



INNOMATICS[®]
RESEARCH LABS

INNOVATION. AUTOMATION. ANALYTICS

PROJECT ON
Grocery Store
Management

Presented By:

K. Bhargavi

K. Shilpa

S. Lakshmi

Grocery Store Management

AGENDA

- Introduction
- Objectives of the Project
- Problem Statement
- Tables
- ER Diagram and Schema explanation
- SQL query results with screenshots
- Key Insights
- Recommendations
- Conclusion



Introduction

- The project focuses on the Retail and Grocery domain, emphasizing efficient management of inventory, suppliers, customers, and employees.
- It aims to build a mini grocery store database that reflects real-world business operations and data relationships.
- Interconnected tables such as Products, Orders, Suppliers, and Employees are used to simulate actual workflows.
- Students use SQL queries to extract, transform, and analyse data for meaningful insights.
- The project enhances understanding of database design, normalization, and relational integrity.
- Through analysis, learners can evaluate sales performance, stock availability, and supplier efficiency.
- The overall goal is to improve decision-making and operational efficiency through data-driven insights.



OBJECTIVES

- To design a centralized SQL database for storing and managing all grocery store information.
- To simplify customer, supplier, employee, and product management.
- To enable easy tracking of orders, sales, and transactions.
- To provide accurate inventory management and avoid stock-out/overstock issues.
- To improve data accuracy, reliability, and security compared to manual record-keeping.
- To generate useful reports and insights for better business decision-making.
- To save time, reduce manual errors, and ensure efficient store operations.



Problem Statement

- Grocery stores generate huge volumes of daily data from customers, products, suppliers, and sales.
- Without a centralized system, it's hard to identify best-selling items, loyal customers, or supplier performance.
- Manual tracking leads to errors, inefficiency, and poor decision-making.
- There's a need for an SQL-driven database system to organize, analyze, and visualize store data efficiently.



Tables

Products Table

Field	Type	Null	Key	Default
prod_id	tinyint	NO	PRI	NULL
prod_name	varchar(255)	YES		NULL
sup_id	tinyint	YES	MUL	NULL
cat_id	tinyint	YES	MUL	NULL
price	decimal(10,2)	YES		NULL

Suppliers Table

Field	Type	Null	Key	Default
sup_id	tinyint	NO	PRI	NULL
sup_name	varchar(255)	YES		NULL
address	text	YES		NULL

Customers Table

Field	Type	Null	Key	Default
cust_id	smallint	NO	PRI	NULL
cust_name	varchar(255)	YES		NULL
address	text	YES		NULL

Categories Table

Field	Type	Null	Key	Default
cat_id	tinyint	NO	PRI	NULL
cat_name	varchar(255)	YES		NULL

Tables

Employees Table

Field	Type	Null	Key	Default
emp_id	tinyint	NO	PRI	NULL
emp_name	varchar(255)	YES		NULL
hire_date	varchar(255)	YES		NULL

