

# Grocery Store Management

An SQL - Based Data Analysis for Business Insights



# Meet the Analyst



**Raghavendra Reddy T**

Data Analyst Trainee at Innomatics Research Labs

- B.Tech in Electronics and Communication Engineering
- Passionate about data-driven insights
- Solving real-world problems with analytics

[LinkedIn](#) | [GitHub](#)

# Agenda

## 1 Project Objectives & Domain Overview

Understanding the goal

## 2 Database Design: ER Diagram & Schema

Visual schema structure and table relationships

## 3 Key Analysis Questions

Key business queries

## 4 SQL Analysis: Queries & Results

SQL Insights

## 5 Business Insights & Recommendations

Strategic Actions

## 6 Conclusion: Overall Key Finding

Key Takeaway

## 7 Experience & Challenges

Lessons learned

## 8 Q&A

Your questions

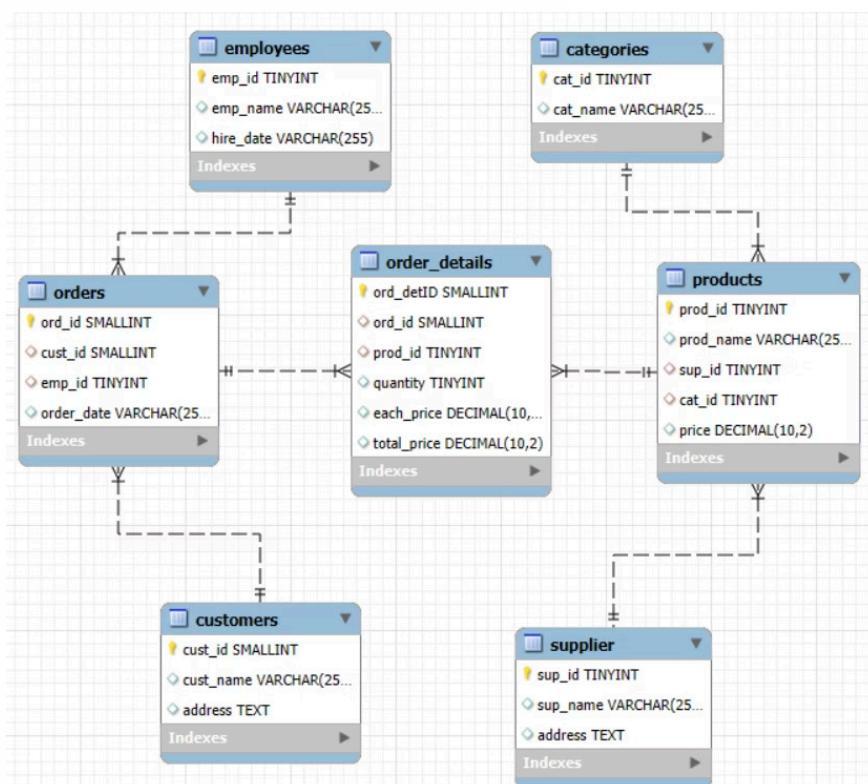
# Project Objective

- **Design & Implement a Robust Database:**
  - To develop a well-structured relational database schema capable of effectively managing grocery store data (products, sales, customers, etc.).
- **Master SQL for Data Analysis:**
  - To proficiently use SQL for complex data retrieval, manipulation, and in-depth analysis of sales patterns and customer behavior.
- **Uncover Actionable Business Insights:**
  - To identify key performance indicators such as top-selling products, most valuable customers, and significant revenue trends.
- **Apply & Demonstrate Technical Skills:**
  - To practically apply and showcase competence in essential SQL techniques, including joins, aggregations, subqueries, and data filtering.

# Understanding Our Domain: Retail & Grocery

- **Core Operations in Retail & Grocery:**
  - This industry fundamentally revolves around managing key areas like:
    - **Inventory:** Tracking product stock levels.
    - **Suppliers:** Managing relationships and procurement.
    - **Customer Orders:** Processing sales transactions.
    - **Employees:** Handling sales and operations.
    - **Product Categories:** Organizing diverse product ranges.
- **The Critical Role of Data:**
  - In today's competitive market, effective data management is not just beneficial, it's essential. It empowers businesses to:
    - Accurately **Track Sales & Revenue.**
    - Proactively **Monitor Product Availability.**
    - Gain insights by **Analyzing Customer & Employee Activity.**
    - Continuously **Improve Operational Efficiency.**
- **Our Project's Connection:**
  - This "Grocery Store Management" database project simulates these real-world data challenges, providing a practical platform to apply data analysis techniques relevant to this vital sector.

