



INNOMATICS<sup>®</sup>  
RESEARCH LABS

INNOVATION. AUTOMATION. ANALYTICS

**PROJECT ON**  
**Grocery Store**  
**Management**

# Grocery Store Management



# Introduction

- The Retail and Grocery domain requires effective management of inventory, suppliers, customer orders, employees, and product categories.
- Efficient data handling helps businesses track sales and revenue, monitor product availability, and analyze operations.
- This project simulates a mini grocery store database with interconnected entities like products, orders, and suppliers.
- It offers hands-on experience in understanding data relationships and operational workflows.
- Students will use SQL to extract and transform data, gaining valuable business insights.
- The goal is to improve decision-making and operational efficiency through data analysis.

# Tables

## Products Table

Field	Type	Null	Key	Default
prod_id	tinyint	NO	PRI	<b>NUL</b>
prod_name	varchar(255)	YES		<b>NUL</b>
sup_id	tinyint	YES	MUL	<b>NUL</b>
cat_id	tinyint	YES	MUL	<b>NUL</b>
price	decimal(10,2)	YES		<b>NUL</b>

## Suppliers Table

Field	Type	Null	Key	Default
sup_id	tinyint	NO	PRI	<b>NUL</b>
sup_name	varchar(255)	YES		<b>NUL</b>
address	text			<b>NUL</b>

## Customers Table

Field	Type	Null	Key	Default
cust_id	smallint	NO	PRI	<b>NUL</b>
cust_name	varchar(255)	YES		<b>NUL</b>
address	text	YES		<b>NUL</b>

## Categories Table

Field	Type	Null	Key	Default
cat_id	tinyint	NO	PRI	<b>NUL</b>
cat_name	varchar(255)	YES		<b>NUL</b>

# Tables

## Employees Table

Field	Type	Null	Key	Default
emp_id	tinyint	NO	PRI	<b>NUL</b>
emp_name	varchar(255)	YES		<b>NUL</b>
hire_date	varchar(255)	YES		<b>NUL</b>

