CASE STUDY ON GROCERY STORE USING SQL

Table of Contents

Introduction

Overview

The grocery industry generates vast amounts of data daily from transactions, inventory, customer behaviou

r, and supply chain logistics. This case study aims to analyze such data using SQL (Structured Query Language) to derive actionable insights. By leveraging SQL's powerful querying capabilities, we can uncover trends, optimize operations, and enhance decision-making processes.

Objectives

1. Understand Sales Patterns: Identify peak sales periods, best-selling products, and underperforming items.

2. Customer Insights: Analyze customer purchasing behaviour to segment customers and tailor marketing strategies.

3. Inventory Management: Ensure optimal stock levels, identify slow-moving items, and minimize wastage.

4. Supplier Performance: Evaluate supplier reliability and cost-effectiveness.

Data Description

The dataset for this case study includes multiple tables representing different aspects of the grocery business:

Suppliers:

The **Suppliers** table contains information about the vendors who supply products to the grocery store.

Products:

The **Products** table contains details about the products available in the grocery store.

Orders;

The **Orders** table records the orders placed by customers.

OrderDetails:

The **OrderDetails** table captures the details of each product in an order.

Customers:

The **Customers** table holds information about the grocery store's customers.

Employees:

The **Employees** table includes information about the employees of the grocery store.

Categories:

The **Categories** table categorizes the products available in the grocery store.

Relationships Between Tables

- **Suppliers** to **Products**: One-to-many (one supplier can supply multiple products).
- **Products** to **OrderDetails**: One-to-many (one product can appear in multiple order details).
- **Orders** to **OrderDetails**: One-to-many (one order can have multiple order details).
- **Customers** to **Orders**: One-to-many (one customer can place multiple orders).
- **Employees** to **Orders**: One-to-many (one employee can handle multiple orders).
- **Categories** to **Products**: One-to-many (one category can include multiple products).

ER DIAGRAM

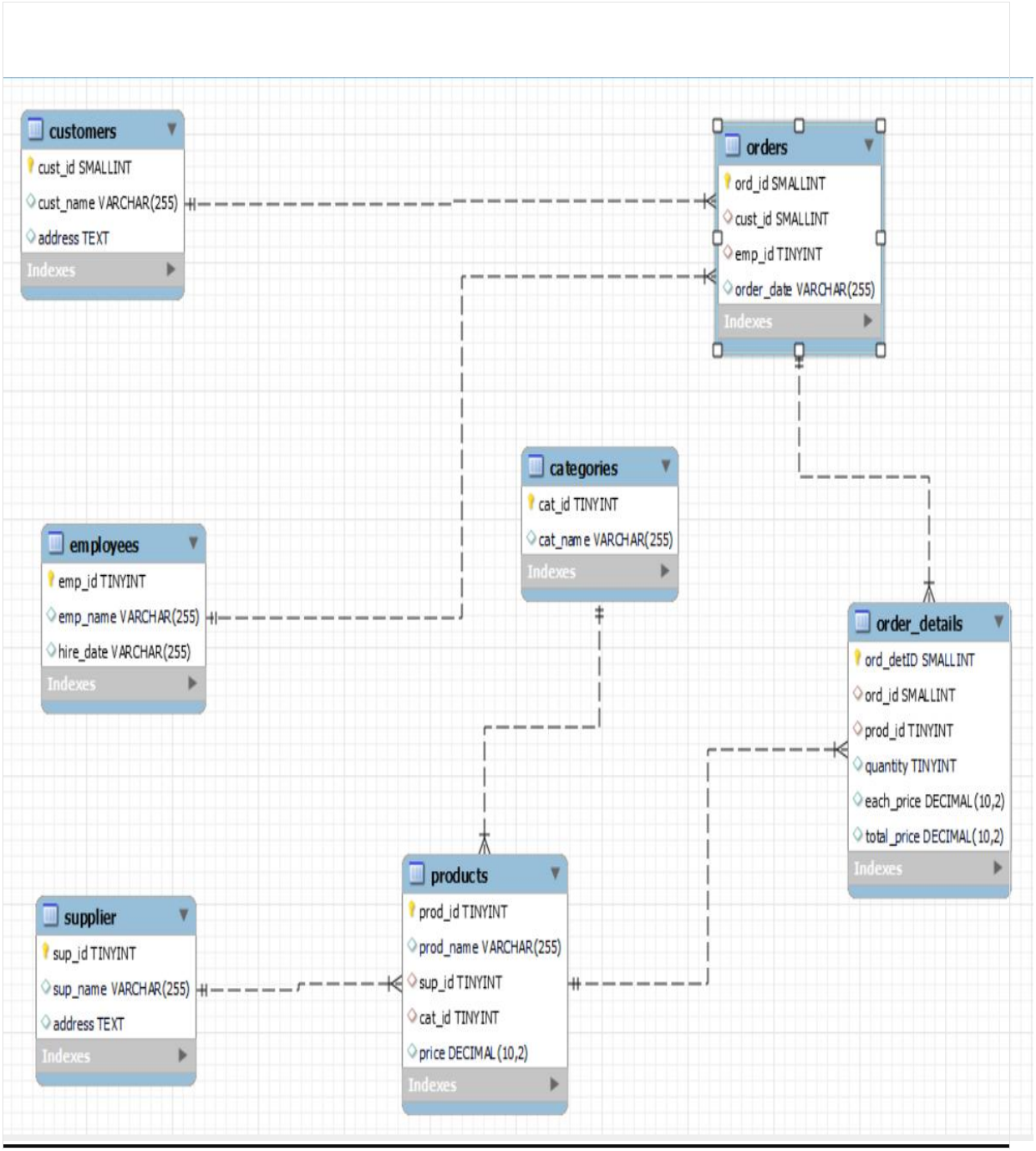


fig: ER Diagram Relationship of Grocery Store

Data Preprocessing

Data preprocessing is a critical step in data analysis, ensuring that the dataset is clean, consistent, and ready for analysis. Here are the detailed steps for data preprocessing in the context of a grocery store case study using SQL:

