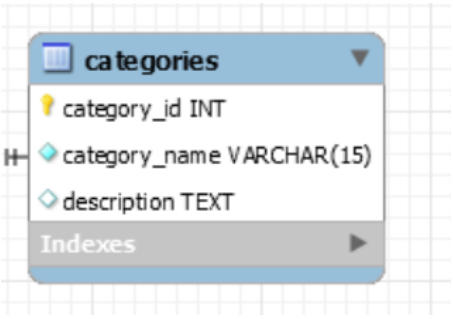
Customer_id, employee_id, and time_key respectively belongs to customer_info dimension, employee dimension, and time_info dimension. In addition, time_key is kept as a primary key to identify the records uniquely. One customer can visit store multiple times on different days. A multi-dimensional model or a star schema should have at least one variation of the 'Time'.



Dimension Tables

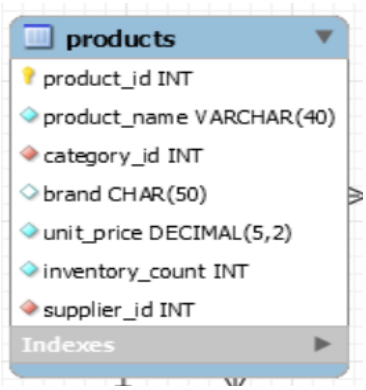
1. Categories Dimension

The description of the categories is stored in this table. The idea to have this table is we can identify which product belongs to which category. We have category_id as a primary key in category table.



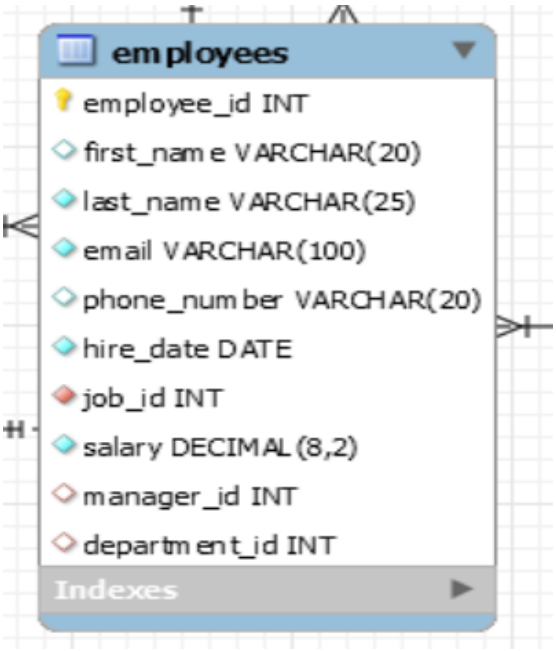
2. Product Dimension

We have product_id as a primary key in this dimension table. Along with this we have category_id and supplier_id as foreign keys. The idea to have this table is it will be easy to identify which products belong to which categories and further tracking will be easy. It has flexible structure; thus, any new product can be easily added.



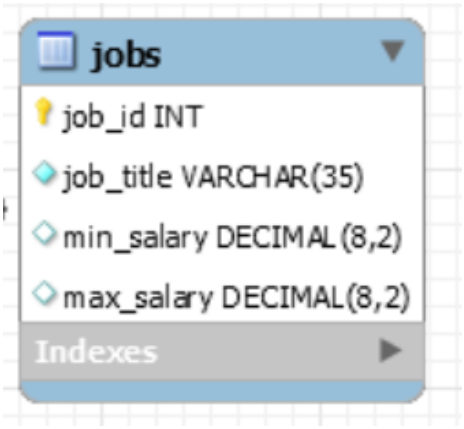
3. Employees Dimension

In the employee dimension table, there is information stored about the employee details like employees first name, last name, email, phone, and salary. The primary key is the employee_id. The aim of this table is to have all the employee details. It has a flexible structure; thus, a new employee can be added easily.



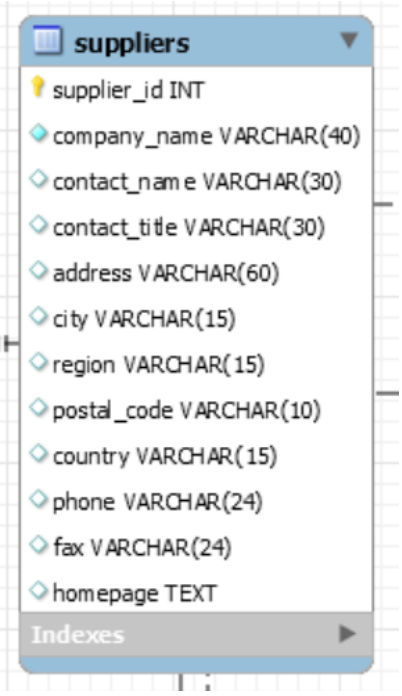
4. Jobs Dimension

We used "Jobs" dimension table to store employee details like job_id, job_title, min_salary, max_salary. We store specific Job details of an employee in this table and job_id serves as a primary key. The idea to have this table is we can calculate the expenses on employee's salaries.



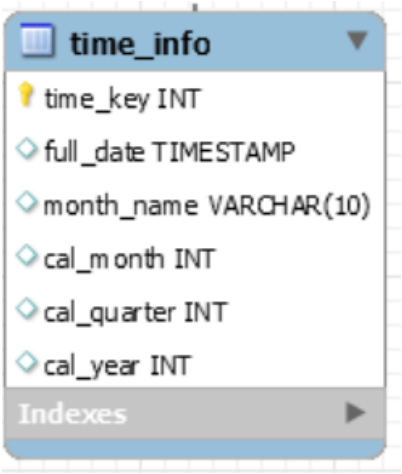
5. Suppliers Dimension

The suppliers dimension table is an important table to have in the data warehouse as it helps in managing the inventory in a systematic manner. It has supplier_id as the primary key and supplier details like company name, city, and country details.



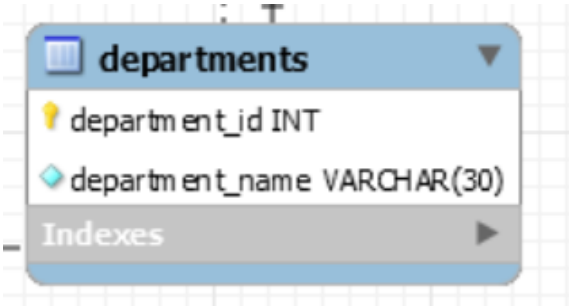
6. Time_info Dimension

Tracking sales as per the timeline be it monthly, yearly, or daily is essential to run a successful business. Hence to track monthly sales at our KTSY store, we decided to include a time_info dimension table with crucial details like time_key, full_date, month_name, cal_month, cal_quarter, cal_year. As time_key is unique and is useful to track monthly sales, we have used the same as primary key.



