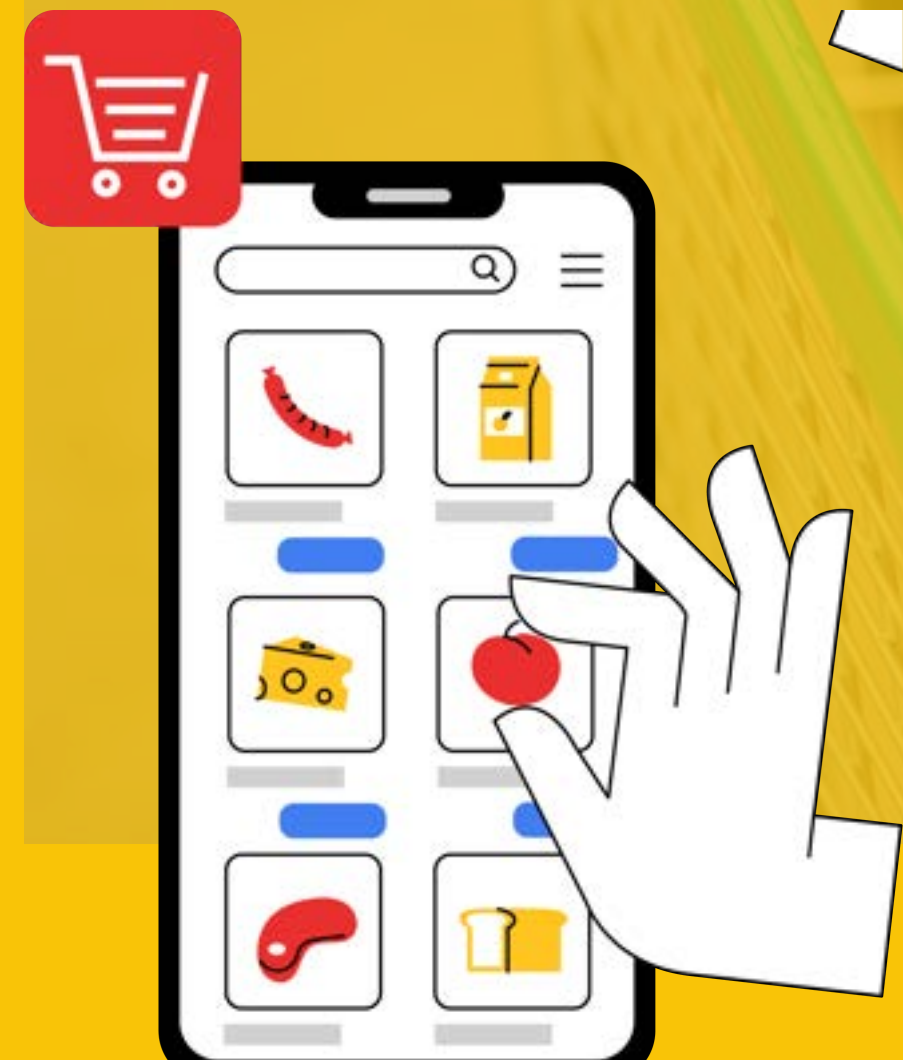




# BLINKIT ANALYSIS SQL PROJECT

CREATED BY: ROSHNI KUMARI







# Project Overview

## **Blinkit SQL Project – Sales & Order Analytics**

**This project simulates Blinkit's sales and order data to practice SQL skills. It includes creating sample datasets, solving business-related queries, and compiling the results into a visual PDF report.**

## **Project Summary**

- **Created 4 key tables: customers, products, orders, and order\_items**
- **Framed and solved real-world SQL questions**
- **Took help from ChatGPT for sample data, question generation, and query validation**
- **Final output compiled with screenshots into a clean, structured PDF**





# SQL Questions

## 1. Top Spending Customers

List the top 5 customers who spent the most in total, along with their total amount spent.

## 2. Popular Categories

Find the product category with the highest number of items sold.

## 3. Average Order Value per City

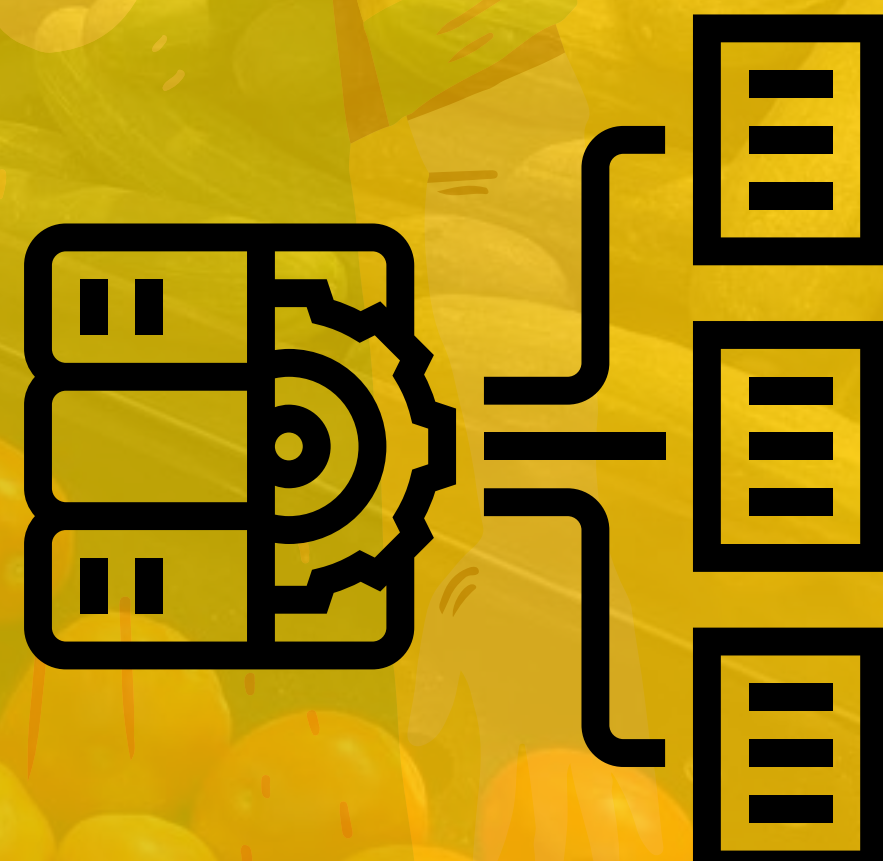
Calculate the average order total for customers in each city.