Data Cleaning:

- **Removing Duplicates:** Ensure there are no duplicate records in any of the tables. Duplicate records can skew analysis results.
- **Handling Missing Values:** Identify and handle missing values. Depending on the context, missing values can be filled, replaced, or removed.
- **Standardizing Formats:** Ensure that data formats are consistent. For example, dates should follow a standard format, phone numbers should have a consistent format, etc.

Data Integration:

- **Joining Tables:** Combine data from different tables to create comprehensive datasets for analysis.
- **Creating Aggregated Tables:** Create summary tables that aggregate data for quicker analysis.

Data Transformation:

- **Normalizing Data:** Scale numerical values to a common scale if needed (e.g., scaling prices, quantities).
- **Creating Derived Columns:** Generate new columns from existing data to facilitate analysis

Data Validation:

- **Consistency Checks:** Ensure that the data maintains consistency across the tables.
- **Referential Integrity:** Ensure that foreign key constraints are maintained to enforce relationships between tables.

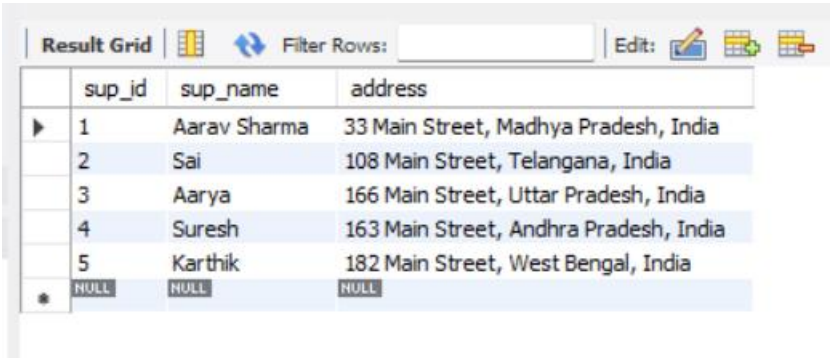
Data Enrichment:

- **Adding External Data:** Enhance your dataset by integrating external data sources.
- **Feature Engineering:** Create new features from existing data to improve analysis.

SQL Queries and Analysis

DISPLAYING TABLES IN GROCERY STORE DATABASE

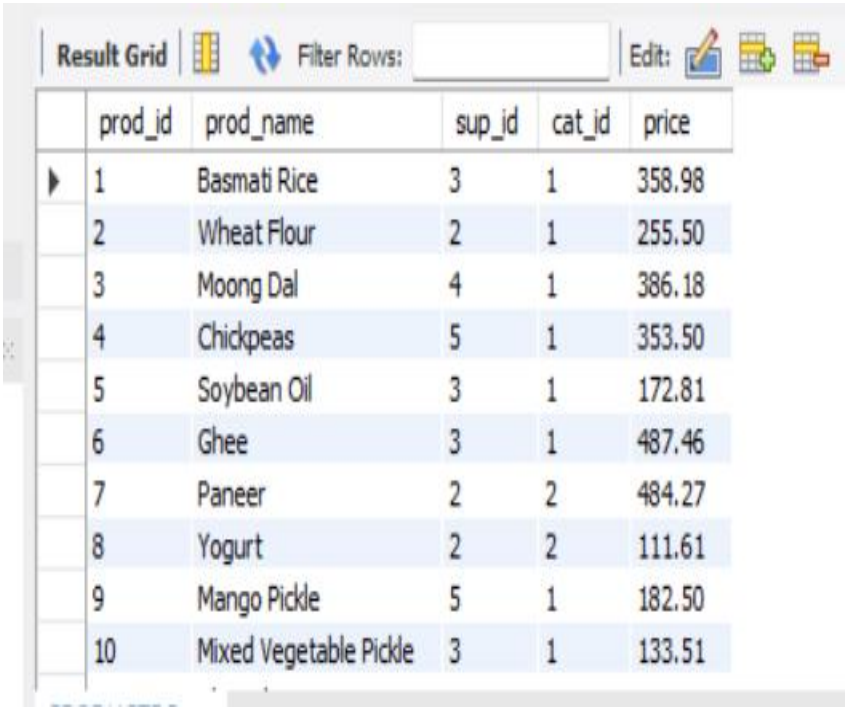
SELECT * FROM SUPPLIER;



Result Grid | Filter Rows: | Edit:

sup_id	sup_name	address
1	Aarav Sharma	33 Main Street, Madhya Pradesh, India
2	Sai	108 Main Street, Telangana, India
3	Aarya	166 Main Street, Uttar Pradesh, India
4	Suresh	163 Main Street, Andhra Pradesh, India
5	Karthik	182 Main Street, West Bengal, India
NULL	NULL	NULL