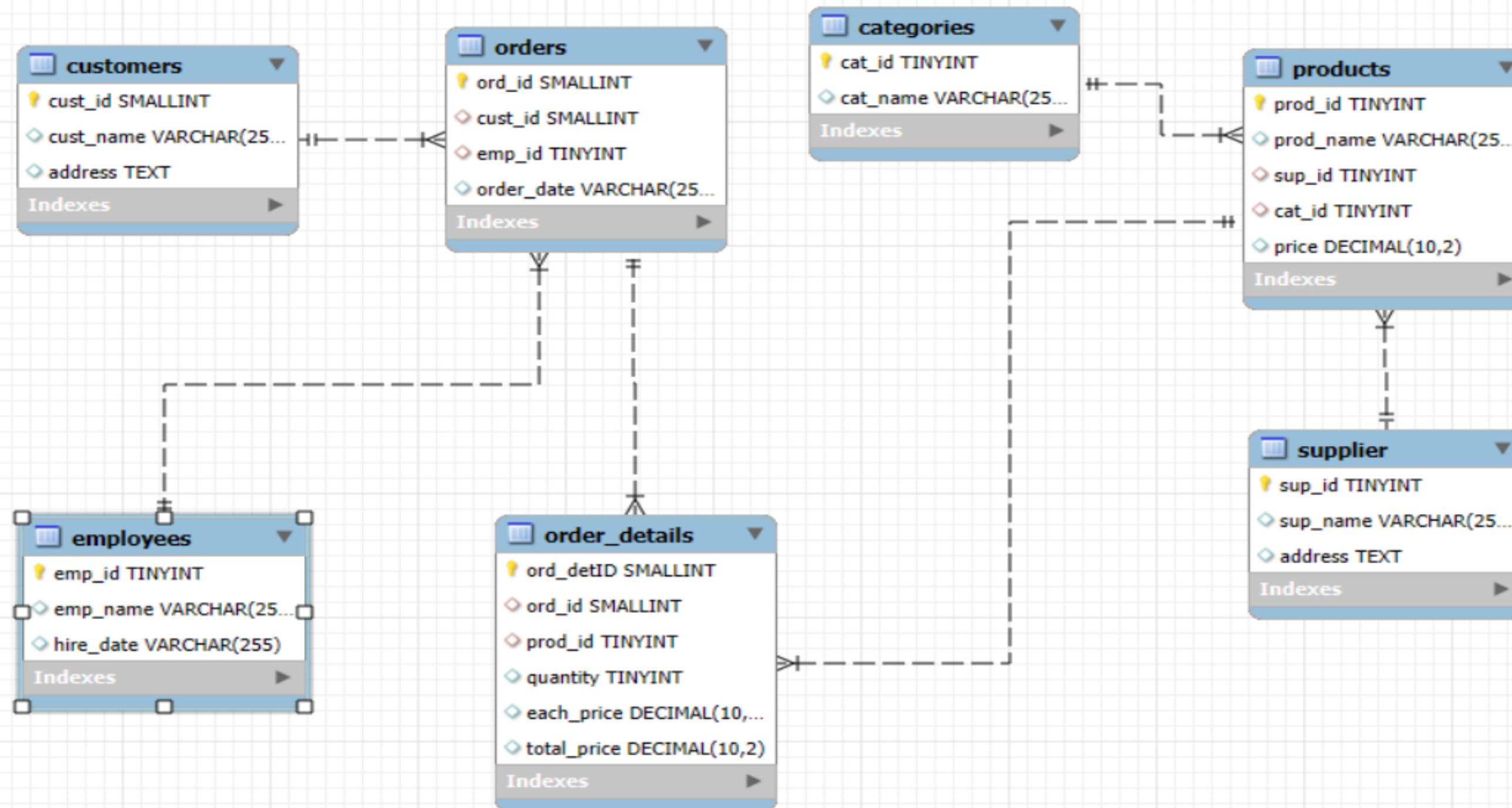
Orders Table

Field	Type	Null	Key	Default
ord_id	smallint	NO	PRI	NULL
cust_id	smallint	YES	MUL	NULL
emp_id	tinyint	YES	MUL	NULL
order_date	varchar(255)	YES		NULL

Order_details Table

Field	Type	Null	Key	Default	Extra
ord_detID	smallint	NO	PRI	NULL	auto_increment
ord_id	smallint	YES	MUL	NULL	
prod_id	tinyint	YES	MUL	NULL	
quantity	tinyint	YES		NULL	
each_price	decimal(10,2)	YES		NULL	
total_price	decimal(10,2)	YES		NULL	

ER Diagram



1. How many unique customers have placed orders?

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

	unique_customers
▶	156

2. Which customers have placed the highest number of orders?

```
select c.cust_id, c.cust_name, count(o.cust_id) as no_of_orders
from customers c
join orders o on c.cust_id=o.cust_id
group by c.cust_id
order by no_of_orders desc
limit 1;
```



cust_id	cust_name	no_of_orders
165	Jyotika	7

3. What is the total and average purchase value per customer?

```
select o.cust_id, c.cust_name,
sum(od.total_price) as total_purchase,
avg(od.total_price) as avg_purchase
from orders o
join order_details od on o.ord_id=od.ord_id
join customers c on o.cust_id=c.cust_id
group by o.cust_id;
```

