



**INNOMATICS<sup>®</sup>**  
RESEARCH LABS

**INNOVATION. AUTOMATION. ANALYTICS**

# Grocery Store Management

# GROCERY STORE MANAGEMENT



## Supplier Table

	Field	Type	Null	Key	Default	Extra
▶	SupplierID	int	YES		NULL	
	SupplierName	text	YES		NULL	
	Address	text	YES		NULL	

## Category Table

	Field	Type	Null	Key	Default	Extra
▶	CategoryID	int	YES		NULL	
	CategoryName	text	YES		NULL	

## Product Table

	Field	Type	Null	Key	Default	Extra
▶	ProductID	int	YES		NULL	
	Name	text	YES		NULL	
	SupplierID	int	YES		NULL	
	CategoryID	int	YES		NULL	
	Price	double	YES		NULL	

## Customers Table

	Field	Type	Null	Key	Default	Extra
▶	CustomerID	int	YES		NULL	
	Name	text	YES		NULL	
	Address	text	YES		NULL	

## Order\_ Details Tables

	Field	Type	Null	Key	Default	Extra
▶	OrderDetailID	int	YES		NULL	
	OrderID	int	YES		NULL	
	ProductID	int	YES		NULL	
	Quantity	int	YES		NULL	
	PriceEach	double	YES		NULL	
	TotalPrice	double	YES		NULL	