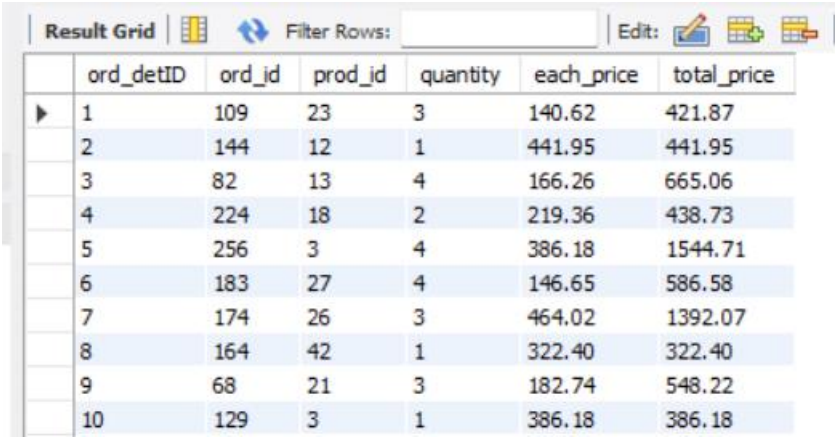
SELECT * FROM PRODUCTS



Result Grid | Filter Rows: | Edit:

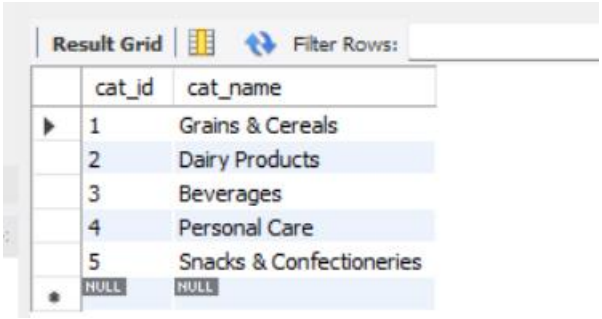
prod_id	prod_name	sup_id	cat_id	price
1	Basmati Rice	3	1	358.98
2	Wheat Flour	2	1	255.50
3	Moong Dal	4	1	386.18
4	Chickpeas	5	1	353.50
5	Soybean Oil	3	1	172.81
6	Ghee	3	1	487.46
7	Paneer	2	2	484.27
8	Yogurt	2	2	111.61
9	Mango Pickle	5	1	182.50
10	Mixed Vegetable Pickle	3	1	133.51

SELECT * FROM ORDER_DETAILS;



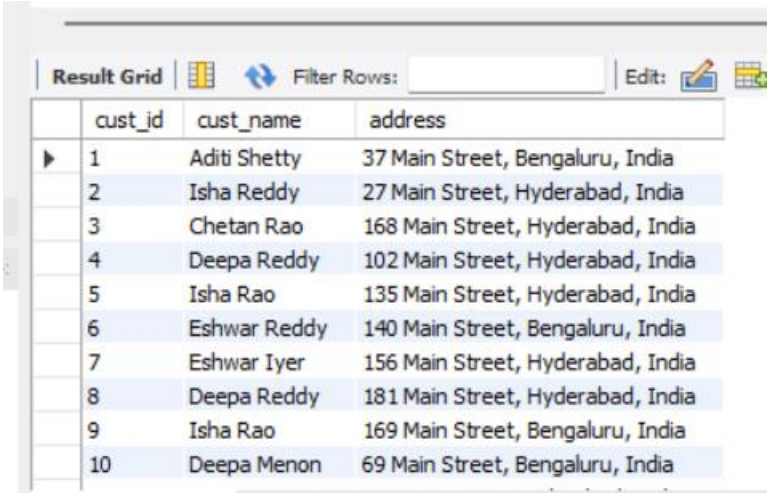
ord_detID	ord_id	prod_id	quantity	each_price	total_price
1	109	23	3	140.62	421.87
2	144	12	1	441.95	441.95
3	82	13	4	166.26	665.06
4	224	18	2	219.36	438.73
5	256	3	4	386.18	1544.71
6	183	27	4	146.65	586.58
7	174	26	3	464.02	1392.07
8	164	42	1	322.40	322.40
9	68	21	3	182.74	548.22
10	129	3	1	386.18	386.18

SELECT * FROM CATEGORIES;



cat_id	cat_name
1	Grains & Cereals
2	Dairy Products
3	Beverages
4	Personal Care
5	Snacks & Confectioneries
NULL	NULL

SELECT * FROM CUSTOMERS;