cust_id	cust_name	total_purchase	avg_purchase
158	Eshwar Menon	3061.90	765.475000
129	Kiran Pillai	2625.93	656.482500
27	Chetan Gowda	5750.59	821.512857
122	Chetan Reddy	3869.54	1289.846667
168	Kasturi	3865.31	1288.436667
157	Deepa Gowda	1745.11	872.555000
125	Gita Nair	6305.09	1261.018000
167	Karishma	5426.90	493.354545
166	Kapila	11099.51	1109.951000
120	Kiran Iyer	5588.91	698.613750
145	Chetan Rao	5351.18	668.897500
141	Hari Nair	5722.71	817.530000
182	Nikita	5595.89	932.648333
163	Esha	2515.25	1257.625000
94	Gita Menon	6581.93	731.325556
21	Bala Menon	380.72	190.360000
113	Chetan Nair	4044.01	577.715714
56	Eshwar Rao	5726.35	954.391667
185	Girish Gupta	4778.52	1194.630000
31	Hari Rao	6624.78	662.478000
106	Gita Menon	5504.08	1100.816000
8	Deepa Reddy	7929.13	881.014444

4. Who are the top 5 customers by total purchase amount?

```
select o.cust_id,c.cust_name,  
sum(od.total_price) as total_purchase  
from orders o  
join order_details od on o.ord_id=od.ord_id  
join customers c on o.cust_id=c.cust_id  
group by o.cust_id  
order by total_purchase desc  
limit 5;
```

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

5. How many products exist in each category?

```
select c.cat_id,c.cat_name,count(p.prod_id)
from categories c
left join products p
on c.cat_id=p.cat_id
group by cat_id;
```



cat_id	cat_name	count(p.prod_id)
1	Grains & Cereals	18
2	Dairy Products	6
3	Snacks & Confectioneries	17
4	Personal Care	6
5	Household	3

6. What is the average price of products by category?

```
select c.cat_name, avg(p.price)
from categories c
join products p
on c.cat_id=p.cat_id
group by cat_name;
```

cat_name	avg(p.price)
Grains & Cereals	287.673333
Dairy Products	366.943333
Snacks & Confectioneries	278.892353
Personal Care	364.991667
Household	363.336667

7. Which products have the highest total sales volume (by quantity)?

```
select p.prod_name,sum(od.quantity) as total_high_sales
from products p
join order_details od on p.prod_id=od.prod_id
group by prod_name
order by total_high_sales desc
limit 1;
```

prod_name	total_high_sales
Bath Soap	60

8. What is the total revenue generated by each product?

```
SELECT
    SUM(o.total_price) AS total_revenue, p.prod_name
FROM
    order_details o
    LEFT JOIN
    products p ON o.prod_id = p.prod_id
GROUP BY prod_name;
```

total_revenue	prod_name
5062.45	Black Pepper
18561.92	Cashews
5819.23	Green Tea
5703.43	Salt
19695.02	Moong Dal
7918.87	Dishwashing Soap
14848.72	Detergent Powder
11283.92	Tomato Ketchup
8406.10	Cumin Seeds
14113.00	Bath Soap
11100.45	Mayonnaise
6104.70	Coffee Powder
13161.31	Ghee
10084.19	Chapati
8309.02	Cinnamon Sticks
20995.92	Biscuits
10949.84	Conditioner
16776.90	Toilet Cleaner
9709.00	Wheat Flour
3110.56	Soybean Oil
6752.59	Mango Pickle
6208.20	Cardamom

9. How do product sales vary by category and supplier?

```
select c.cat_name,s.sup_name,sum(o.total_price)
as total_sales
from
    order_details o
join
    products p on o.prod_id=p.prod_id
join
    categories c on p.cat_id=c.cat_id
join
    supplier s on p.sup_id=s.sup_id
group by c.cat_name,s.sup_name;
```

cat_name	sup_name	total_sales
Grains & Cereals	Aarya	67701.10
Grains & Cereals	Sai	18018.02
Grains & Cereals	Suresh	26248.89
Grains & Cereals	Karthik	39473.49
Grains & Cereals	Aarav Sharma	6104.70
Dairy Products	Sai	50740.60
Dairy Products	Aarya	18519.61
Dairy Products	Karthik	11100.45
Snacks & Confectioneries	Karthik	8520.43
Snacks & Confectioneries	Suresh	65307.14
Snacks & Confectioneries	Sai	17103.15
Snacks & Confectioneries	Aarya	65538.71
Snacks & Confectioneries	Aarav Sharma	26948.15
Personal Care	Suresh	10132.75
Personal Care	Sai	10949.84
Personal Care	Aarya	69378.41
Household	Karthik	22767.59
Household	Sai	16776.90