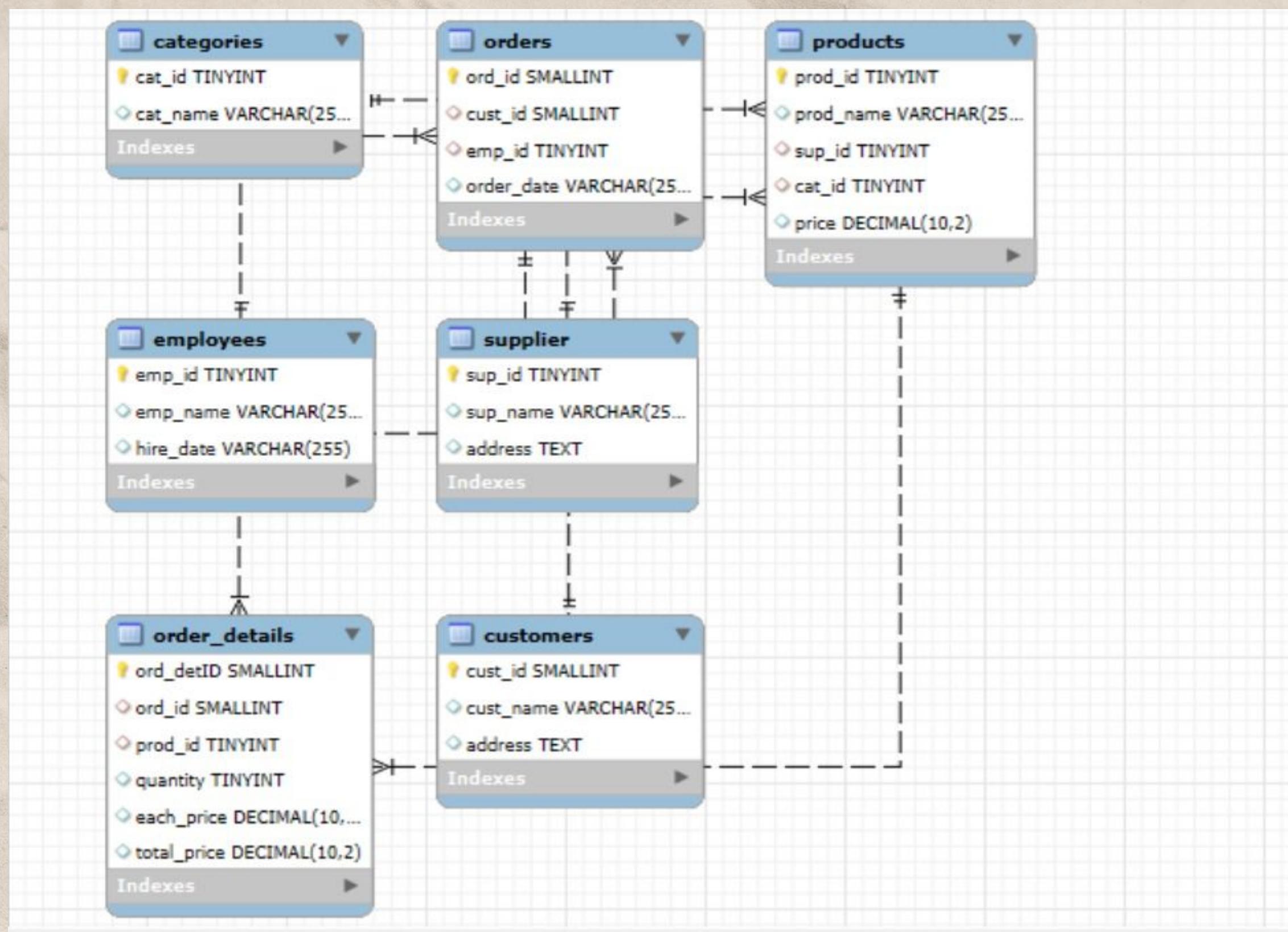
## Order\_details Table

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

## Orders Table

Field	Type	Null	Key	Default
ord_id	smallint	NO	PRI	<b>NUL</b>
cust_id	smallint	YES	MUL	<b>NUL</b>
emp_id	tinyint	YES	MUL	<b>NUL</b>
order_date	varchar(255)	YES		<b>NUL</b>

# Relationship Schema



# Problem Statement

In the dynamic environment of a retail grocery store, managing and analyzing large volumes of data related to products, suppliers, customers, and orders is crucial for efficient operations and informed decision-making. This project aims to design and implement a relational database system that captures essential store activities and enables users to retrieve and analyze data through SQL. The system will support complex queries to identify key business insights such as top-performing products, high-value customers, and revenue patterns, while reinforcing practical SQL skills including joins, aggregations, subqueries, and filtering.

# 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.ord_id) AS total_orders  
FROM  
    customers c  
    JOIN  
    orders o ON c.cust_id = o.cust_id  
GROUP BY c.cust_id  
ORDER BY total_orders DESC  
LIMIT 1;
```



cust_id	cust_name	total_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) AS total_products  
FROM  
    categories c  
    LEFT JOIN  
    products p ON c.cat_id = p.cat_id  
GROUP BY c.cat_id;
```



# Food!

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

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

SELECT

c.cat\_id, c.cat\_name, AVG(p.price) AS avg\_price

FROM

categories c

LEFT JOIN

products p ON c.cat\_id = p.cat\_id

GROUP BY c.cat\_id;



cat_id	cat_name	avg_price
1	Grains & Cereals	287.673333
2	Dairy Products	366.943333
3	Beverages	278.892353
4	Personal Care	364.991667
5	Snacks & Confectioneries	363.336667