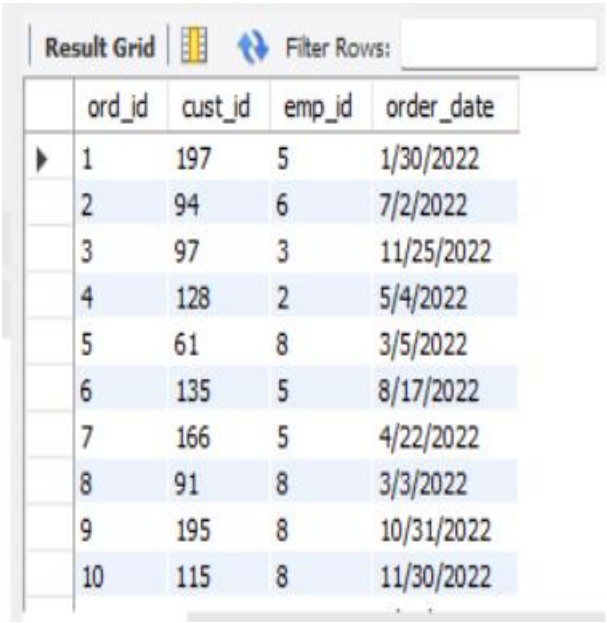
cust_id	cust_name	address
1	Aditi Shetty	37 Main Street, Bengaluru, India
2	Isha Reddy	27 Main Street, Hyderabad, India
3	Chetan Rao	168 Main Street, Hyderabad, India
4	Deepa Reddy	102 Main Street, Hyderabad, India
5	Isha Rao	135 Main Street, Hyderabad, India
6	Eshwar Reddy	140 Main Street, Bengaluru, India
7	Eshwar Iyer	156 Main Street, Hyderabad, India
8	Deepa Reddy	181 Main Street, Hyderabad, India
9	Isha Rao	169 Main Street, Bengaluru, India
10	Deepa Menon	69 Main Street, Bengaluru, India

SELECT * FROM EMPLOYEES;

Result Grid | Filter Rows:

emp_id	emp_name	hire_date
1	Aarav Kumar 1	2/3/2021
2	Aditya Singh 1	1/8/2021
3	Pari Kumar 1	11/12/2021
4	Aditya Verma 1	1/9/2021
5	Pari Sharma 1	2/9/2021
6	Zara Verma 1	10/16/2021
7	Vihaan Singh 1	8/26/2020
8	Diya Sharma 1	8/21/2021
9	Arjun Kumar 1	5/29/2021
10	Arjun Verma 1	4/14/2021

SELECT * FROM ORDERS



Result Grid | Filter Rows:

ord_id	cust_id	emp_id	order_date
1	197	5	1/30/2022
2	94	6	7/2/2022
3	97	3	11/25/2022
4	128	2	5/4/2022
5	61	8	3/5/2022
6	135	5	8/17/2022
7	166	5	4/22/2022
8	91	8	3/3/2022
9	195	8	10/31/2022
10	115	8	11/30/2022

1.List all the categories in the Grocery store?

QUERY:

```
SELECT DISTINCT CAT_NAME FROM CATEGORIES;
```

CAT_NAME
Grains & Cereals
Dairy Products
Beverages
Personal Care
Snacks & Confectioneries

2. List all the products in the Grocery Store?

Query:

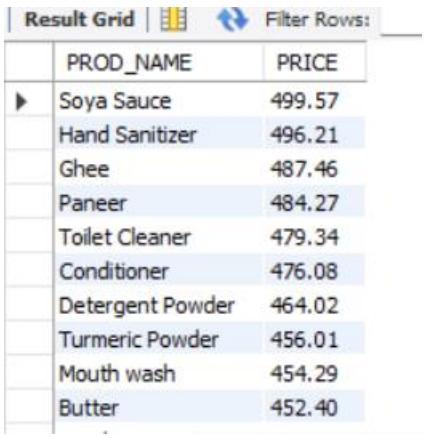
```
SELECT DISTINCT PROD_NAME FROM PRODUCTS;
```

PROD_NAME
Basmati Rice
Wheat Flour
Moong Dal
Chickpeas
Soybean Oil
Ghee
Paneer
Yogurt
Mango Pickle
Mixed Vegetable Pickle

3. Retrieve all products along with their unit prices

Query:

```
SELECT
        PROD_NAME,PRICE
FROM
        PRODUCTS
GROUP BY PRODUCT_NAME,PRICE;
```



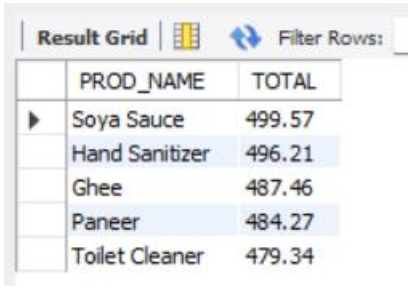
Result Grid | Filter Rows:

PROD_NAME	PRICE
Soya Sauce	499.57
Hand Sanitizer	496.21
Ghee	487.46
Paneer	484.27
Toilet Cleaner	479.34
Conditioner	476.08
Detergent Powder	464.02
Turmeric Powder	456.01
Mouth wash	454.29
Butter	452.40

4.Retrieve details of the top 5 products with the highest unit price.