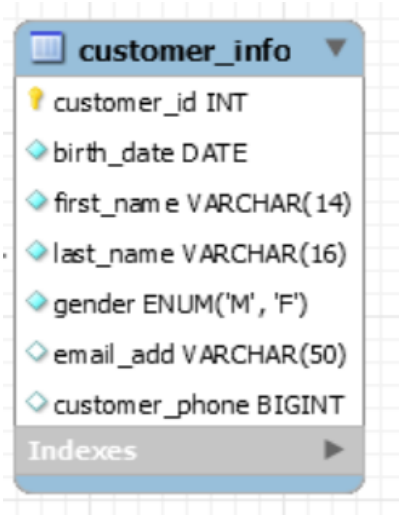
7. Departments Dimension

The department dimension is an extension of the employee table and has details about the department name and department_id which acts as a primary key.



8. Customer_info Dimension

customer_info dimension describes every customer to whom the company's products are sold. customer_id is the primary key and there are other attributes like first_name, last_name, gender, email, and phone number.



The essential components of our warehouse architecture are discussed below:

External Sources: External source is a source from where data is collected irrespective of the type of data. We have generated the data by ourselves after performing critical analysis on the business questions.

Staging Area: We use ETL tools in general to format the data.

E(Extracted): Data is extracted from External data sources.

T(Transform): Data is transformed into the standard format.

L(Load): Data is loaded into a data warehouse after transforming it into the standard format.

However, the data we generated for our KTSY store is specific to our business needs hence we did not use any specific ETL tools as the data is already available in a proper format.

Data-warehouse: After cleaning the data, it is stored in the data warehouse as a central repository. It actually stores the metadata, and the actual data gets stored in the data marts. We noticed that the data warehouse stores the data in its purest form in this top-down approach.

Data Marts: Data mart is a part of the storage component. The information of a particular function of our store is handled by a single authority using "Access Controls". We can also say that our data mart contains a subset of the data stored in the data warehouse.

Data Mining: Data Mining is used to find the hidden patterns that are present in the database or in a data warehouse with the help of algorithms of data mining. The next improvement for our data model is to include data mining algorithms to find out hidden patterns.

Advantages of Top-Down Approach: This model provides a consistent dimensional view of data marts. This model is considered as the strongest model for business changes. Creating a data mart from a data warehouse is easy.

b) Assumptions and gaps

- The data warehouse focuses on only one grocery store named "KTSY" in the city of San Jose.
- The store does not do any home deliveries for their products.
- The prices of all the products in the store are in USD.
- There are no discounts offered in this store for now.
- We have product suppliers from all over the world.

3. Value proposition of Data warehouse and KTSY Store

a) Value proposition of Data warehouse

Companies are realizing that information is the key to achieving a competitive advantage for survival. The data warehouse is used for strategic decision making where management in the organization will utilize existing information to make decisions that have an impact on the future. Thus, it is important to have information that is valid, accurate, and relevant on hand always. Enhancing data quality is what makes the data warehouse a necessity in today's business.

In the absence of a data warehouse, it can be difficult to understand the relationship between the products that are getting sold during a specific weekend or a holiday. With a data warehouse, the organization can have data from multiple sources and transform it into meaningful analysis that will eventually help the store increase sales and build more loyal customers.

For instance, a major chunk of annual sales happens during festive seasons for retail players. Hence, it is extremely critical for them to scale up to meet the increased demand during such

peak seasons and then scale down to avoid excess inventory when the sales return to normal. Store management cannot understand this without the help of a properly designed data warehouse.

Data warehouse identifies new relationships in the data which might have been overlooked.

The increase in revenue of the store can be evaluated by comparing the number of new customers with the old one and whether the old customers are coming back; the market condition and whether the company is ahead of its competitors; whether the company is keeping up with changes in the environment.

In addition, it can also be measured on the hours saved in retrieving material that can be quantified in monetary (Dollar) value. The decreased costs are evaluating the overall efficiency in the company after implementing the data warehouse.

b) Why is this data warehouse worthwhile investment

- It can give new and often ***surprising insights about customer behavior***; thereby helping the retailers meet their ever-changing needs and desires.
- On the supply side, a data warehouse can help retailers ***identify their best vendors*** and determine what separates them from not so good vendors.
- It can give retailers ***better understanding of inventory*** and its movement and help improve storefront operations through ***better category management*** through a host of analyses and reports.
- A data warehouse can also ***improve retailers' internal organizational support functions like finance and human resource management.***
- With that approach, data warehouse can enable a complete view of the customer, helping to improve campaign performance, ***minimize churn***, and, ultimately, ***to grow revenue***.
- Investing in building a data warehouse will eventually help the retail store in reducing their costs and increase their product sales ultimately increasing the revenue and building good customer relationships.

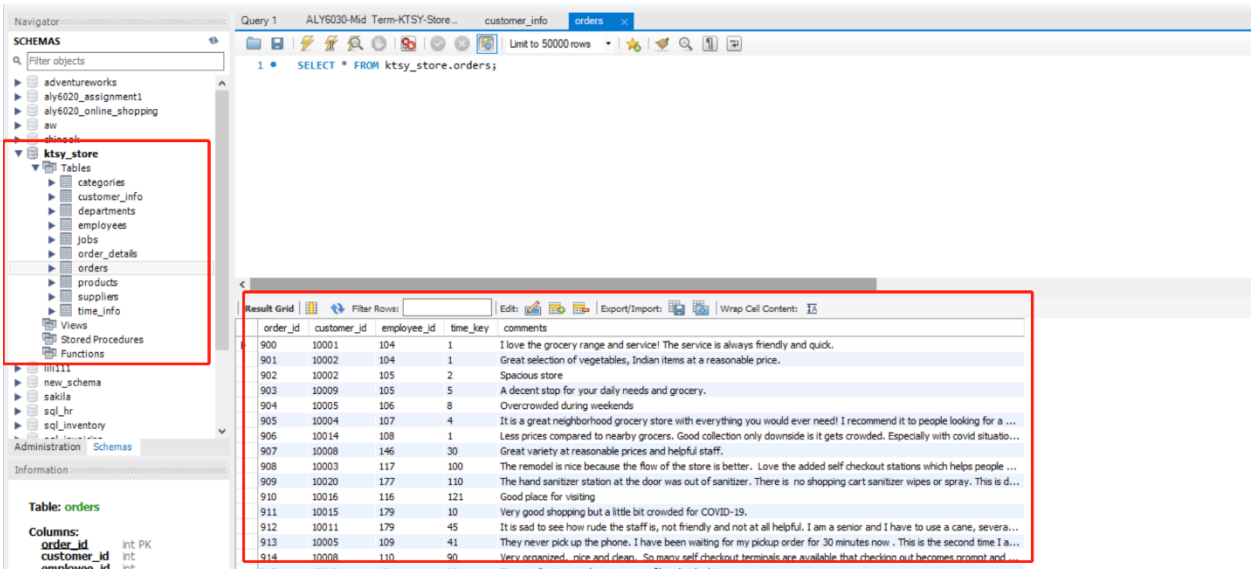
c) Value proposition of KTSY Store

- The KTSY store offers low prices for a broad range of products that are always in stock and ***guarantees consumer satisfaction***.
- The store has enough ***trained sales employees*** present physically at the store to help customers with the shopping.
- With the inclusion of a data warehouse, the store’s efficiency and productivity will become more profitable as it will have an information system to track the inventory.
- The store has witnessed a gradual ***increase in the customer footfall*** in the month of July and been doing a good in the sales department compared to the other months as well.