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

**SELECT**

p.prod\_name, SUM(quantity) **AS** high\_sales

**FROM**

order\_details o

**LEFT JOIN**

products p **ON** o.prod\_id = p.prod\_id

**GROUP BY** prod\_name

**ORDER BY** high\_sales **DESC**

**LIMIT** 1;



prod_name	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;
```

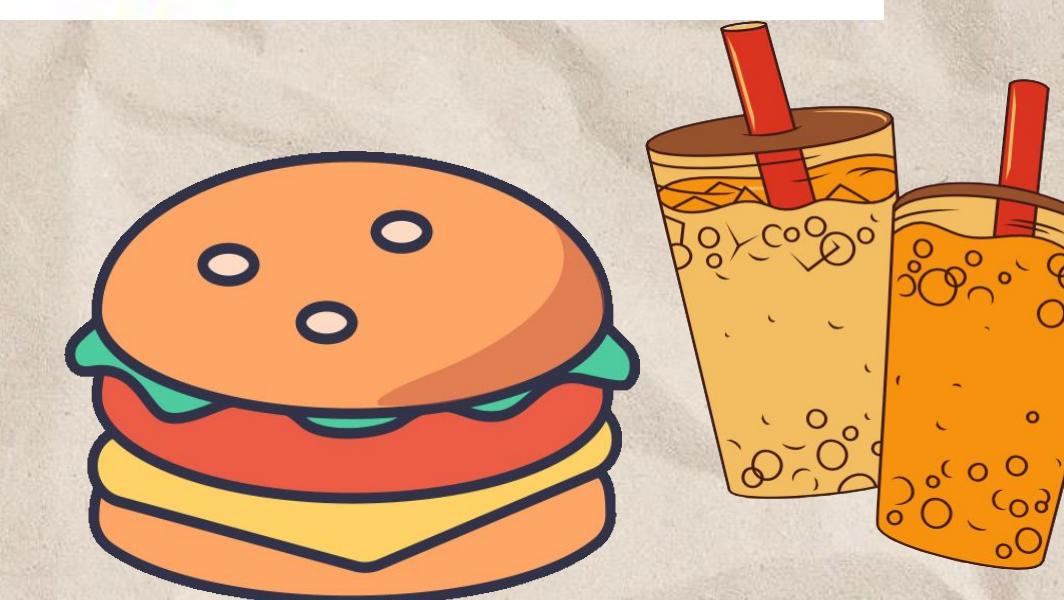


total_revenue	prod_name
5062.45	Black Pepper
18561.92	Cashews
5819.23	Green Tea
5703.43	Salt
19695.02	Moong Dal
7918.87	Dishwashing Soap
14848.72	Detergent Powder
11283.92	Tomato Ketchup
8406.10	Cumin Seeds
14113.00	Bath Soap
11100.45	Mayonnaise
6104.70	Coffee Powder
13161.31	Ghee
10084.19	Chapati
8309.02	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
ORDER BY c.cat_name , total_sales DESC;
    
```



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

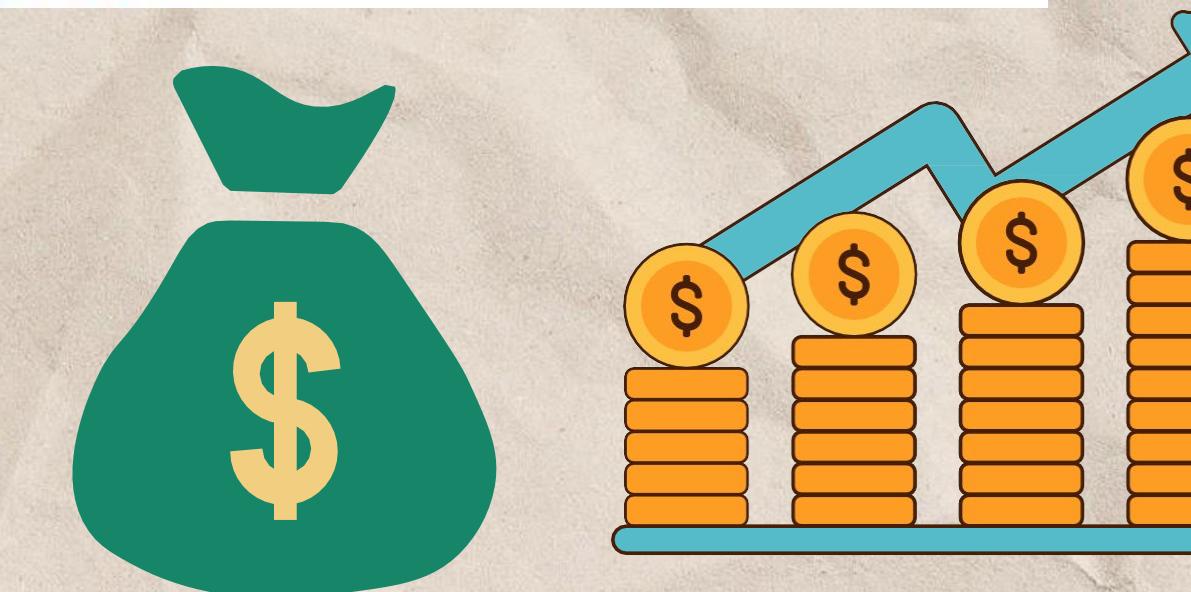


order_date	orders_count
9/10/2022	4

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

```

SELECT
    LEFT(o.order_date, 7) AS order_month,
    COUNT(DISTINCT o.ord_id) AS order_volume,
    SUM(od.total_price) AS total_revenue
FROM
    orders o
        JOIN
    order_details od ON o.ord_id = od.ord_id
GROUP BY order_month
ORDER BY order_month;
    
```