# Database Design: The Blueprint of Our Grocery Store



- **Entities & Relationships:**
  - Our database models **7 key entities** crucial for grocery operations.
  - Relationships (One-to-Many, Many-to-Many via junction tables) ensure data integrity and accurate linkage.
- **Core Tables & Their Roles:**
  - **customers:** Stores customer profiles.
  - **products:** Manages product catalog, pricing, and links to categories & suppliers.
  - **categories:** Defines product groupings (e.g., Beverages, Snacks).
  - **supplier:** Tracks information about product providers.
  - **employees:** Records staff details.
  - **orders:** Captures individual customer order information.
  - **order\_details:** The crucial link! Details each product within an order (quantity, price).
- **Integrity & Structure:**
  - **Primary Keys** (e.g., `prod_id` in `products`) uniquely identify each record.
  - **Foreign Keys** (e.g., `cust_id` in `orders`) enforce referential integrity, linking tables logically.
  - Data types chosen for efficiency and accuracy (e.g., `DECIMAL` for prices).

# Key Analysis Questions (Use Cases)

Our SQL analysis explored several key use cases to extract insights from the grocery store data:

## 1. Customer Insights

- Gain an understanding of customer engagement and purchasing behavior.

## 2. Product Performance

- Evaluate how well products are performing in terms of sales and revenue.

## 3. Sales and Order Trends

- Analyze business performance through orders and revenue over time.

## 4. Supplier Contribution

- Identify the most active and profitable suppliers.

## 5. Employee Performance

- Assess how employees are handling and influencing sales.

## 6. Order Details Deep Dive

- Explore item-level sales patterns and pricing behavior.

# SQL Analysis: Queries & Results

## Customer Insights:

- **Question 1.1:** How many unique customers have placed orders?

## SQL Query:

```
select  
    count(distinct (cust_id)) as unique_customers  
from orders;
```

## Result:

	unique_customers
▶	156

- **Observation:**

- Our analysis shows that **156 unique customers** have placed one or more orders.

## Customer Insights:

- **Question 1.2:** Which customers have placed the highest number of orders?

## SQL Query:

```
select cust_id,cust_name,  
       count(ord_id) as no_of_orders  
from customers  
inner join orders using(cust_id)  
group by cust_id,cust_name  
order by no_of_orders desc  
limit 5;
```

## Result:

cust_id	cust_name	no_of_orders
165	Jyotika	7
61	Aditi Rao	6
19	Chetan Naidu	5
128	Hari Naidu	5
145	Chetan Rao	5

## Customer Insights:

- **Question 1.3:** What is the total and average purchase value per customer?

### SQL Query:

```
SELECT cust_id, cust_name,
       sum(order_total_for_one_order) as total_purchase_value_by_customer,
       round(avg(order_total_for_one_order), 2) as average_value_of_this_customers_orders
  from customers
 inner join orders using(cust_id)
 inner join( select ord_id,
                  sum(total_price) as order_total_for_one_order
             from order_details
            group by ord_id
        ) as order_level_totals using(ord_id)
 group by cust_id, cust_name
 order by total_purchase_value_by_customer desc;
```

### Result:

cust_id	cust_name	total_purchase_value_by_customer	average_value_of_this_customers_orders
19	Chetan Naidu	11256.82	2814.21
166	Kapila	11099.51	2774.88
67	Eshwar Rao	10819.96	3606.65
61	Aditi Rao	10230.64	2557.66
7	Eshwar Iyer	9188.45	2297.11
160	Hari Naidu	8686.88	2895.63
188	Preeti Malhotra	8496.65	4248.33
195	Amit Saxena	8266.34	2066.59
165	Jyotika	8228.92	1175.56
174	Komal	8182.60	2727.53
8	Deepa Reddy	7929.13	2643.04
114	Isha Shetty	7838.15	2612.72
3	Chetan Rao	7693.41	1923.35
148	Chetan Iyer	7645.35	2548.45
186	Neha Joshi	7614.44	3807.22

## Customer Insights:

- **Question 1.4:** Who are top 5 customers by total purchase amount?