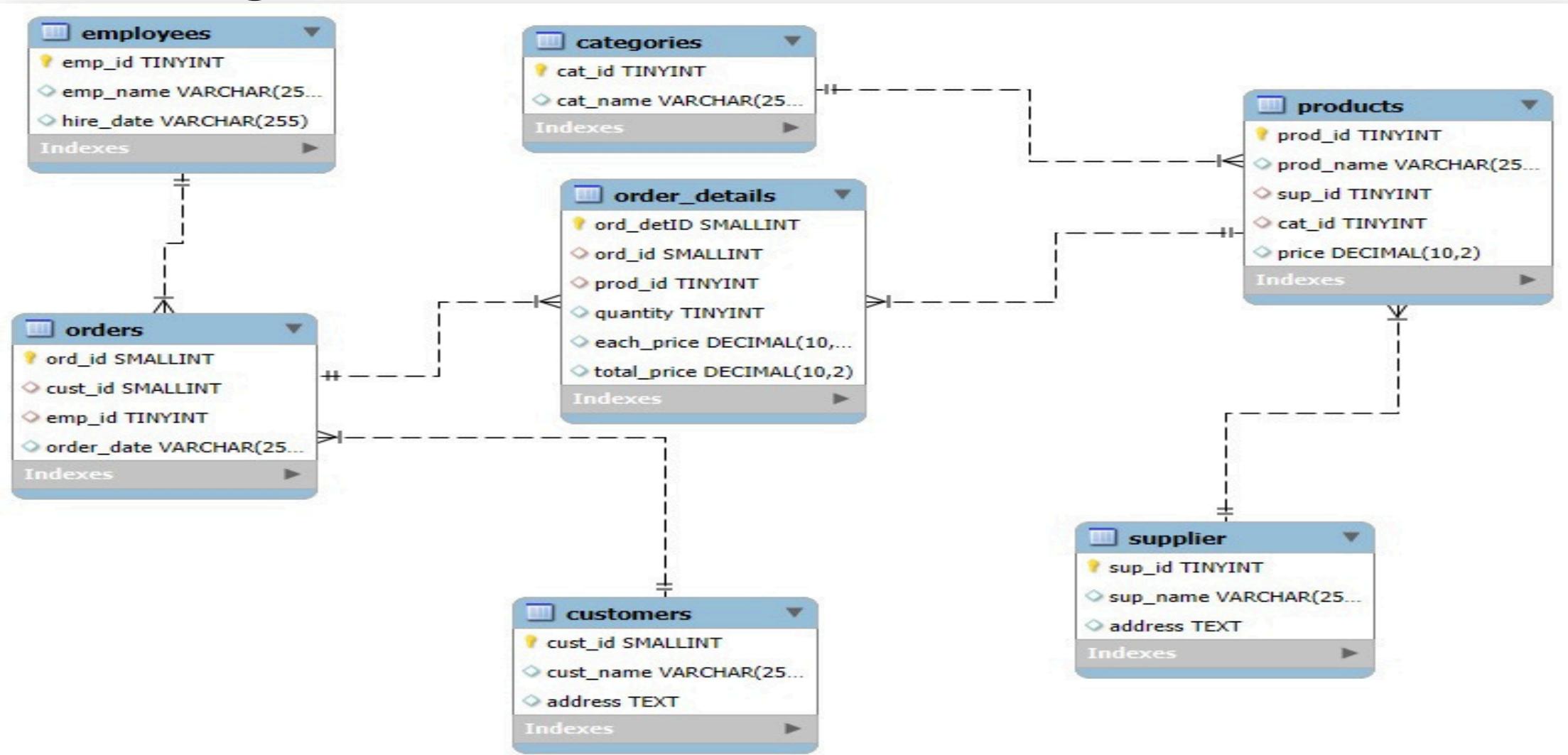
## ORDER TABLES

	Field	Type	Null	Key	Default	Extra
▶	OrderID	int	YES		NULL	
	CustomerID	int	YES		NULL	
	EmployeeID	int	YES		NULL	
	OrderDate	text	YES		NULL	

## STORE EMPLOYEE TABLES

	Field	Type	Null	Key	Default	Extra
▶	EmployeeID	int	YES		NULL	
	Name	text	YES		NULL	
	HireDate	text	YES		NULL	

# ER Diagram:



# **Problem Statement:**

Grocery stores typically manage large volumes of data related to products, suppliers, stock levels, customers, and sales transactions. When these operations are handled manually or using unstructured spreadsheets, it becomes difficult to maintain data accuracy, generate timely reports, prevent stockouts, or track sales performance. The lack of centralized data storage also makes it challenging to analyze trends, forecast inventory needs, and support decision-making.

Therefore, an SQL-based database management system is required to organize the data efficiently, automate queries, and provide accurate, real-time insights into store operations.

1. How many unique customers have placed orders?

```
SELECT COUNT(DISTINCT CustomerID) AS UniqueCustomers  
FROM Orders;
```



	UniqueCustomers
▶	156

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

```
select c.customerid,c.name,count(o.orderid) as orders_count  
from customers c  
join orders o on c.customerid = o.customerid  
group by c.customerid,c.name  
order by orders count desc;
```

	name	customerid	count_
▶	Jyotika	165	7

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

```
SELECT
    c.customerid,
    c.name,
    SUM(od.totalprice) AS total_purchase_value,
    AVG(od.totalprice) AS average_purchase_value
FROM customers c
JOIN orders o
    ON c.customerid = o.customerid
JOIN orderdetails od
    ON o.orderid = od.orderid
GROUP BY c.customerid, c.name
ORDER BY total_purchase_value DESC;
```

	customerid	name	total_purchase	avg_purchase
▶	158	Eshwar Menon	3061.897344749166	765.4743361872916
	129	Kiran Pillai	2625.9382572541513	656.4845643135378
	27	Chetan Gowda	5750.593464329533	821.5133520470762
	122	Chetan Reddy	3869.530789092514	1289.8435963641714
	168	Kasturi	3865.307758826527	1288.4359196088424
	157	Deepa Gowda	1745.1100486969347	872.5550243484673
	125	Gita Nair	6305.088429856438	1261.0176859712876
	167	Karishma	5426.897879755909	493.35435270508265
	166	Kapila	11099.520581048373	1109.9520581048373
	120	Kiran Iyer	5588.903230807855	698.6129038509819
	145	Chetan Rao	5351.166036762838	668.8957545953548
	141	Hari Nair	5722.723866764942	817.5319809664203
	182	Nikita	5595.893321985681	932.6488869976134
	163	Esha	2515.2526111682664	1257.6263055841332
	94	Gita Menon	6581.933499721035	731.3259444134484
	21	Bala Menon	380.7209657612169	190.36048288060846
	113	Chetan Nair	4044.0140586496755	577.7162940928108
	56	Eshwar Rao	5726.348044559587	954.3913407599312
	185	Cirish Guntas	4778.511640680705	1104.6770101722237

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

```
select c.customerid, c.name, sum(od.totalprice) as total
from customers c
join orders o on c.customerid = o.customerid
join orderdetails od on o.orderid = od.orderid
group by c.customerid, c.name
order by total_spent desc
limit 5;
```



	customerid	name	total_spent
▶	19	Chetan Naidu	11256.81340711334
	166	Kapila	11099.520581048373
	67	Eshwar Rao	10819.967618565575
	61	Aditi Rao	10230.647081383104
	7	Eshwar Iyer	9188.45721606837

# 5. How many products exist in each category?

categoryname	product_count
Grains & Cereals	18
Dairy Products	6
Snacks & Confectioneries	17
Personal Care	6
Household	3

```
select cat.categoryname, count(p.productid) as product_count  
from categories cat  
left join products p on cat.categoryid = p.categoryid  
group by cat.categoryname;
```