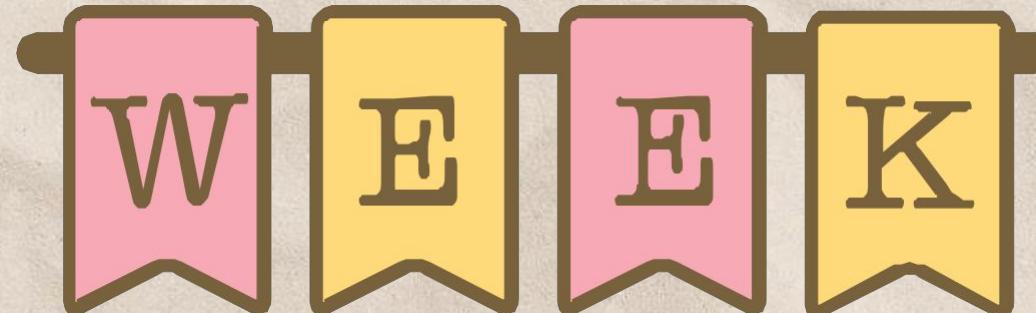
order_month	order_volume	total_revenue
1/1/202	1	711.73
1/10/20	1	133.51
1/11/20	2	2217.16
1/12/20	1	3489.70
1/14/20	2	3333.21
1/15/20	1	5252.28
1/16/20	3	6340.93
1/18/20	1	1449.57
1/22/20	1	2168.52
1/23/20	2	3957.10
1/24/20	2	7187.71
1/25/20	1	3485.80
1/28/20	3	4703.02
1/30/20	2	7433.69
1/31/20	1	3125.69
1/5/202	2	5466.54
1/6/202	1	1944.07
1/7/202	2	5553.92
1/8/202	1	2358.30
10/10/2	2	2378.37
10/14/2	1	1667.05
10/19/2	2	7406.59

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

```

UPDATE orders
SET order_date = DATE_FORMAT(STR_TO_DATE(order_date, '%m/%d/%Y'), '%Y-%m-%d');

SELECT
    DAYNAME(STR_TO_DATE(order_date, '%Y-%m-%d')) AS weekday,
    MONTHNAME(STR_TO_DATE(order_date, '%Y-%m-%d')) AS month,
    COUNT(ord_id) AS total_orders
FROM
    orders
GROUP BY weekday , month
ORDER BY weekday , month;
    
```



	weekday	month	total_orders
▶	Sunday	January	8
	Saturday	July	8
	Friday	November	6
	Wednesday	May	3
	Saturday	March	3
	Wednesday	August	6
	Friday	April	4
	Thursday	March	7
	Monday	October	4
	Wednesday	November	4
	Saturday	June	4
	Tuesday	August	1
	Thursday	July	2
	Friday	July	5
	Tuesday	December	5
	Monday	March	2
	Friday	January	8
	Saturday	April	2
	Monday	February	4

## 15.How many suppliers are there in the database?

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

```
COUNT(*)  
5
```



## 16.Which supplier provides the most products?

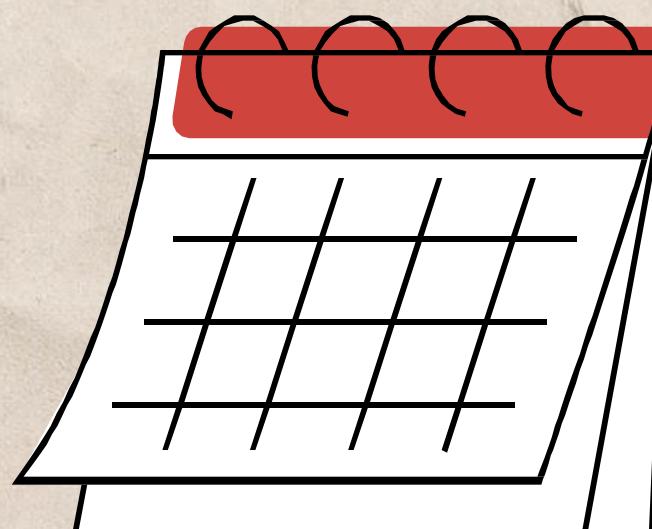
```
SELECT  
    s.sup_id, s.sup_name, COUNT(p.prod_id) AS product_count  
FROM  
    supplier s  
        JOIN  
    products p ON s.sup_id = p.sup_id  
GROUP BY s.sup_id  
ORDER BY product_count DESC  
LIMIT 1;
```



sup_id	sup_name	product_count
3	Aarya	18

## 17.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

## 18. 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

## 19. How many employees have processed orders?

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



COUNT(DISTINCT emp_id)
▶ 10

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

```
SELECT  
    e.emp_id, e.emp_name, COUNT(o.ord_id) AS ordered_most  
FROM  
    orders o  
        JOIN  
    employees e ON o.emp_id = e.emp_id  
GROUP BY e.emp_id  
ORDER BY ordered_most DESC  
LIMIT 1;
```