### SQL Query:

```
select cust_id,cust_name,  
       sum(total_price) as Total_Purchase_Amount  
from customers  
inner join orders using(cust_id)  
inner join order_details using(ord_id)  
group by cust_id,cust_name  
order by total_purchase_amount desc  
limit 5;
```

### Result:

cust_id	cust_name	Total_Purchase_Amount
19	Chetan Naidu	11256.82
166	Kapila	11099.51
67	Eshwar Rao	10819.96
61	Aditi Rao	10230.64
7	Eshwar Iyer	9188.45

## Product Performance:

- **Question 2.1:** How many products exist in each category?

### SQL Query:

```
select cat_id,cat_name,
       count(prod_id) as Total_Products
  from categories
 left join products
 using(cat_id)
 group by cat_id,cat_name
 order by total_products desc;
```

### Result:

cat_id	cat_name	Total_Products
1	Grains & Cereals	18
3	Beverages	17
2	Dairy Products	6
4	Personal Care	6
5	Snacks & Confectioneries	3

## Product Performance:

- **Question 2.2:** What is the average price of products by category?

### SQL Query:

```
select cat_id,cat_name,  
       round(avg(price), 2) as avg_price  
from categories  
inner join products  
using(cat_id)  
group by cat_id,cat_name  
order by avg_price desc;
```

### Result:

cat_id	cat_name	avg_price
2	Dairy Products	366.94
4	Personal Care	364.99
5	Snacks & Confectioneries	363.34
1	Grains & Cereals	287.67
3	Beverages	278.89

## Product Performance:

- **Question 2.3:** Which products have the highest total sales volume (by quantity)?

### SQL Query:

```
select prod_id,prod_name,  
       sum(quantity) as total_quantity_sold  
  from products  
inner join order_details using(prod_id)  
group by prod_id,prod_name  
order by total_quantity_sold desc  
limit 10;
```

### Result:

prod_id	prod_name	total_quantity_sold
32	Bath Soap	60
33	Hand Sanitizer	56
27	Dishwashing Soap	54
48	Biscuits	54
46	Potato Chips	54
3	Moong Dal	51
36	Chapati	50
21	Cumin Seeds	46
34	Facial Tissue	45
22	Mustard Seeds	45

## Product Performance:

- **Question 2.4:** What is the total revenue generated by each product?

### SQL Query:

```
select prod_id,prod_name,  
       sum(total_price) as Total_revenue  
from products  
inner join order_details using(prod_id)  
group by prod_id,prod_name  
order by Total_revenue desc;
```

### Result:

prod_id	prod_name	Total_revenue
33	Hand Sanitizer	27787.76
48	Biscuits	20995.92
3	Moong Dal	19695.02
31	Toothpaste	19688.95
22	Mustard Seeds	19516.68
12	Cashews	18561.92
40	Butter	18548.40
41	Cheese Slices	18519.61
19	Turmeric Powder	17784.29
44	Soya Sauce	16985.38

## Product Performance:

- **Question 2.5:** How do product sales vary by category and supplier?

### SQL Query:

```
select cat_name, sup_name,
       sum(quantity) as total_quantity_sold,
       sum(total_price) as total_revenue
  from order_details
 inner join products using(prod_id)
 inner join categories using(cat_id)
 inner join supplier using(sup_id)
 group by cat_name, sup_name
 order by total_revenue desc;
```

### Result:

cat_name	sup_name	total_quantity_sold	total_revenue
Personal Care	Aarya	205	69378.41
Grains & Cereals	Aarya	226	67701.10
Beverages	Aarya	196	65538.71
Beverages	Suresh	272	65307.14
Dairy Products	Sai	121	50740.60
Grains & Cereals	Karthik	140	39473.49
Beverages	Aarav Sharma	82	26948.15
Grains & Cereals	Suresh	67	26248.89
Snacks & Confectioneries	Karthik	86	22767.59
Dairy Products	Aarya	42	18519.61
Grains & Cereals	Sai	75	18018.02
Beverages	Sai	70	17103.15
Snacks & Confectioneries	Sai	35	16776.90

## Sales and Order Trends:

- **Question 3.1:** How many orders have been placed in total?

### SQL Query:

```
select count(ord_id) as total_orders  
from orders;
```

### Result:

total_orders
300

## Sales and Order Trends:

- **Question 3.2:** What is the average value per order?

### SQL Query:

```
select round(avg(order_total),2) as average_order_value
from (
    select ord_id,
    sum(total_price) as order_total
    from order_details
    group by ord_id
) as order_details;
```

### Result:

average_order_value
2153.63

## Sales and Order Trends:

- **Question 3.3:** On which dates were the most orders placed?