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

```
select cat.categoryname,avg(p.price) as avg_price  
from categories cat  
join products p on cat.categoryid = p.categoryid  
group by cat.categoryname;
```



alamy

	categoryname	avg_price
▶	Grains & Cereals	287.67326207777774
	Dairy Products	366.94369421666664
	Snacks & Confectioneries	278.8934064129412
	Household	363.33601913333337
	Personal Care	364.99154823333333

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

	name	total_qty
▶	Bath Soap	60
	Hand Sanitizer	56
	Dishwashing Soap	54
	Biscuits	54
	Potato Chips	54
	Moong Dal	51
	Chapati	50
	Cumin Seeds	46
	Mustard Seeds	45
	Facial Tissue	45
	Toothpaste	44
	Mayonnaise	43
	Cashews	42
	Brown Bread	42
	Cheese Slices	42
	Butter	41
	Red Chili Powder	40
	Turmeric Powder	39
	Wheat Flour	38
	White Bread	38
	Jaggery	37
	Mango Pickle	37
	Cinnamon Sticks	37
	Black Pepper	36
	Tomato Ketchup	35
	Green Tea	35

```
select p.name,sum(od.quantity) as total_qty  
from products p  
join orderdetails od on p.productid = od.productid  
group by p.name  
order by total_qty desc;
```





```
select p.name,sum(od.totalprice) as revune  
from products p  
join orderdetails od on p.productid = od.productid  
group by p.name  
order by revune desc;
```

	name	revune
►	Hand Sanitizer	27787.7375380624
	Biscuits	20995.91542113018
	Moong Dal	19695.000539642097
	Toothpaste	19688.915112007253
	Mustard Seeds	19516.681792541927
	Cashews	18561.951841182195
	Butter	18548.43469889168
	Cheese Slices	18519.611222304677
	Turmeric Powder	17784.28916623548
	Soya Sauce	16985.373678400047
	Toilet Cleaner	16776.93114379072
	White Bread	16576.188568069545
	Paneer	15980.833408960965
	Detergent Powder	14848.704522101772
	Brown Bread	14538.060960123265
	Mouth wash	14537.135849234472
	Chocolate Bar	14278.168393836675
	Bath Soap	14112.984518210824
	Ghee	13161.288620246347
	Basmati Rice	11487.50885594245
	Tomato Ketchup	11283.918701633222
	Mayonnaise	11100.472261245737
	Conditioner	10949.853068835755
	Masala Tea	10283.024107008245
	Shampoo	10132.77932605984
	Chapati	10084.100549298700

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

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

```
select c.categoryname, s.suppliername,  
sum(od.quantity) as qty_sold,  
sum(od.totalprice) as total_revune  
from products p  
join categories c on p.categoryid = c.categoryid  
join suppliers s on p.supplierid = s.supplierid  
join orderdetails od on p.productid = od.productid  
group by c.categoryname, s.suppliername  
order by total_revune desc;
```

	categoryname	suppliername	qty_sold	total_revune
▶	Personal Care	Aarya	205	69378.3449114391
	Grains & Cereals	Aarya	226	67701.12114267625
	Snacks & Confectioneries	Aarya	196	65538.72632937467
	Snacks & Confectioneries	Suresh	272	65307.18296356301
	Dairy Products	Sai	121	50740.621495369945
	Grains & Cereals	Karthik	140	39473.49444946641
	Snacks & Confectioneries	Aarav Sharma	82	26948.17217315072
	Grains & Cereals	Suresh	67	26248.873655213643
	Household	Karthik	86	22767.542692147545
	Dairy Products	Aarya	42	18519.611222304677
	Grains & Cereals	Sai	75	18017.97092319238
	Snacks & Confectioneries	Sai	70	17103.164666122644
	Household	Sai	35	16776.93114379072
	Dairy Products	Karthik	43	11100.472261245737
	Personal Care	Sai	23	10949.853068835755
	Personal Care	Suresh	28	10132.77932605984
	Snacks & Confectioneries	Karthik	27	8520.423896711907
	Grains & Cereals	Aarav Sharma	34	6104.715663493892

## 10. How many orders have been placed in total?

```
SELECT COUNT(*) AS TotalOrders  
FROM Orders;
```



	TotalOrders
▶	300



# 11.What is the average value per order?

```
-- What is the average value per order?  
SELECT orderid, AVG(order_total) AS avg_order_value  
FROM (  
    SELECT od.orderid, SUM(od.totalprice) AS order_total  
    FROM orderdetails od  
    GROUP BY od.orderid  
) t  
group by orderid;
```



	orderid	avg_order_value
▶	109	1753.2508788489392
	144	2625.9382572541513
	82	5252.280909402865
	224	3869.530789092514
	256	1544.7059246778117
	183	1745.1100486969347
	174	4670.596219447797
	164	2265.8903529192235
	68	2928.626739529932
	129	934.400379648134
	249	1841.6256259615143
	36	1850.588331632953