10. How many orders have been placed in total?

```
SELECT  
  
    COUNT(*) AS total_orders  
  
FROM  
  
    orders;
```

total_orders

300

11. What is the average value per order?

```
SELECT  
    SUM(total_price) / COUNT(DISTINCT ord_id) AS avg_order_value  
FROM  
    order_details;
```

avg_order_value

2153.632539

12. On which dates were the most orders placed?

```
SELECT
    order_date, COUNT(ord_id) AS orders_count
FROM
    orders
GROUP BY order_date
ORDER BY orders_count DESC
LIMIT 1;
```

order_date	orders_count
2022-09-10	4

13. How do order patterns vary across weekdays and months?

```

update orders
set order_date = date_format(str_to_date(order_date, '%Y-%m-%d'), '%m/%d/%Y');

select
  DAYNAME(STR_TO_DATE(order_date, '%m/%d/%Y')) as weekday,
  MONTHNAME(STR_TO_DATE(order_date, '%m/%d/%Y')) as month,
  COUNT(ord_id) as total_orders
from
  orders
group by weekday, month
order by weekday, month;
  
```

weekday	month	total_orders
Friday	April	4
Friday	August	2
Friday	December	5
Friday	February	2
Friday	January	8
Friday	July	5
Friday	June	2
Friday	March	6
Friday	May	5
Friday	November	6
Friday	October	1
Friday	September	6
Monday	April	1
Monday	August	4
Monday	December	7
Monday	February	4
Monday	January	5
Monday	June	3
Monday	March	2

14. How many suppliers are there in the database?

```
SELECT  
  
    COUNT(*)  
  
FROM  
  
    supplier;
```

```
COUNT(*)
```

```
5
```


15. Which supplier provides the most products?

```
SELECT
    s.sup_id, s.sup_name, COUNT(p.prod_id) AS product_count
FROM
    supplier s
    JOIN
    products p ON s.sup_id = p.sup_id
GROUP BY s.sup_id
ORDER BY product_count DESC
LIMIT 1;
```

sup_id	sup_name	product_count
3	Aarya	18


16. What is the average price of products from each supplier?

```
SELECT
    s.sup_id, s.sup_name, AVG(p.price)
FROM
    products p
    JOIN
    supplier s ON s.sup_id = p.sup_id
GROUP BY s.sup_id;
```

sup_id	sup_name	AVG(p.price)
1	Aarav Sharma	271.366667
2	Sai	342.672000
3	Aarya	319.326667
4	Suresh	281.818000
5	Karthik	288.225556

17. Which suppliers contribute the most to total product sales (by revenue)

```
SELECT
    s.sup_id, s.sup_name, SUM(od.total_price) AS total_revenue
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_id
ORDER BY total_revenue DESC
LIMIT 1;
```



	sup_id	sup_name	total_revenue
▶	3	Aarya	221137.83

18. How many employees have processed orders?

```
SELECT  
    COUNT(DISTINCT emp_id)  
FROM  
    orders;
```

	COUNT(DISTINCT emp_id)
▶	10

19. What is the total sales value processed by each employee?

SELECT

e.emp_id, e.emp_name, SUM(od.total_price) AS total_sales

FROM

employees e

JOIN

orders o ON e.emp_id = o.emp_id

JOIN

order_details od ON o.ord_id = od.ord_id

GROUP BY e.emp_id

ORDER BY total_sales DESC;

	emp_id	emp_name	total_sales
▶	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