### SQL Query:

```
select order_date,  
       count(ord_id) as order_count  
from orders  
group by order_date  
order by order_count desc  
limit 10;
```

### Result:

order_date	order_count
3/30/2022	4
9/10/2022	4
1/30/2022	3
5/24/2022	3
1/14/2022	3
12/21/2022	3

## Sales and Order Trends:

- **Question 3.4:** What are the monthly trends in order volume and revenue?

### SQL Query:

```
select substring_index(order_date, '/', -1) as year,
       month(str_to_date(order_date, '%m/%d/%Y')) as month_number,
       monthname(str_to_date(order_date, '%m/%d/%Y')) as month_name,
       count(ord_id) as monthly_order_count,
       sum(total_price) as monthly_revenue
  from orders
 inner join order_details using(ord_id)
 group by year, month_number, month_name
 order by cast(year as unsigned), month_number;
```

### Result:

year	month_number	month_name	monthly_order_count	monthly_revenue
2022	1	January	79	70312.45
2022	2	February	66	66929.42
2022	3	March	57	45977.16
2022	4	April	32	29118.54
2022	5	May	46	41305.62
2022	6	June	31	27378.69
2022	7	July	50	48674.66
2022	8	August	41	36045.01
2022	9	September	57	52626.61
2022	10	October	32	25917.32
2022	11	November	47	46141.33
2022	12	December	62	60903.12

## Sales and Order Trends:

- **Question 3.5:** How do order patterns vary across weekdays and weekends?

### SQL Query:

```
select
  case
    when dayname(str_to_date(order_date, '%m/%d/%Y')) in ('saturday', 'sunday') then 'weekend'
    else 'weekday'
  end as day_type,
  count(ord_id) as order_count
from orders
group by day_type;
```

### Result:

day_type	order_count
weekend	90
weekday	210

## Supplier Contribution:

- **Question 4.1:** How many suppliers are there in the database?

### SQL Query:

```
select count(sup_id) as Total_Suppliers  
from supplier;
```

### Result:

Total_Suppliers
5

## Supplier Contribution:

- **Question 4.2:** Which supplier provides the most products?

### SQL Query:

```
select sup_id,sup_name,  
       count(prod_id) as product_count  
from supplier  
inner join products using(sup_id)  
group by sup_id,sup_name  
order by product_count desc  
limit 5;
```

### Result:

sup_id	sup_name	product_count
3	Aarya	18
2	Sai	10
4	Suresh	10
5	Karthik	9
1	Aarav Sharma	3

## Supplier Contribution:

- **Question 4.3:** What is the average price of products from each supplier?

### SQL Query:

```
select sup_id,sup_name,  
       round(avg(price), 2)as avg_product_price  
from supplier  
inner join products using(sup_id)  
group by sup_id,sup_name  
order by avg_product_price desc;
```

### Result:

sup_id	sup_name	avg_product_price
2	Sai	342.67
3	Aarya	319.33
5	Karthik	288.23
4	Suresh	281.82
1	Aarav Sharma	271.37

## Supplier Contribution:

- **Question 4.4:** Which suppliers contribute the most to total product sales (by revenue)?

### SQL Query:

```
select sup_id,sup_name,  
       sum(total_price) as total_revenue  
from supplier  
inner join products using(sup_id)  
inner join order_details using(prod_id)  
group by sup_id,sup_name  
order by total_revenue desc;
```

### Result:

sup_id	sup_name	total_revenue
3	Aarya	221137.83
2	Sai	113588.51
4	Suresh	101688.78
5	Karthik	81861.96
1	Aarav Sharma	33052.85

## Employee Performance:

- **Question 5.1:** How many employees have processed orders?

### SQL Query:

```
select count(distinct emp_id) as employees_processing_orders  
from orders;
```

### Result:

employees_processed_orders
10

## Employee Performance:

- Question 5.2: Which employees have handled the most orders?

### SQL Query:

```
select emp_id,emp_name,  
       count(ord_id) as orders_handled  
from employees  
inner join orders using(emp_id)  
group by emp_id,emp_name  
order by orders_handled desc  
limit 5;
```

### Result:

emp_id	emp_name	orders_handled
8	Diya Sharma 1	38
2	Aditya Singh 1	37
9	Arjun Kumar 1	32
3	Pari Kumar 1	31
5	Pari Sharma 1	31

## Employee Performance:

- **Question 5.3:** What is the total sales value processed by each employee?