12. On  
which dates  
were the  
most orders  
placed?



```
SELECT orderdate, COUNT(*) AS TotalOrders  
FROM orders  
GROUP BY orderdate  
ORDER BY TotalOrders DESC;
```

	orderdate	totalorders
	9/10/2022	4
	3/30/2022	4
▶	1/30/2022	3
	4/22/2022	3
		-

### 13.What are the monthly trends in order volume and revenue?

```
SELECT left(o.orderdate,7) as orderMonth,  
       COUNT(DISTINCT o.orderid) AS order_volume,  
       SUM(od.quantity * od.priceeach) AS TotalRevenue  
  FROM orders o  
 JOIN orderdetails od ON o.orderid = od.orderid  
 GROUP BY orderMonth  
 ORDER BY orderMonth;
```

ordermonth	order_volume	totalrevenue
1/1/202	1	711.7337763654041
1/10/20	1	133.509371642973
1/11/20	2	2217.1671727184425
1/12/20	1	3489.711438343752
1/14/20	2	3333.206112625238
1/15/20	1	5252.280909402865
1/16/20	3	6340.929931799683
1/18/20	1	1449.5748893116331
1/22/20	1	2168.5201991713247
1/23/20	2	3957.098409848345
1/24/20	2	7187.7236988635195
1/25/20	1	3485.79268361059
1/28/20	3	4703.030318201322

## 14. How do order patterns vary across weekdays and months?

```
SELECT
    CASE
        WHEN DAYOFWEEK(STR_TO_DATE(o.OrderDate, '%m/%d/%Y')) IN (1, 7) THEN 'Weekend'      -- Saturday or Sunday
        ELSE 'Weekday'
    END AS day_type,
    COUNT(DISTINCT o.OrderID) AS orders_count, SUM(od.TotalPrice) AS revenue
FROM Orders o
JOIN OrderDetails od
    ON o.OrderID = od.OrderID
GROUP BY day_type;
```



Result Grid | Filter Rows:

	day_type	orders_count	revenue
▶	Weekday	183	390922.5691772748
	Weekend	73	160407.43280688356

15.How many suppliers are there in the database?

```
SELECT COUNT(*) AS TotalSuppliers  
FROM suppliers;
```

TotalSuppliers	
▶	5



## 16.Which supplier provides the most products?

```
SELECT s.supplierid, s.suppliername, COUNT(p.productid) AS ProductCount  
FROM suppliers s  
JOIN products p ON s.supplierid = p.supplierid  
GROUP BY s.supplierid, s.suppliername  
ORDER BY ProductCount DESC  
LIMIT 1;
```



	supplierid	suppliername	ProductCount
▶	3	Aarya	18

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

```
SELECT s.supplierid, s.suppliername, AVG(p.price) AS AvgProductPrice  
FROM suppliers s  
JOIN products p ON s.supplierid = p.supplierid  
GROUP BY s.supplierid, s.suppliername  
ORDER BY AvgProductPrice DESC;
```



	supplierid	suppliername	AvgProductPrice
▶	2	Sai	342.67186315000004
	3	Aarya	319.32674083999996
	5	Karthik	288.22674025555557
	4	Suresh	281.81799177
	1	Aarav Sharma	271.36853063333336

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

```
SELECT s.supplierid, s.suppliername,  
       SUM(od.totalprice) AS TotalRevenue  
  FROM suppliers s  
 JOIN products p ON s.supplierid = p.supplierid  
 JOIN orderdetails od ON p.productid = od.productid  
 GROUP BY s.supplierid, s.suppliername  
 ORDER BY TotalRevenue DESC;
```

	supplierid	suppliername	TotalRevenue
▶	3	Aarya	221137.80360579453
	2	Sai	113588.54129731141
	4	Suresh	101688.83594483654
	5	Karthik	81861.93329957161
	1	Aarav Sharma	33052.887836644615

## 19. How many employees have processed orders?

```
SELECT COUNT(DISTINCT employeeid) AS ActiveEmployees  
FROM orders;
```



ActiveEmployees
10

## 20.Which employees have handled the most orders?

```
SELECT e.employeeid, e.name,  
       COUNT(DISTINCT o.orderid) AS OrdersHandled  
FROM store_employees e  
JOIN orders o ON e.employeeid = o.employeeid  
GROUP BY e.employeeid, e.name  
ORDER BY OrdersHandled DESC;
```

	employeeid	name	OrdersHandled
▶	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
	6	Zara Verma 1	30
	7	Vihaan Singh 1	29
	4	Aditya Verma 1	26
	1	Aarav Kumar 1	23
	10	Arjun Verma 1	23