Query:

```
SELECT
    PROD_NAME, SUM(PRICE) AS TOTAL
FROM
    PRODUCTS
GROUP BY PROD_NAME
ORDER BY TOTAL DESC
LIMIT 5;
```



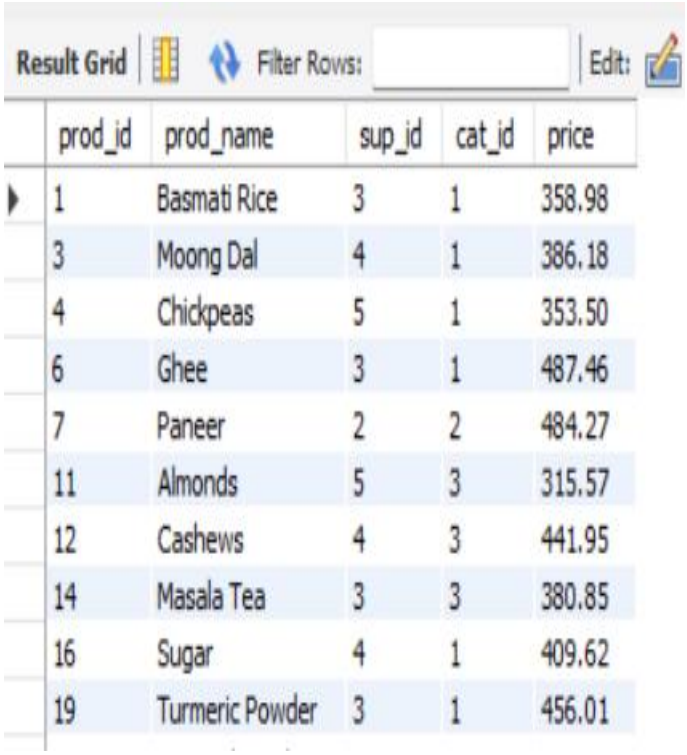
Result Grid | Filter Rows:

PROD_NAME	TOTAL
Soya Sauce	499.57
Hand Sanitizer	496.21
Ghee	487.46
Paneer	484.27
Toilet Cleaner	479.34

5.List the products that have a higher unit price than the average unit price of all products.

Query:

```
SELECT *
   FROM  PRODUCTS
WHERE
    PRICE > (SELECT
          AVG(PRICE)
       FROM
          PRODUCTS);
```



Result Grid | Filter Rows: | Edit:

prod_id	prod_name	sup_id	cat_id	price
1	Basmati Rice	3	1	358.98
3	Moong Dal	4	1	386.18
4	Chickpeas	5	1	353.50
6	Ghee	3	1	487.46
7	Paneer	2	2	484.27
11	Almonds	5	3	315.57
12	Cashews	4	3	441.95
14	Masala Tea	3	3	380.85
16	Sugar	4	1	409.62
19	Turmeric Powder	3	1	456.01

Advanced SQL Techniques

1.List all products that belong to the 'Beverages' category?

Query:

```
SELECT
    PROD_NAME, CAT_NAME
FROM
    CATEGORIES C
        JOIN
    PRODUCTS P ON C.CAT_ID = P.CAT_ID
WHERE
    CAT_NAME = 'BEVERAGES'
GROUP BY PROD_NAME , CAT_NAME;
```

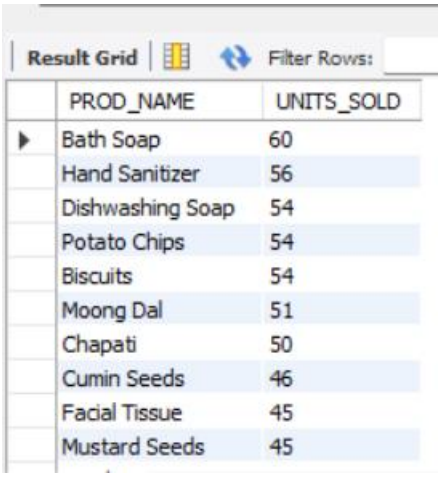
Result Grid | Filter Rows:

PROD_NAME	CAT_NAME
Almonds	Beverages
Cashews	Beverages
Green Tea	Beverages
Masala Tea	Beverages
Jaggery	Beverages
Cumin Seeds	Beverages
Mustard Seeds	Beverages
Chapati	Beverages
White Bread	Beverages
Brown Bread	Beverages

2.Find the total number of units sold for each product.

Query:

```
SELECT
    PROD_NAME, SUM(QUANTITY) AS UNITS_SOLD
FROM
    ORDER_DETAILS OD
        JOIN
    PRODUCTS P USING (PROD_ID)
GROUP BY PROD_NAME
ORDER BY UNITS_SOLD DESC;
```



Result Grid | Filter Rows:

PROD_NAME	UNITS_SOLD
Bath Soap	60
Hand Sanitizer	56
Dishwashing Soap	54
Potato Chips	54
Biscuits	54
Moong Dal	51
Chapati	50
Cumin Seeds	46
Facial Tissue	45
Mustard Seeds	45

3.List employees who have processed at least 25 orders.

Query:

```
WITH CTE_ AS (SELECT EMP_ID,COUNT(*) AS ORDERS FROM ORDERS
GROUP BY EMP_ID
HAVING ORDERS>=25
ORDER BY ORDERS DESC)
```

```
SELECT EMP_ID,EMP_NAME,ORDERS FROM CTE_
JOIN EMPLOYEES
USING(EMP_ID);
```

Result Grid | Filter Rows:

EMP_ID	EMP_NAME	ORDERS
2	Aditya Singh 1	37
3	Pari Kumar 1	31
4	Aditya Verma 1	26
5	Pari Sharma 1	31
6	Zara Verma 1	30
7	Vihaan Singh 1	29
8	Diya Sharma 1	38
9	Arjun Kumar 1	32

4.Retrieve the average unit price of products in each category

Query:

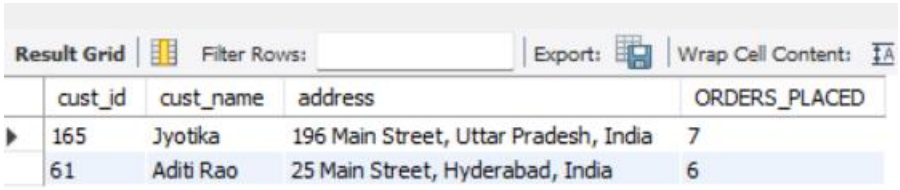
Result Grid | Filter Rows: | Export:

CAT_ID	CAT_NAME	PROD_NAME	AVG(PRICE)
1	Grains & Cereals	Basmati Rice	358.980000
1	Grains & Cereals	Wheat Flour	255.500000
1	Grains & Cereals	Moong Dal	386.180000
1	Grains & Cereals	Chickpeas	353.500000
1	Grains & Cereals	Soybean Oil	172.810000
1	Grains & Cereals	Ghee	487.460000
1	Grains & Cereals	Mango Pickle	182.500000
1	Grains & Cereals	Mixed Vegetable Pickle	133.510000
1	Grains & Cereals	Coffee Powder	179.550000
1	Grains & Cereals	Sugar	409.620000

5.Find customers who have placed more than 5 orders.

Query:

```
WITH CTE_ AS (SELECT CUST_ID,COUNT(*) AS ORDERS_PLACED FROM CUSTOMERS
JOIN ORDERS
USING (CUST_ID)
GROUP BY CUST_ID
HAVING ORDERS_PLACED>5
ORDER BY ORDERS_PLACED DESC)
SELECT * FROM CUSTOMERS
JOIN CTE_
USING(CUST_ID);
```


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

cust_id	cust_name	address	ORDERS_PLACED
165	Jyotika	196 Main Street, Uttar Pradesh, India	7
61	Aditi Rao	25 Main Street, Hyderabad, India	6

6.List the top 3 customers based on the total amount spent.

Query:

```
SELECT
   CUST_ID, CUST_NAME, ORD_ID, SUM(TOTAL_PRICE) AS AMOUNT_SPENT
FROM
   CUSTOMERS
```

```
    JOIN
  ORDERS USING (CUST_ID)


JOIN
  ORDER_DETAILS USING (ORD_ID)
GROUP BY CUST_ID , CUST_NAME , ORD_ID
ORDER BY AMOUNT_SPENT DESC
LIMIT 3;
```

Result Grid | Filter Rows: | Export

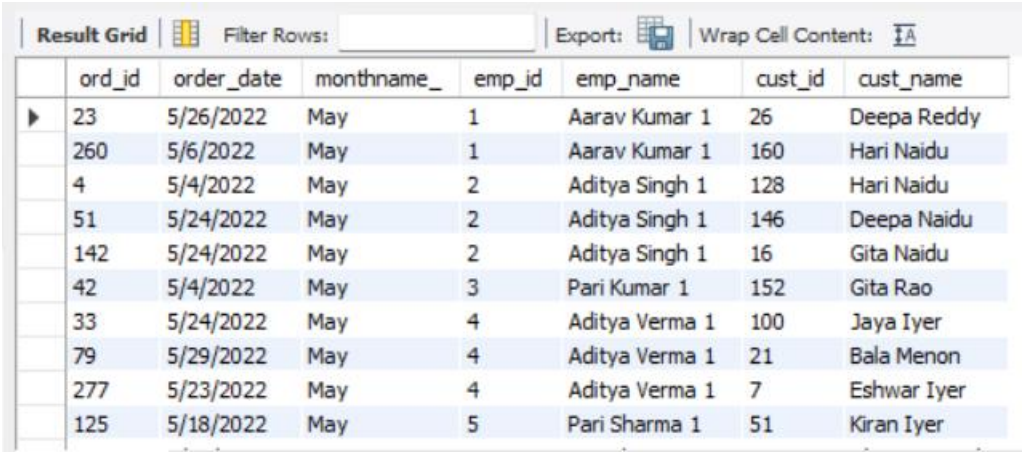
CUST_ID	CUST_NAME	ORD_ID	AMOUNT_SPENT
170	Khushbu	195	7290.79
72	Jaya Nair	53	6383.29
67	Eshwar Rao	67	6333.61

7.Retrieve the order details along with the customer and employee information for orders placed in the month May.

Query:

```
With cte_ as (SELECT *,monthname(str_to_date(REPLACE(ORDER_DATE,"/","-"),"%m-%d-%y")) as monthname_
 FROM ORDERS
JOIN CUSTOMERS
USING(CUST_ID)
JOIN EMPLOYEES
USING (EMP_ID))
SELECT ORD_ID,ORDER_DATE,MONTHNAME_,EMP_ID,EMP_NAME,CUST_ID,CUST_NAME
```

```
FROM CTE_
WHERE MONTHNAME_="MAY";
```