- The store sales hit a roadblock in the month of march due to COVID but there has been a tremendous increase in the growth rate for the month of July ***(71% increase)***.
- The store sales show a promising increase in the 3rd quarter!

4. Load schema with data

We created 10 tables and inserted our data into our tables. Below is the screenshot of the **order** tables



5. Business questions and supported queries

We specify 17 business questions for our business analysis and write the queries to support those questions. Each query would follow with its result.

```
/* What is the resource head count and salary expenses for each department */
SELECT
    d.department_id,
    d.department_name,
    COUNT(e.employee_id) AS 'Resource head count ',
    SUM(e.salary) AS 'Expenses'
FROM
    departments d
        JOIN
    employees e ON d.department_id = e.department_id
GROUP BY d.department_name
ORDER BY d.department_id ASC;
```

department_id	department_name	Resource head count	Expenses
1	Administration	1	4400.00
2	Marketing	2	19000.00
3	Purchasing	1	3100.00
4	Human Resources	1	6500.00
5	Shipping	7	41200.00
6	IT	1	9000.00
7	Public Relations	1	10000.00
8	Sales	20	178300.00
10	Finance	4	30600.00
11	Accounting	2	20300.00

2
```sql
/*How many distinct products ktsy_store have? */
SELECT
    COUNT(DISTINCT product_name) AS 'Total number of Products'
FROM
    products;
```

Result Grid | Filter Rows:

Total number of Products
38

3
```sql
/* What is the inventory count for each categories? */
SELECT
    products.category_id,
    categories.category_name,
    SUM(products.inventory_count) AS 'Total inventory count by categories'
FROM
    products
        JOIN
    categories ON products.category_id = categories.category_id
GROUP BY categories.category_name
ORDER BY products.category_id ASC;
```

category_id	category_name	Total inventory count by categories
11	Beverages	1530
22	Condiments	2053
33	Confections	2765
44	Dairy Products	2240
55	Grains/Cereals	1482
66	Meat/Poultry	520
77	Produce	254
99	Personal Care	2711

4
```sql
/* Which are the top 10 expensive products? */
SELECT DISTINCT
    product_name AS Ten_Most_Expensive_Products, unit_price
FROM
    products AS a
WHERE
    10 >= (SELECT
            COUNT(DISTINCT unit_price)
        FROM
            Products AS b
        WHERE
            b.unit_price >= a.unit_price)
ORDER BY unit_price DESC;
```

Ten_Most_Expensive_Products	unit_price
Makeup remover	15.75
Body wash	13.75
Body Lotion	12.99
Ground Chicken	12.75
Prawns	11.75
Whipping Cream	11.35
cereals	10.50
Tissues	10.50
Body wash	10.35
Bacon	9.75
Nutella	9.75
Dark Cholocate	9.35

5
```sql
/* What types of products are not getting sold? */
SELECT
    product_id, product_name, brand, unit_price
FROM
    products
WHERE
    product_id NOT IN (SELECT
            product_id
        FROM
            order_details
        WHERE
            product_id IS NOT NULL)
ORDER BY product_name ASC;
```

product_id	product_name	brand	unit_price
54	Body Lotion	Victoria Secret	8.35
48	Body wash	Victoria Secret	10.35
44	cookie biscuits	Chips Ohio	6.35
20	Ground Chicken	Poultry	12.75
15	Ice-cream	Magnum	6.00
38	Macoroni Pasta	Barilla	4.95
23	Prawns	Local	11.75
18	Tiramisu	Local	4.75
30	Waffle	Eggo	3.75

6
```sql
/* Which is the most favoured brand by categories and most favoured product in that category*/
SELECT
    products.supplier_id,
    suppliers.company_name,
    products.product_id,
    products.product_name,
    products.brand,
    categories.category_name,
    COUNT(products.product_name) AS 'Items sold'
FROM
    products
        JOIN
    order_details ON products.product_id = order_details.product_id
        JOIN
    orders ON order_details.order_id = orders.order_id
        JOIN
    categories ON products.category_id = categories.category_id
        JOIN
    suppliers ON products.supplier_id = suppliers.supplier_id
GROUP BY products.category_id
ORDER BY products.supplier_id ASC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

supplier_id	company_name	product_id	product_name	brand	category_name	Items sold
1	Exotic Liquids	11	Roasted Coffee	Starbucks	Beverages	20
2	New Orleans Cajun Delights	41	Bacon	Farmer John	Meat/Poultry	1
8	Specialty Biscuits, Ltd.	17	Garlic Bread	Texas Toast	Confections	8
8	Specialty Biscuits, Ltd.	43	Nuts	Local	Produce	1
9	PB Knäckebröd AB	16	cereals	Kellogs	Grains/Cereals	13
11	Heli Süßwaren GmbH & Co. KG	10	Pepperjack Cheese	Kraft	Dairy Products	30
12	Plutzer Lebensmittelgroßmärkte AG	24	Pesto sauce	Ralph	Condiments	14
13	Nord-Ost-Fisch Handelsgesellschaft mbH	19	Tissues	kirkland	Personal Care	13

```sql
/* What is the sales revenue by products for each quarter? */
SELECT
    products.product_name,
    categories.category_name,
    time_info.cal_quarter,
    ROUND(SUM(order_details.unit_price * order_details.quantity),
            2) AS 'Sales revenue'
FROM
    products
        JOIN
    order_details ON products.product_id = order_details.product_id
        JOIN
    orders ON order_details.order_id = orders.order_id
        JOIN
    categories ON products.category_id = categories.category_id
        JOIN
    time_info ON orders.time_key = time_info.time_key
GROUP BY products.product_name
ORDER BY time_info.cal_quarter ASC;
```

product_name	category_name	cal_quarter	Sales revenue
Pepperjack Cheese	Dairy Products	1	11.25
Roasted Coffee	Beverages	1	207.08
Vanilla Yogurt	Dairy Products	1	118.75
Milk	Dairy Products	1	71.68
Salted Butter	Dairy Products	1	190.15
cereals	Grains/Cereals	1	105.00
Garlic Bread	Confections	1	38.00
Rotini Pasta	Grains/Cereals	1	105.05
Toothpaste	Personal Care	1	7.30
Nutella	Condiments	1	185.25
Ketchup	Condiments	1	23.31
Coca cola Soda	Beverages	1	54.25
Body wash	Personal Care	1	178.75
Orange Juice	Beverages	1	21.56
Protein Oats	Grains/Cereals	1	11.10
Nuts	Produce	1	6.75
Body Lotion	Personal Care	1	207.84
Mayonnaise	Condiments	1	60.15
Dark Cholocate	Confections	1	74.80
Pesto sauce	Condiments	2	10.70
Cream cookies	Confections	2	74.75
Dark chocolate	Confections	2	11.25
Chocolate cookie	Confections	2	9.18
Almond Milk	Dairy Products	2	57.00
Tomato Basil sauce	Condiments	2	6.66
Whipping Cream	Dairy Products	2	22.70
Tissues	Personal Care	3	31.50
Makeup remover	Personal Care	3	63.00
Caramel Frappe	Beverages	3	19.00
Bacon	Meat/Poultry	3	29.25
Cheddar Cheese	Dairy Products	3	71.25

8 /* What is the demand for products based on the month */
/* Provide quantity sales for every product per month and provide ranking based on that. */