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

```
SELECT e.employeeid, e.name,  
       SUM(od.totalprice) AS TotalSales  
  FROM store_employees e  
 JOIN orders o ON e.employeeid = o.employeeid  
 JOIN orderdetails od ON o.orderid = od.orderid  
 GROUP BY e.employeeid, e.name  
 ORDER BY TotalSales DESC;
```

	employeeid	name	TotalSales
▶	2	Aditya Singh 1	79252.27287503115
	6	Zara Verma 1	71562.77735649435
	8	Diya Sharma 1	67241.86324808231
	3	Pari Kumar 1	66818.37989874712
	9	Arjun Kumar 1	54018.34636253491
	1	Aarav Kumar 1	52602.86326421212
	7	Vihaan Singh 1	48577.903053390895
	5	Pari Sharma 1	40334.23183705283
	10	Arjun Verma 1	36716.86024034629
	4	Aditya Verma 1	34204.5038482668

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

```
SELECT quantity, totalprice  
FROM orderdetails;
```



Result Grid	Filter Rows:	Export:	
	priceeach	quantity	totalprice
▶		140.6243555888351	3
▶		441.95123431386185	1
▶		166.26417041398335	4
▶		219.36278409304612	2
▶		386.1764811694529	4
▶		146.64515129714397	4
▶		464.02201631568045	3
▶		322.39767718952066	1
▶		182.7412994928937	3
▶		386.1764811694529	1
▶		235.21640863684706	4
▶		258.15051770338926	3
▶		179.55046069099683	1
▶		235.21640863684706	3
▶		258.15051770338926	4
▶		487.45513408319806	4
▶		201.6839929677742	1
▶		224.5675782233399	4
▶		464.02201631568045	4



## 23.What is the average quantity ordered per product?

```
SELECT p.productid, p.name, AVG(od.quantity) AS AvgQuantity  
FROM products p  
JOIN orderdetails od ON p.productid = od.productid  
GROUP BY p.productid, p.name  
ORDER BY AvgQuantity DESC;
```



	productid	name	avg_quantity_per_line
▶	40	Butter	4.5556
	31	Toothpaste	3.6667
	46	Potato Chips	3.6000
	42	Tomato Ketchup	3.5000
	22	Mustard Seeds	3.4615
	3	Moong Dal	3.4000
	6	Ghee	3.3750
	9	Mango Pickle	3.3636
	32	Bath Soap	3.3333
	45	Chili Sauce	3.2500
	18	Salt	3.2500
	34	Facial Tissue	3.2143
	1	Basmati Rice	3.2000
	26	Detergent Pow...	3.2000
	35	Mouth wash	3.2000
	16	Sugar	3.2000
	13	Green Tea	3.1818

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

```
SELECT p.productid, p.name, od.eachprice
```

```
FROM products p
```

```
JOIN orderdetails od ON p.productid = od.productid
```

```
ORDER BY p.name, od.eachprice;
```



	productid	name	priceeach
▶	23	Black Pepper	140.6243555888351
	12	Cashews	441.95123431386185
	13	Green Tea	166.26417041398335
	18	Salt	219.36278409304612
	3	Moong Dal	386.1764811694529
	27	Dishwashing Soap	146.64515129714397
	26	Detergent Powder	464.02201631568045
	42	Tomato Ketchup	322.39767718952066
	21	Cumin Seeds	182.7412994928937
	3	Moong Dal	386.1764811694529
	32	Bath Soap	235.21640863684706
	43	Mayonnaise	258.15051770338926
	15	Coffee Powder	179.55046069099683
	32	Bath Soap	235.21640863684706
	43	Mayonnaise	258.15051770338926
	6	Ghee	487.45513408319806
	26	Chapati	301.5820070677742

# **Conclusion:**

**The SQL-based Grocery Store Management System successfully demonstrates how a relational database can enhance operational efficiency in a retail environment. By automating data management, improving accuracy, and enabling powerful data analysis, the system reduces manual workload and supports better business decisions.**

**The project highlights the importance of structured data design, optimized SQL queries, and well-defined relationships between tables. Overall, the solution provides a reliable, scalable, and efficient approach to managing everyday grocery store operations and serves as a strong foundation for integrating advanced features such as dashboards, analytics, and POS automation.**

**THANK  
YOU**



**Group Members:**

**T.Kiran kumar  
P.Naveen**