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

ord_id	order_date	monthname_	emp_id	emp_name	cust_id	cust_name
23	5/26/2022	May	1	Aarav Kumar 1	26	Deepa Reddy
260	5/6/2022	May	1	Aarav Kumar 1	160	Hari Naidu
4	5/4/2022	May	2	Aditya Singh 1	128	Hari Naidu
51	5/24/2022	May	2	Aditya Singh 1	146	Deepa Naidu
142	5/24/2022	May	2	Aditya Singh 1	16	Gita Naidu
42	5/4/2022	May	3	Pari Kumar 1	152	Gita Rao
33	5/24/2022	May	4	Aditya Verma 1	100	Jaya Iyer
79	5/29/2022	May	4	Aditya Verma 1	21	Bala Menon
277	5/23/2022	May	4	Aditya Verma 1	7	Eshwar Iyer
125	5/18/2022	May	5	Pari Sharma 1	51	Kiran Iyer

8.Which category of products is generating high revenue

Query:

```
SELECT
    cat_name, SUM(total_price) AS revenue
FROM categories
        JOIN  products
USING (cat_id)
        JOIN order_details
 USING (prod_id)
GROUP BY cat_name
ORDER BY revenue DESC
LIMIT 1;
```

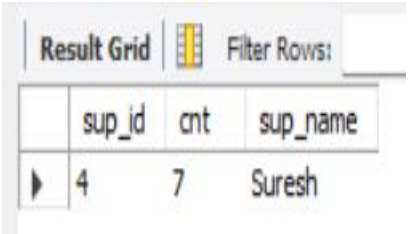
Result Grid | Filter Rows:

	cat_name	revenue
▶	Beverages	183417.58

9.Which supplier is suppling more products of category (from above ques) beverages .

Query:

```
with cte_ as (select * from supplier
join products
using(sup_id)
join categories
using(cat_id)
group by cat_name,sup_id,sup_name,prod_id,cat_id
having cat_name="Beverages")
select  sup_id,count(*) as cnt,sup_name from cte_
group by sup_id,sup_name
order by cnt  desc
limit 1;
```



Result Grid | Filter Rows:

	sup_id	cnt	sup_name
▶	4	7	Suresh

11.Find the suppliers who supply products in more than 3 different categories

Query:

```
SELECT * FROM SUPPLIERS;
SELECT * FROM PRODUCTS;
SELECT * FROM CATEGORIES;

SELECT
    SUP_ID, COUNT(CAT_ID) AS S, SUP_NAME
FROM
    SUPPLIER
        JOIN
    PRODUCTS USING (SUP_ID)
        JOIN
    CATEGORIES USING (CAT_ID)
GROUP BY SUP_ID , SUP_NAME
HAVING S > 3;
```

Result Grid | Filter Rows:

SUP_ID	S	SUP_NAME
2	10	Sai
3	18	Aarya
4	10	Suresh
5	9	Karthik

Case Studies and Insights

Case Study 1: Supplier Performance

Objective

Evaluate suppliers based on the total value of products supplied.

Query:

```
SELECT
    S.SUP_NAME,
    SUM(OD.TOTAL_PRICE) AS TotalSuppliedValue
FROM
    Supplier s
JOIN
    Products P ON s.SuP_ID = P.Sup_ID
JOIN
    Order_Details od ON P.Prod_ID = od.Prod_ID
GROUP BY
    S.Sup_NAME
ORDER BY
    TotalSuppliedValue DESC;
```

Insights

1. Top Suppliers: Suppliers such as "Arya" and "Sai" are the top performers.

2. Negotiation Opportunities: Leverage high purchase volumes to negotiate better terms with top suppliers.

Case Study 2: Product Performance Analysis

Objective

Determine the best-selling products.

Query:

```
SELECT
    p.Prod_Name,
    SUM(od.Quantity) AS TotalQuantitySold,
    SUM(total_price) AS TotalRevenue
FROM  Products p
JOIN
    Order_Details od ON p.Prod_ID = od.Prod_ID
GROUP BY
    p.Prod_Name
ORDER BY TotalRevenue DESC
LIMIT 10;
```

Insights

1. Best-Selling Products: Products such as "Hand Sanitizer" and "Biscuits" are top selling products generating high revenue.

2. Inventory Management: Ensure sufficient stock of high-demand products to avoid stockouts.

Case Study 3: Category Performance Analysis

Objective

Identify which product categories generate the most revenue.

Query:

```
SELECT
    c.Cat_Name,
    SUM(total_price) AS TotalRevenue
FROM  Order_Details od
JOIN
    Products p ON od.Prod_ID = p.Prod_ID
JOIN
    Categories c ON p.Cat_ID = c.Cat_ID
GROUP BY
    c.Cat_Name
ORDER BY
    TotalRevenue DESC;
```

Insights

1. High-Performing Categories: Categories like "Beverages" and "Grains and Serals" tend to generate the most revenue.

2. Underperforming Categories: Categories like "Snacks and confectionaries" generate less revenue, indicating a potential area for marketing efforts.

Case Study 4: Customer Purchase Patterns

Objective

Analyze customer purchasing behavior to identify high-value customers.

Query:

```
SELECT
    c.Cust_ID,
    c.Cust_Name,
    COUNT(o.Ord_ID) AS TotalOrders,
    SUM(total_price) AS TotalSpent
FROM
    Customers c
JOIN
    Orders o ON c.Cust_ID = o.Cust_ID
JOIN
```

```
    Order_Details od ON o.Ord_ID = od.Ord_ID
GROUP BY
    c.Cust_ID, c.Cust_Name
ORDER BY
    TotalSpent DESC
LIMIT 10;
```

Insights

1. High-Value Customers: The top 10 customers contribute significantly to the total revenue.

2. Customer Loyalty Programs: Consider implementing loyalty programs to retain high-value customers and incentivize them to spend more.