```sql
WITH test_1 AS(
   SELECT
ti.Month_Name,
SUM(od.quantity) AS quantity_sales,
p.product_name,
p.brand,
ti.cal_month
FROM orders o
JOIN customer_info ci
    ON o.customer_id = ci.customer_id
JOIN order_details od
    ON o.order_id = od.order_id
JOIN products p
    ON p.product_id = od.product_id
JOIN time_info ti
    ON ti.time_key = o.time_key
GROUP BY p.product_name, p.brand
)
  SELECT
      Month_Name,
      product_name,
      brand,
      quantity_sales,
      DENSE_RANK() OVER (PARTITION BY cal_month ORDER BY quantity_sales DESC) AS demand_rank
 FROM test_1
 ORDER BY cal_month ASC;
```

Month_Name	product_name	brand	quantity_sales	demand_rank
January	Salted Butter	Clover	25	1
January	Vanilla Yogurt	Clover	25	1
January	Body Lotion	Aveno	16	2
January	Roasted Coffee	Starbucks	9	3
January	Dark Cholocate	Lindt	8	4
January	Coca cola Soda	Coca-cola	7	5
January	Roasted Coffee	Nestle	7	5
January	Pepperjack Cheese	Kraft	3	6
January	Protein Oats	Nature Valley	2	7
February	Cheddar Cheese	Clover	15	1
February	Body wash	Dove	13	2
February	Milk	Lucerne	6	3
February	Mayonnaise	Best Foods	6	3
February	Orange Juice	Simply	4	4
February	Ketchup	Heinz	3	5
February	Nuts	Local	1	6
March	Salted Butter	Signature	4	1
March	Milk	Clover	2	2
March	Toothpaste	Colgate	2	2
June	Nutella	Ferrero	19	1
June	Rotini Pasta	Signature	13	2
June	Cream cookies	Oreo	13	2
June	Roasted Coffee	Peets	12	3
June	Almond Milk	Silk	12	3
June	cereals	Kellogs	10	4
June	Garlic Bread	Texas Toast	8	5
June	Dark chocolate	Hersheys	3	6
June	Whipping Cream	Clover	2	7
June	Chocolate cookie	Twix	2	7
June	Pesto sauce	Ralph	2	7
June	Tomato Basil sauce	Prego	1	8
July	Caramel Frappe	Starbucks	4	1
July	Makeup remover	Neutrogena	4	1
July	Mayonnaise	Kraft	3	2
July	Tissues	kirkland	3	2
July	Bacon	Farmer John	3	2

9
```sql
/* What is the customer footfall for each quarter? */
SELECT
    t.Month_name,
    t.cal_quarter,
    COUNT(o.customer_id) AS 'Customer Footfall'
FROM
    time_info t
        JOIN
    orders o ON t.time_key = o.time_key
GROUP BY t.Month_name
ORDER BY t.cal_quarter ASC;
```

	Month_name	cal_quarter	Customer Footfall
▶	January	1	12
	February	1	12
	March	1	5
	June	2	11
	July	3	16

10
```sql
/* What type of products are oftenly purchased by customers? */
SELECT
    cat.category_name,
    cat.description,
    COUNT(category_name) AS purchased_amount
FROM
    categories cat
        JOIN
    products p ON cat.category_id = p.category_id
        JOIN
    order_details od ON od.product_id = p.product_id
        JOIN
    orders o ON od.order_id = o.order_id
GROUP BY category_name
ORDER BY purchased_amount DESC;
```

	category_name	description	purchased_amount
▶	Dairy Products	Cheeses	30
	Beverages	Soft drinks, coffees, teas, beers, and ales	20
	Condiments	Sweet and savory sauces, relishes, spreads, an...	14
	Grains/Cereals	crackers, pasta, and cereal	13
	Personal Care	Bath tissues	13
	Confections	Desserts, candies, and sweet breads	8
	Meat/Poultry	Prepared meats	1
	Produce	Dried fruit and bean curd	1

11
```sql
/* What is the purchse pattern of customers looks like? */
SELECT
    ti.Full_date, ci.first_name, ci.last_name, p.product_name
FROM
    orders o
        JOIN
    customer_info ci ON o.customer_id = ci.customer_id
        JOIN
    order_details od ON o.order_id = od.order_id
        JOIN
    products p ON p.product_id = od.product_id
        JOIN
    time_info ti ON ti.time_key = o.time_key
ORDER BY ti.Full_date ASC;
```

Full_date	first_name	last_name	product_name
2020-01-21 00:00:00	Georgi	Facello	Pepperjack Cheese
2020-01-21 00:00:00	Georgi	Facello	Roasted Coffee
2020-01-21 00:00:00	Georgi	Facello	Body Lotion
2020-01-21 00:00:00	Berni	Genin	Milk
2020-01-21 00:00:00	Bezalel	Simmel	Roasted Coffee
2020-01-21 00:00:00	Bezalel	Simmel	Salted Butter
2020-01-21 00:00:00	Bezalel	Simmel	Coca cola Soda
2020-01-22 00:00:00	Bezalel	Simmel	Vanilla Yogurt
2020-01-23 00:00:00	Prasadram	Heyers	Roasted Coffee
2020-01-24 00:00:00	Chirstian	Koblick	Vanilla Yogurt
2020-01-25 00:00:00	Sumant	Peac	Roasted Coffee
2020-01-25 00:00:00	Sumant	Peac	Salted Butter

12
```sql
/* Rank products based on monthly sales. Display the results for each month */
WITH test_1 AS(
   SELECT
   ti.Month_Name,
   ROUND(SUM(od.unit_price*quantity),2) AS month_sales_amount,
   p.product_name,
   p.brand,
   ti.cal_month
FROM  order_details od
JOIN  orders o
    ON o.order_id = od.order_id
JOIN time_info ti
    ON ti.time_key = o.time_key
JOIN products p
    ON p.product_id = od.product_id
GROUP BY product_name, p.brand
ORDER BY cal_month, month_sales_amount DESC
)
  SELECT
      Month_Name,
      product_name,
      brand,
      month_sales_amount,
      DENSE_RANK() OVER (PARTITION BY cal_month ORDER BY month_sales_amount DESC) AS sales_amount_rank
 FROM test_1
 ORDER BY cal_month ASC;
```

Month_Name	product_name	brand	month_sales_amount	sales_amount_rank
January	Body Lotion	Aveno	207.84	1
January	Salted Butter	Clover	168.75	2
January	Vanilla Yogurt	Clover	118.75	3
January	cereals	Kellogs	105.00	4
January	Roasted Coffee	Peets	83.88	5
January	Roasted Coffee	Starbucks	78.75	6
January	Dark Cholocate	Lindt	74.80	7
January	Milk	Lucerne	55.50	8
January	Coca cola Soda	Coca-cola	54.25	9
January	Roasted Coffee	Nestle	44.45	10
January	Pepperjack Ch...	Kraft	11.25	11
January	Protein Oats	Nature V...	11.10	12
February	Nutella	Ferrero	185.25	1
February	Body wash	Dove	178.75	2
February	Rotini Pasta	Signature	90.35	3
February	Mayonnaise	Best Foods	44.10	4
February	Garlic Bread	Texas To...	38.00	5
February	Ketchup	Heinz	23.31	6
February	Orange Juice	Simply	21.56	7
February	Salted Butter	Signature	21.40	8
February	Nuts	Local	6.75	9
March	Milk	Clover	16.18	1
March	Toothpaste	Colgate	7.30	2
June	Cream cookies	Oreo	74.75	1
June	Almond Milk	Silk	57.00	2
June	Whipping Cream	Clover	22.70	3
June	Dark chocolate	Hersheys	11.25	4

13
```sql
/* What is the average amount spent by each customer ?*/
SELECT
    ci.customer_id,
    ci.first_name,
    ci.last_name,
    ROUND(AVG(od.unit_price * quantity), 2) AS average_amount_spent
FROM
    order_details od
        JOIN
    orders o ON o.order_id = od.order_id
        JOIN
    customer_info ci ON o.customer_id = ci.customer_id
        JOIN
    time_info ti ON ti.time_key = o.time_key
        JOIN
    products p ON p.product_id = od.product_id
GROUP BY first_name , last_name
ORDER BY ci.customer_id ASC;
```

customer_id	first_name	last_name	average_amount_spent
10001	Georgi	Facello	17.59
10002	Bezalel	Simmel	14.44
10003	Parto	Bamford	12.20
10004	Chirstian	Koblick	19.00
10005	Kyoichi	Maliniak	25.15
10006	Anneke	Preusig	15.33
10007	Tzvetan	Zielinski	43.06
10008	Saniya	Malloufi	13.28
10009	Sumant	Peac	21.00
10010	Diaz	Piveteau	16.48
10011	Mary	Sluis	18.68
10012	Patricio	Bridgland	20.16
10013	Eberhardt	Terkki	11.25
10014	Berni	Genin	17.83
10015	Guoxiang	Nooteboom	12.10
10016	Kazuhito	Cappelletti	18.37
10017	Cristinel	Bouloucos	24.01
10019	Lillian	Haddadi	13.10
10020	Mayuko	Warwick	27.69
10022	Shahaf	Famili	13.90
10023	Bojan	Montema...	14.25
10024	Suzette	Pettey	23.75
10025	Prasadram	Heyers	13.62
10028	Domenick	Tempesti	29.25
10029	Otmar	Herbst	34.11
10030	Elvis	Demeyer	18.12
10031	Karsten	Joslin	27.75
10032	Jeong	Reistad	15.50
10033	Arif	Merlo	17.53
10034	Bader	Swan	5.35
10035	Alain	Chappelet	27.50

14 /* Which is the best sales month in ktsy_store? */