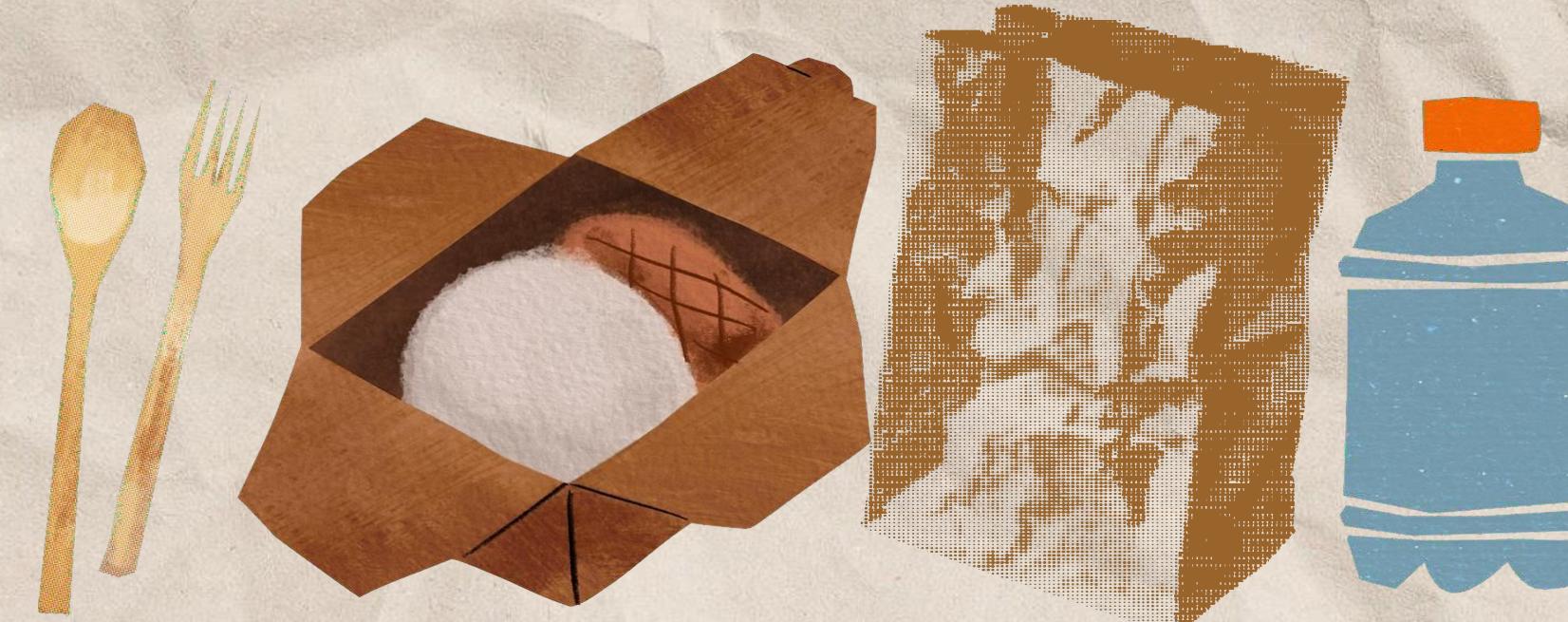


	emp_id	emp_name	ordered_most
▶	8	Diya Sharma	38

## 21.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



## 22.What is the average order value handled per employee?

```
SELECT  
    e.emp_id,  
    e.emp_name,  
    SUM(od.total_price) / COUNT(DISTINCT o.ord_id) AS avg_value  
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 avg_value DESC;
```



	emp_id	emp_name	avg_value
▶	1	Aarav Kumar 1	2768.572632
	6	Zara Verma 1	2650.472593
	2	Aditya Singh 1	2330.949706
	3	Pari Kumar 1	2227.279667
	7	Vihaan Singh 1	2112.081739
	9	Arjun Kumar 1	2077.627308
	8	Diya Sharma 1	2037.631818
	10	Arjun Verma 1	1835.842000
	5	Pari Sharma 1	1833.373636
	4	Aditya Verma 1	1554.750455

## 23.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

## 24.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

## 25. 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



Thank You  
for your attention

**Group Members:**  
**Mrutyunjaya Debata**  
**Narottam Kar**  
**Daniel Paul**