20. What is the relationship between quantity ordered and total price?

```
SELECT
  quantity,
  AVG(od.total_price) AS avg_total_price,
  COUNT(*) AS order_count
FROM order_details od
GROUP BY quantity
ORDER BY quantity;
```



	quantity	avg_total_price	order_count
▶	1	319.274516	124
	2	595.716667	117
	3	898.266377	138
	4	1338.607143	105
	5	1530.401121	116

21. What is the average quantity ordered per product?

```
SELECT
    p.prod_id,
    p.prod_name,
    COUNT(od.ord_id) AS total_orders,
    AVG(od.quantity) AS avg_quantity_ordered
FROM
    products p
    JOIN
    order_details od ON p.prod_id = od.prod_id
GROUP BY p.prod_id
ORDER BY avg_quantity_ordered DESC;
```

	prod_id	prod_name	total_orders	avg_quantity_ordered
▶	40	Butter	9	4.5556
	31	Toothpaste	12	3.6667
	46	Potato Chips	15	3.6000
	42	Tomato Ketchup	10	3.5000
	22	Mustard Seeds	13	3.4615
	3	Moong Dal	15	3.4000
	6	Ghee	8	3.3750
	9	Mango Pickle	11	3.3636
	32	Bath Soap	18	3.3333
	45	Chili Sauce	8	3.2500
	18	Salt	8	3.2500
	34	Facial Tissue	14	3.2143
	1	Basmati Rice	10	3.2000
	26	Detergent Pow...	10	3.2000
	35	Mouth wash	10	3.2000
	16	Sugar	5	3.2000
	13	Green Tea	11	3.1818
	47	Chocolate Bar	11	3.1818
	49	Instant Noodles	11	3.1818

22.How does the unit price vary across products and orders?

```
SELECT
  p.prod_id,
  p.prod_name,
  od.each_price AS unit_price,
  COUNT(od.ord_id) AS times_ordered
FROM
  products p
  JOIN
  order_details od ON p.prod_id = od.prod_id
GROUP BY p.prod_id , od.each_price
ORDER BY p.prod_id , od.each_price;
```

	prod_id	prod_name	unit_price	times_ordered
▶	1	Basmati Rice	358.98	10
	2	Wheat Flour	255.50	15
	3	Moong Dal	386.18	15
	4	Chickpeas	353.50	7
	5	Soybean Oil	172.81	11
	6	Ghee	487.46	8
	7	Paneer	484.27	11
	8	Yogurt	111.61	7
	9	Mango Pickle	182.50	11
	10	Mixed Vegetable Pickle	133.51	7
	11	Almonds	315.57	9
	12	Cashews	441.95	16
	13	Green Tea	166.26	11
	14	Masala Tea	380.85	10
	15	Coffee Powder	179.55	14
	16	Sugar	409.62	5
	17	Jaggery	200.85	12

Key Insights

- **Customer Insights:** Top 5 customers contribute a large share of revenue — loyalty programs can help retain them.
- **Product Performance:** Certain categories (like Beverages or Dairy) generate the highest revenue and need more stock.
- **Sales & Order Trends:** Sales peak on weekends — marketing efforts should target those days.
- **Supplier Contribution:** A few suppliers account for most product deliveries — reliability should be ensured.
- **Employee Performance:** Some employees handle higher order volumes — training others can improve balance.



Recommendations

- **Enhance Customer Loyalty:** Launch reward programs and personalized offers for frequent buyers.
- **Boost Product Sales:** Promote top-selling items and bundle low-selling ones with popular products.
- **Optimize Inventory:** Maintain safety stock for high-demand products and reduce excess stock.
- **Strengthen Supplier Relationships:** Prioritize reliable suppliers and negotiate better contracts.
- **Data-Driven Decision Making:** Continue SQL-based analysis for monthly trend monitoring and performance tracking.



Conclusion

- ❖ The SQL-based Grocery Store Management System effectively stores and analyzes operational data.
- ❖ It improves efficiency, accuracy, and visibility across sales, suppliers, and inventory.
- ❖ Managers can make **data-backed decisions** to enhance productivity and profitability.
- ❖ With SQL queries and insights, the store can transform data into **actionable intelligence**.





Thank You