```
SELECT
    ti.Month_Name,
    ROUND(SUM(od.unit_price * quantity), 2) AS monthly_sales
FROM
    order_details od
        JOIN
    orders o ON o.order_id = od.order_id
        JOIN
    time_info ti ON ti.time_key = o.time_key
GROUP BY Month_Name
ORDER BY monthly_sales DESC;
```

Month_Name	monthly_sales
July	612.21
January	511.10
February	416.42
June	357.58
March	186.95

15
```sql
/* What is the growth rate in sales revenue? Compare the sales revenue of the current month with the previous month sales revenue*/
WITH monthly_sales AS(
   SELECT
       ti.Month_Name,
       ROUND(SUM(od.unit_price*quantity),5) AS monthly_sales,
       ti.Full_date
    FROM  order_details od
    JOIN  orders o
        ON o.order_id = od.order_id
    JOIN time_info ti
        ON ti.time_key = o.time_key
    GROUP BY Month_Name
    ORDER BY ti.Full_date
)
  SELECT
      Month_Name,
      monthly_sales,
      LAG(monthly_sales,1) OVER( ORDER BY Full_date)  previous_sales,
      FORMAT ( (monthly_sales - previous_sales) / previous_sales  *100,
      4
      ) AS growth_rate,'%'
 FROM (SELECT
    Month_Name, monthly_sales,
    Full_date,
    LAG(monthly_sales,1) OVER(ORDER BY Full_date)  previous_sales
    FROM monthly_sales) AS growth_rate;
```

	Month_Name	monthly_sales	previous_sales	growth_rate	%
▶	January	511.10000	NULL	NULL	%
	February	416.42000	511.10000	-18.5248	%
	March	186.95000	416.42000	-55.1054	%
	June	357.58000	186.95000	91.2704	%
	July	612.21000	357.58000	71.2092	%

16
```sql
/* From comments, what kind of information management can get so that they can implement some necessary changes?*/
/* Filter reviews which is likely positive*/
SELECT
    *
FROM
    orders
WHERE
    comments REGEXP 'best|love|favorite|awesome|great';
/* Filter reviews which is likely negative*/
SELECT
    *
FROM
    orders
WHERE
    comments REGEXP 'bad|poor|not';
```

order_id	customer_id	employee_id	time_key	comments
900	10001	104	1	I love the grocery range and service! The service is always friendly and quick.
901	10002	104	1	Great selection of vegetables, Indian items at a reasonable price.
905	10004	107	4	It is a great neighborhood grocery store with everything you would ever need! I recommend it to people looking for a ...
907	10008	146	30	Great variety at reasonable prices and helpful staff.
908	10003	117	100	The remodel is nice because the flow of the store is better. Love the added self checkout stations which helps people ...
915	10010	101	98	Finest collections and great range of hand picked items
921	10006	119	125	One of the best grocery stories in the town, prompt customer support
922	10011	119	100	Wide range of items available with a great quality' and at affordable pricing
925	10020	179	17	Manager Kate is an adorable person and customer friendly, acts promptly on customer's complaints. Best service!
928	10028	177	107	Love the products range in the store. The meat produce is really fresh!
929	10031	108	114	I love the store, their food selection and their grocery pick up service
930	10020	100	111	I have already loved this store but I am loving them even more with their delivery services
939	10033	103	37	This is my favorite grocery store and place to grab a quick healthy dinner.
940	10032	120	120	This is the best grocery!
941	10031	146	17	Great place for cookies and liquor
942	10025	178	3	Great place for cookies and liquor
943	10025	177	11	This is one of my favorite WF markets it has massive and well stocked.
945	10023	108	101	Easy shopping large aisles, great selection on be veggies fish meat
946	10022	178	108	Love the quality of the products. Will definitely visit more often!'
948	10019	107	44	Love the products range in the store. The meat produce is really fresh!
950	10016	110	67	Donuts and coffee, scratchers! Its great. Packages too!
951	10006	103	88	Great location for small buying.
952	10007	105	99	Great prices on beer and great selection with quality service.
953	10009	115	35	I love Swetha, Yes, she is the best sales girl in the store

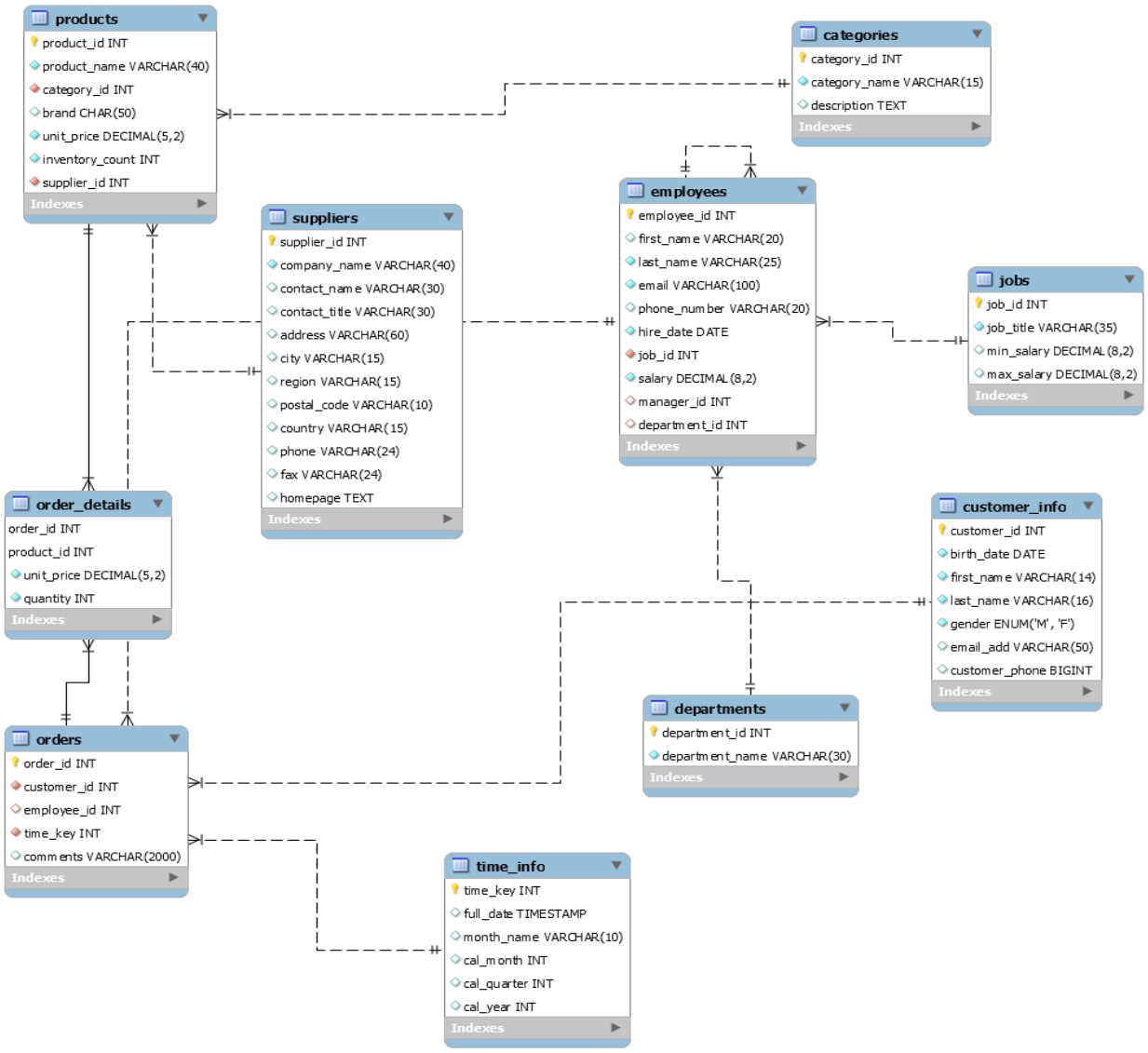
order_id	customer_id	employee_id	time_key	comments
912	10011	179	45	It is sad to see how rude the staff is, not friendly and not at all helpful. I am a senior and I have to use a cane, severa...
916	10012	100	33	Delay in delivery, not upto the expectations
917	10013	106	78	Poor customer service
919	10001	117	90	Did not deliver all the products ordered, worst service
926	10020	109	22	Poor service, there is not fresh meat!
954	10005	146	39	Everything was over priced. Not worth it.
955	10030	178	130	Poor service! It is sad to see how rude the staff is, not friendly and not at all helpful. I am a senior and I have to use a...

17
```sql
/* What is the number of units in stock by category and supplier continent */
SELECT
    categories.category_name AS 'Product Category',
    CASE
        WHEN
            suppliers.country IN ('UK' , 'Spain',
                'Sweden',
                'Germany',
                'Norway',
                'Denmark',
                'Netherlands',
                'Finland',
                'Italy',
                'France')
        THEN
            'Europe'
        WHEN suppliers.country IN ('USA' , 'Canada', 'Brazil') THEN 'America'
        ELSE 'Asia-Pacific'
    END AS 'Supplier Continent',
```