### SQL Query:

```
select emp_id,emp_name,  
       sum(total_price) as total_sales_processed  
from employees  
inner join orders using(emp_id)  
inner join order_details using(ord_id)  
group by emp_id,emp_name  
order by total_sales_processed desc;
```

### Result:

emp_id	emp_name	total_sales_processed
2	Aditya Singh 1	79252.29
6	Zara Verma 1	71562.76
8	Diya Sharma 1	67241.85
3	Pari Kumar 1	66818.39
9	Arjun Kumar 1	54018.31
1	Aarav Kumar 1	52602.88
7	Vihaan Singh 1	48577.88
5	Pari Sharma 1	40334.22

## Employee Performance:

- **Question 5.4:** What is the average order value handled per employee?

### SQL Query:

```
select emp_name, round(avg(order_total), 2) as average_order_value_handled
from (
    select emp_name, ord_id, sum(total_price) as order_total
    from employees
    inner join orders using(emp_id)
    inner join order_details using(ord_id)
    group by emp_name,ord_id
) as employee_order_summary
group by emp_name
order by average_order_value_handled desc;
```

### Result:

emp_name	average_order_value_handled
Aarav Kumar 1	2768.57
Zara Verma 1	2650.47
Aditya Singh 1	2330.95
Pari Kumar 1	2227.28
Vihaan Singh 1	2112.08
Arjun Kumar 1	2077.63
Diya Sharma 1	2037.63
Arjun Verma 1	1835.84
Pari Sharma 1	1833.37
Aditya Verma 1	1554.75

## Order Details Deep Dive:

- **Question 6.1:** What is the relationship between quantity ordered and total price?

### SQL Query:

```
select quantity,  
       avg(total_price) as avg_total_price_of_quantity,  
       avg(each_price) as avg_unit_price_of_quantity  
from order_details  
group by quantity  
order by quantity;
```

### Result:

quantity	avg_total_price_of_quantity	avg_unit_price_of_quantity
1	319.274516	319.274516
2	595.716667	297.857949
3	898.266377	299.422029
4	1338.607143	334.651905
5	1530.401121	306.079655

## Order Details Deep Dive:

- **Question 6.2:** What is the average quantity ordered per product?

### SQL Query:

```
select prod_id,prod_name,  
       round(avg(quantity), 2) as avg_quantity_ordered_per_product  
from products  
inner join order_details using(prod_id)  
group by prod_id,prod_name  
order by avg_quantity_ordered_per_product desc;
```

### Result:

prod_id	prod_name	avg_quantity_ordered_per_product
40	Butter	4.56
31	Toothpaste	3.67
46	Potato Chips	3.60
42	Tomato Ketchup	3.50
22	Mustard Seeds	3.46
3	Moong Dal	3.40
6	Ghee	3.38
9	Mango Pickle	3.36
32	Bath Soap	3.33
45	Chili Sauce	3.25
18	Salt	3.25
34	Facial Tissue	3.21

## Order Details Deep Dive:

- **Question 6.3:** How does the unit price vary across products and orders?

### SQL Query:

```
select prod_id, prod_name,
       count(distinct each_price) as number_of_distinct_prices,
       min(each_price) as min_unit_price,
       max(each_price) as max_unit_price,
       round(avg(each_price), 2) as avg_unit_price,
       count(ord_detID) as times_sold
  from products
 inner join order_details using(prod_id)
 group by prod_id, prod_name
 order by times_sold desc;
```

### Result:

prod_id	prod_name	number_of_distinct_prices	min_unit_price	min_unit_price	avg_unit_price	times_sold
33	Hand Sanitizer	1	496.21	496.21	496.21	19
32	Bath Soap	1	235.22	235.22	235.22	18
27	Dishwashing Soap	1	146.65	146.65	146.65	18
48	Biscuits	1	388.81	388.81	388.81	17
43	Mayonnaise	1	258.15	258.15	258.15	16
12	Cashews	1	441.95	441.95	441.95	16
36	Chapati	1	201.68	201.68	201.68	16
2	Wheat Flour	1	255.50	255.50	255.50	15
3	Moong Dal	1	386.18	386.18	386.18	15
21	Cumin Seeds	1	182.74	182.74	182.74	15
46	Potato Chips	1	169.93	169.93	169.93	15
41	Cheese Slices	1	440.94	440.94	440.94	15
37	White Bread	1	436.22	436.22	436.22	15
28	Toilet Cleaner	1	479.34	479.34	479.34	15
44	Soya Sauce	1	499.57	499.57	499.57	14
38	Brown Bread	1	346.14	346.14	346.14	14