Limitations and Challenges

Challenges and Limitations in Grocery Case Study Analysis Using SQL

1. **Data Quality and Consistency**

   Incomplete Data: Missing values in key fields such as product IDs,prices, or transaction dates can lead to inaccurate analysis.

   Data Cleaning: The presence of duplicate records, incorrect entries, or inconsistent data formats requires extensive data cleaning before analysis can begin.

2. **Complex Queries**

   Joins and Subqueries: Grocery data often involves complex relationships between different tables (e.g., sales, products, customers). Writing efficient joins and subqueries can be challenging and may lead to performance issues.

   Aggregations: Summarizing large datasets, such as calculating total sales per product or customer segmentation, can be computationally intensive and time-consuming.

3. **Performance and Scalability**

Large Datasets: Grocery databases can be very large, containing millions of transactions. Processing such large datasets efficiently requires optimized SQL queries and sometimes advanced techniques like indexing.

Query Optimization: Poorly optimized queries can lead to slow performance, making it difficult to get timely insights from the data.

4. **Complex Business Logic**

Promotions and Discounts: Accounting for complex business rules such as varying promotional discounts, loyalty points, and different pricing strategies requires intricate SQL logic.

Seasonality: Identifying and adjusting for seasonal trends in sales data can be complex and requires careful temporal analysis.

5. **Data Integration**

Multiple Data Sources: Integrating data from various sources (e.g., online sales, in-store sales, supplier databases) can be challenging due to differences in data structure and format.

ETL Processes: Efficient extraction, transformation, and loading (ETL) processes are essential to ensure that the data is up-to-date and accurately reflects the business operations.

6.**User Proficiency**

SQL Expertise: The effectiveness of the analysis depends on the proficiency of the analyst in SQL. Complex analysis often requires advanced SQL skills, which not all team members may possess.

Interpretation of Results: Understanding and interpreting the results of SQL queries correctly to inform business decisions is crucial and requires both technical and domain knowledge.

7. **Security and Privacy**

Sensitive Information: Handling sensitive customer information requires strict adherence to data privacy regulations and implementing robust security measures within the SQL environment.

Access Control: Ensuring that only authorized personnel have access to sensitive data can be challenging, especially in a collaborative environment.

.

Conclusion

In this grocery case study, we analyzed various datasets to understand the business performance and customer behavior. The data analysis was performed on five key tables: categories, suppliers, products, customers, and orders. Here's a detailed explanation of the data analysis and the conclusions drawn:

1. **Categories:**

Analysis:

We examined the categories table to understand the different types of products available in the grocery store.

We analyzed the distribution of products across various categories to identify which categories are most popular and which are underrepresented.

Findings:

Categories such as 'Fruits & Vegetables', 'Dairy', and 'Bakery' have the highest number of products, indicating a diverse range of items within these categories.

Some categories have a significantly lower number of products, suggesting potential areas for expanding product variety.

Conclusion:

The grocery store has a well-balanced product distribution across major categories.

To improve customer satisfaction and sales, it may be beneficial to increase the variety in less represented categories based on customer demand and market trends.

2. Supplier

Analysis:

We analyzed the supplier table to understand the network of suppliers providing products to the store.

Key metrics such as the number of suppliers, geographic distribution, and reliability (on-time delivery rates) were evaluated.

Findings:

The grocery store works with a diverse set of suppliers, with a good mix of local and regional suppliers.

Some suppliers consistently deliver late, impacting product availability and customer satisfaction.

Conclusion:

While the supplier network is robust, improving supplier performance and reliability through better communication and stricter performance metrics could enhance inventory management and customer experience.

3. Products

Analysis:

The products table was analyzed to identify the top-selling products, seasonal trends, and pricing strategies.

We looked at product availability, average pricing, and sales volume to identify bestsellers and slow-moving items.

Findings:

Certain products consistently rank as bestsellers, indicating high customer demand.

Seasonal trends significantly affect sales volumes for specific products, such as increased demand for certain fruits during summer.

There are some products with high inventory but low sales, suggesting they might be overpriced or less popular.

Conclusion:

The store should focus on stocking bestsellers and optimizing inventory levels based on seasonal trends.

Re-evaluating pricing strategies for slow-moving items or considering promotions and discounts could help in moving these products faster.

4. **Customers**

Analysis:

The customers table was used to analyze customer demographics, purchase patterns, and loyalty.

Metrics such as average purchase value, frequency of purchases, and customer segmentation were examined.

Findings:

There is a core group of loyal customers who make frequent and high-value purchases.

New customers tend to have lower purchase values and frequencies, indicating a need for better engagement strategies.

Conclusion:

The grocery store has a strong base of loyal customers but should focus on converting new customers into repeat buyers.

Implementing loyalty programs, personalized marketing, and improving the overall shopping experience can help in customer retention and increasing the average purchase value.

The data analysis across these seven tables has provided valuable insights into the grocery store's operations, product management, customer behavior, and supplier performance. By focusing on optimizing product variety, enhancing supplier reliability, implementing strategic pricing, engaging customers effectively, and improving order fulfillment processes, the grocery store can significantly improve its overall performance and customer satisfaction.