```
    SUM(products.inventory_count) AS Units_In_Stock
FROM
    suppliers suppliers
        INNER JOIN
    products products ON products.supplier_id = suppliers.supplier_id
        INNER JOIN
    categories categories ON categories.category_id = products.category_id
GROUP BY categories.category_name , CASE
    WHEN
        suppliers.country IN ('UK' , 'Spain',
            'Sweden',
            'Germany',
            'Norway',
            'Denmark',
            'Netherlands',
            'Finland',
            'Italy',
            'France')
    THEN
        'Europe'
    WHEN suppliers.country IN ('USA' , 'Canada', 'Brazil') THEN 'America'
    ELSE 'Asia-Pacific'
END;
```

Product Category	Supplier Continent	Units_In_Stock
Beverages	Europe	1212
Beverages	America	128
Beverages	Asia-Pacific	190
Condiments	Europe	1853
Condiments	America	200
Confections	Europe	2622
Confections	America	143
Dairy Products	Europe	1407
Dairy Products	Asia-Pacific	213
Dairy Products	America	620
Grains/Cereals	Europe	834
Grains/Cereals	America	648
Meat/Poultry	America	520
Produce	Europe	254
Personal Care	Europe	2316
Personal Care	Asia-Pacific	395

6. Schematics of how data would flow



We focused on three business functions.

1. **Customer support**

The idea to have **Customer support** function is if any customer needs any assistance while shopping at the store, or faces any issues regarding his/her order or in returning purchased products, the customer support team will be there to help customers and resolve their issues.

We assigned ***employee_id*** as a foreign key in the orders table. This gives the information about which employee helped which customer. Here we can see ***one to many relationship*** (1: M) between **orders** and **employees** tables. One employee can assist many customers.

Since we have recorded a day-to-day footfall of customers, there is a possibility that two different employees can assist the same customer on different days.

Similarly, we have ***customer_id*** and ***time_key*** as foreign keys in the table. This gives the information about customer's purchase details and the day on which they made a purchase. There is ***1: M relationship*** between **orders** and **customer_info**. One customer can have multiple order ids.

We have ***comments*** (feedback) in the **orders** table. This is to understand the customer's sentiments. We export this data into the python environment and performed sentiment analysis. With the help of sentiment analysis, we understood customers' thinking and the issues they are facing. This will help us in improving customer service and experience.

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

full_date	customer_firstname	customer_lastname	employee_id	employee_firstname	employee_lastname	comments
2020-06-19 00:00:00	Georgi	Facello	117	Sigal	Tobias	Did not deliver all the products ordered, worst service
2020-01-21 00:00:00	Georgi	Facello	104	Bruce	Ernst	I love the grocery range and service! The service is always friendly and quick.
2020-01-21 00:00:00	Bezalel	Simmel	104	Bruce	Ernst	Great selection of vegetables, Indian items at a reasonable price.
2020-01-22 00:00:00	Bezalel	Simmel	105	David	Austin	Spacious store
2020-06-29 00:00:00	Parto	Bamford	117	Sigal	Tobias	The remodel is nice because the flow of the store is better. Love the added self che...
2020-01-24 00:00:00	Chirstian	Koblick	107	Diana	Lorentz	It is a great neighborhood grocery store with everything you would ever need! I rec...
2020-01-28 00:00:00	Kyoichi	Maliniak	106	Valli	Pataballa	Overcrowded during weekends
2020-03-01 00:00:00	Kyoichi	Maliniak	109	Daniel	Faviet	They never pick up the phone. I have been waiting for my pickup order for 30 minute...
2020-02-28 00:00:00	Kyoichi	Maliniak	146	Karen	Partners	Everything was over priced. Not worth it.
2020-06-17 00:00:00	Anneke	Preusig	103	Alexander	Hunold	Great location for small buying.
2020-07-24 00:00:00	Anneke	Preusig	119	Karen	Colmenares	One of the best grocery stories in the town, prompt customer support
2020-06-28 00:00:00	Tzvetan	Zielinski	105	David	Austin	Great prices on beer and great selection with quality service.
2020-02-17 00:00:00	Tzvetan	Zielinski	146	Karen	Partners	Delay in issuing refunds.
2020-06-19 00:00:00	Saniya	Malloufi	110	John	Chen	Very organized, nice and clean. So many self checkout terminals are available that c...
2020-02-19 00:00:00	Saniya	Malloufi	146	Karen	Partners	Great variety at reasonable prices and helpful staff.
2020-01-25 00:00:00	Sumant	Peac	105	David	Austin	A decent stop for your daily needs and grocery.
2020-02-24 00:00:00	Sumant	Peac	115	Alexander	Khoo	I love Swetha, Yes, she is the best sales girl in the store
2020-06-27 00:00:00	Diaz	Piveteau	101	Neena	Kochhar	Finest collections and great range of hand picked items
2020-06-29 00:00:00	Mary	Sluis	119	Karen	Colmenares	Wide range of items available with a great quality' and at affordable pricing

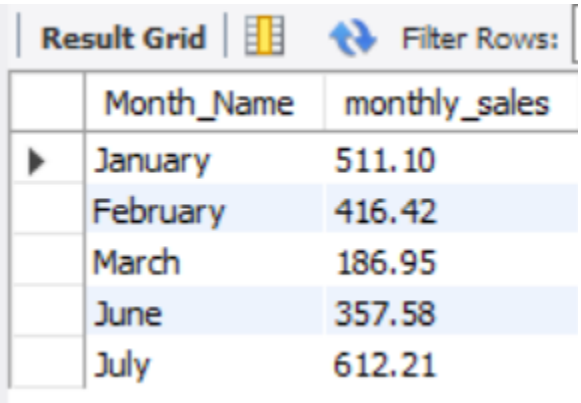
2. Sales analysis

The second function we chose is **Sales**. We defined this in the **order_details table**. This is also our ***Fact table***. In **order_details** table, we defined ***quantity*** and ***unit_price*** of each product which is connected to ***order_id*** in **orders table** and ***product_id*** in **products table**.

The relationship between **products table** and **order_details table** is ***one to many***. Because one order_id can contain multiple products. Furthermore, one customer can visit a store multiple times on different days and can purchase multiple products. Hence the ***order_id*** will be different each time. So, to identify records uniquely, we assigned ***{order_id, product_id***} as **a composite key**.

If we want the details regarding sales revenue in each month, or by each product, or most preferred product brand, or customer's buying pattern, all this information can be retrieved by connecting **order_details**, **customer_info, orders**, and **time_info** tables. For instance, if we want

a sales revenue for each month, this information can be retrieved by joining **order_details**, **order,** and **time_info** table. The output is as follows:



Result Grid | Filter Rows:

Month_Name	monthly_sales
January	511.10
February	416.42
March	186.95
June	357.58
July	612.21

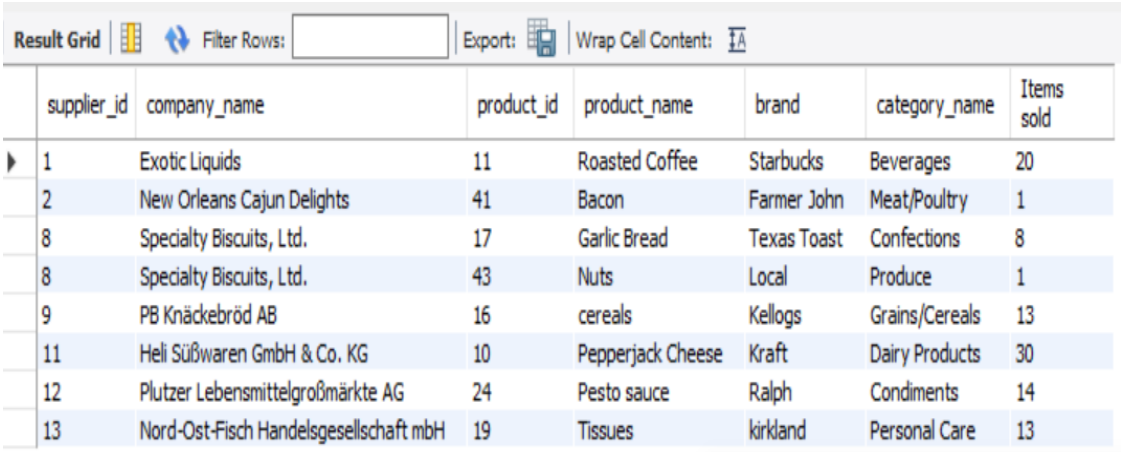
We can see sales in each month. At the same time, we can observe that in the month of July, sales revenue was highest.

We designed a warehouse in such a way that by using simple joins, we able to retrieve the data for analysis.

3. Vendor management

Since we have a retail store, vendor management is an important function and suppliers are a vital part of it. We have suppliers from all over the world. Different suppliers provide multiple products from different brands. We assigned ***supplier_id*** as a foreign key in the **products tables**. One supplier can provide multiple products. Hence, there is ***one to many relationship*** (1: M) between **products** and **suppliers** tables.

The idea to have this function is, we will be able to compare the prices offered by different suppliers. In addition to that, we will be able to offer products at reasonable prices to our customers without compromising on quality.



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

supplier_id	company_name	product_id	product_name	brand	category_name	Items sold
1	Exotic Liquids	11	Roasted Coffee	Starbucks	Beverages	20
2	New Orleans Cajun Delights	41	Bacon	Farmer John	Meat/Poultry	1
8	Specialty Biscuits, Ltd.	17	Garlic Bread	Texas Toast	Confections	8
8	Specialty Biscuits, Ltd.	43	Nuts	Local	Produce	1
9	PB Knäckebröd AB	16	cereals	Kellogs	Grains/Cereals	13
11	Heli Süßwaren GmbH & Co. KG	10	Pepperjack Cheese	Kraft	Dairy Products	30
12	Plutzer Lebensmittelgroßmärkte AG	24	Pesto sauce	Ralph	Condiments	14
13	Nord-Ost-Fisch Handelsgesellschaft mbH	19	Tissues	kirkland	Personal Care	13

When we performed an analysis to find out about the most favored product and its respective brand by the customers, we found that in beverages most favored product is Roasted coffee of Starbucks brand, in dairy most favored product is pepper jack cheese of Kraft brand, and so on. So, this function gives us brief information about the products and respective brands are supplied by which supplier. For example, **Exotic Liquids** supply **Roasted coffee** of **Starbucks** brand. In this way, we can check and compare the products offered by each supplier and concentrate on those suppliers which supply these product brands. This will help us in improving our B2B as well as B2C relationships and ultimately, we will be able to increase our sales and profit margin.

7. Customer sentiment analysis using Polarity and Subjectivity

In today's world, we see a definitive need for every business to understand the future or existing customer's needs, interests, and behavior. With the help of this powerful AI technique ***"Natural Language Processing"*** businesses can automate the process of drawing conclusions and making predictions by analyzing the unstructured or semi-structured data. Using Sentiment Analysis in python, we can estimate a customer's buying behavior and predict possible customer churn. We can do the analysis by examining the two properties "Polarity and Subjectivity", returned from the function "**Textblob**".

The application of sentiment analysis is indeed interesting because it helps us understand customer's sentiments that can be used to enhance the customer experience. **With the help of this analysis,** we were able **to understand what issues customers are facing** and we will be able to take the necessary steps towards improving customer experience. This will help us in **customer retention, reducing the churn, and in attracting new customers**.

For our KTSY model, we tried to predict the customer sentiment using two properties, subjectivity, and polarity. We have used the NLP package "Textblob" in python and tried to predict polarity and subjectivity scores for a given customer comment.

Polarity: Polarity is a float value which lies in the range of [-1,1] where 1 means positive statement and -1 means a negative statement.

Subjectivity: Subjective sentences generally refer to personal opinion, emotion, or judgment whereas objective refers to factual information. Subjectivity is also a float which lies in the range of [0,1].

Analysis in Python: For our analysis, we imported the basic python packages and the NLP package "Textblob". At a later stage, we retrieved the orders table from MySQL workbench in a CSV format and loaded the data into a Python data frame. The orders table contains the details of the customer's comments.

Exploratory Data Analysis: As the data in the warehouse is already formatted and scrubbed, we realized there is no further scrubbing required. Also, we noticed there are no specific trends in

customer comments. Hence, we have only checked for case sensitivity and converted the upper case letters and words in customer's comments to lower case using the “str.lower” function.

Analyzing Positive Comments: The idea behind analyzing positive the sentiment is to search for positive words from customer’s comments and calculate the subjectivity and polarity scores for the comments. To do this, we applied the lambda function on the comments column and applied 1 to the cell if there is a positive word traced and 0 to the cell if a negative comment is encountered.

Basic Sentiment Analysis: The Sentiment property returns a named tuple of the form polarity and subjectivity. The data type of polarity score is float within the range [-1.0, 1.0]. The subjectivity data type is float within the range [0.0, 1.0] where 0.0 is very objective and 1.0 is very subjective.

```
orders_df[['polarity', 'subjectivity']] = orders_df['comments'].apply(lambda Text: pd.Series(TextBlob(Text).sentiment)) #Checking polarity and subjectivity using sentiment property
orders_df
```

	order_id	customer_id	employee_id	time_key	comments	positive_sentiment	polarity	subjectivity
0	900	10001	104	1	i love the grocery range and service! the serv...	1	0.444444	0.533333
1	901	10002	104	1	great selection of vegetables, indian items at...	1	0.500000	0.675000
2	902	10002	105	2	spacious store	0	0.000000	0.000000
3	903	10009	105	5	a decent stop for your daily needs and grocery.	0	0.083333	0.333333
4	904	10005	106	8	overcrowded during weekends	0	0.000000	0.000000
5	905	10004	107	4	it is a great neighborhood grocery store with ...	1	0.500000	0.375000
6	906	10014	108	1	less prices compared to nearby grocers. good c...	1	0.080667	0.733333
7	907	10008	146	30	great variety at reasonable prices and helpful...	1	0.500000	0.675000
8	908	10003	117	100	the remodel is nice because the flow of the st...	1	0.400000	0.650000
9	909	10020	177	110	the hand sanitizer station at the door was out...	0	0.000000	0.000000
10	910	10016	116	121	good place for visiting	1	0.700000	0.600000
11	911	10015	179	10	very good shopping but a little bit crowded fo...	1	0.361250	0.640000
12	912	10011	179	45	it is sad to see how rude the staff is, not fr...	0	-0.246875	0.525000
13	913	10005	109	41	they never pick up the phone. i have been wait...	0	0.000000	0.000000

Interpretation: The screenshot above shows the polarity and subjectivity scores for top 13 customer rows.

Customer_id 10001: 0.44 defines the polarity of the sentence is slightly positive which shows that the customer is positively inclined towards the store and subjectivity value 0.53 indicates the customer’s opinion is fairly subjective and is partially dependent on factual information about the store i.e., objectivity.

Customer_id 10002: 0.50 indicates fair positivity which means the customer has a good positive opinion on the store and subjectivity value 0.675 indicates that the statement is mostly the customer's own opinion.

Customer_id 10003: 0.0 indicates that the customer has a neutral polarity which means the customer has neither a positive opinion nor a negative opinion on the store and 0.0 subjectivity indicates that the statement is very objective which means it is specific to the store.

Customer_id 10011: Polarity -0.246 indicates that the customer has a negative opinion on the store and the subjectivity score 0.535 indicates that the statement made by the customer is based on his own opinion.

As we all know, in the business industry, “customer satisfaction is the key to success”. Hence, we have calculated the “Subjectivity and Polarity” scores for user comments and tried to interpret the customer's behavior. We thereby helped the store management take the necessary steps in the direction of improved customer satisfaction.

8. Conclusion

As the retail industry is expanding and becoming more competitive, data warehousing and analytics have become critical to the success of any retailer. It has become essential for retailers to know how to tie the right, relevant data together to meet business needs and derive valuable consumer insights.

With changes in economic trends especially with innovation in technology, utilization of eCommerce, and globalization data warehousing will seem to be the solution to many organizations. Thus, organizations must consider these changes by incorporating data warehousing to compete in the market.

9. References

1. Shubham Jain (February 11, 2018): Natural Language Processing for Beginners: Using TextBlob retrieved from Analytics Vidhya

https://www.analyticsvidhya.com/blog/2018/02/natural-language-processing-for-beginners-using-textblob/#:~:text=Polarity%20is%20float%20which%20lies,of%20%5B0%2C1%5D

2. Priyanka Gujral (January 11, 2018): Data Warehouse Architecture retrieved from GeeksforGeeks https://www.geeksforgeeks.org/data-warehouse-architecture/

3. Iıl GÜRATAN (July 2005): THE DESIGN AND DEVELOPMENT OF A DATA WAREHOUSE USING SALES DATABASE AND REQUIREMENTS OF A RETAIL GROUP.

https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.427.5803&rep=rep1&type=pdf