## 4. Best-Selling Products

Which 5 products have been ordered the most by quantity?

## 5. Low Stock Alert

List all products with a stock quantity less than 100.







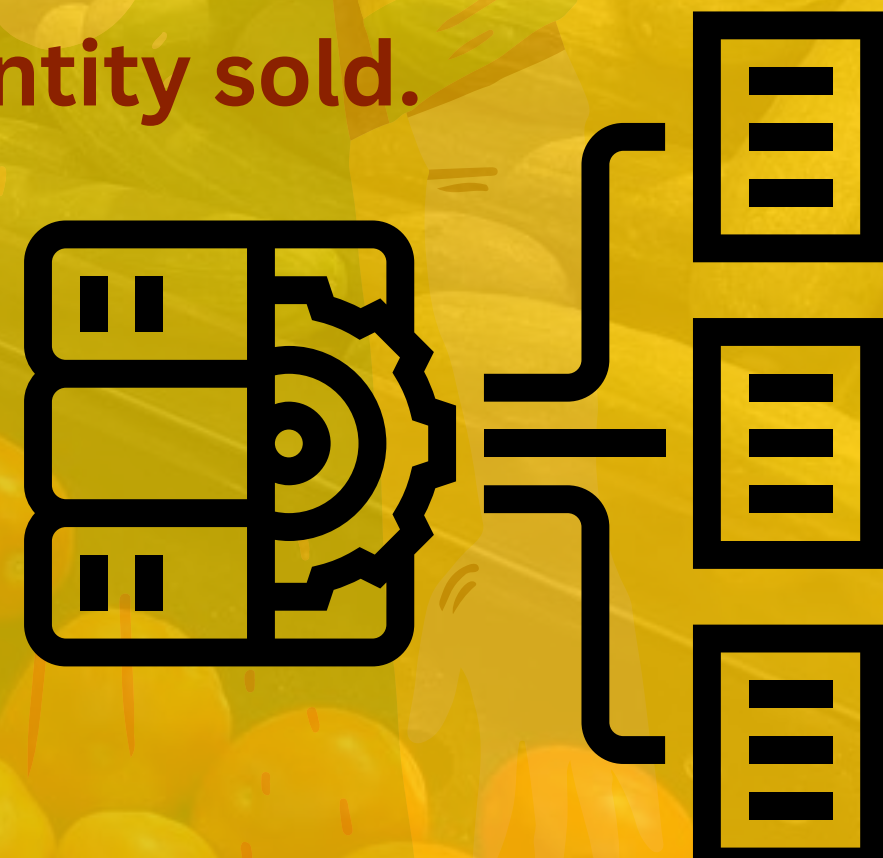
# SQL Questions

## 6. Customer Purchase Summary

For each customer, show total number of orders and total items purchased.

## 7. Product Performance Report

For each product, show total revenue generated and total quantity sold.





# 1. Top Spending Customers

List the top 5 customers who spent the most in total, along with their total amount spent.

```
-- Top Spending Customers
-- List the top 5 customers who spent the most in total, along with their total amount spent.

SELECT
    c.name,
    c.email,
    c.city,
    c.phone,
    SUM(o.total_amount) AS total_amount_spent
FROM
    customers AS c
    JOIN
    orders AS o ON c.customer_id = o.customer_id
GROUP BY c.name , c.email , c.city , c.phone
ORDER BY total_amount_spent DESC
LIMIT 5;
```

	name	email	city	phone	total_amount_spent
►	Isha Kapoor	ishakapoor@example.com	Delhi	9876543210	20999.00
	Manoj Kumar	manojkumar@example.com	Ahmedabad	9876543217	8999.00
	Rahul Verma	rahulverma@example.com	Mumbai	9876543211	5798.00
	Arjun Nair	arjunnair@example.com	Bhopal	9876543222	3798.00
	Tanvi Jain	tanvijain@example.com	Indore	9876543221	2999.00

## 2. Popular Categories

Find the product category with the highest number of items sold.

```
-- Find the product category with the highest number of items sold.
```

```
SELECT
    p.category,
    sum(oi.quantity) as total_items_sold
FROM
    order_items AS oi
    JOIN
        products AS p
    ON oi.product_id = p.product_id
GROUP BY p.category
ORDER BY
    total_items_sold DESC
LIMIT 1;
```

Result Grid



Filter Rows:

	category	total_items_sold
▶	Electronics	12




### 3. Average Order Value per City

Calculate the average order total for customers in each city.

```
-- Average Order Value per City
-- Calculate the average order total for customers in each city.

SELECT
    c.city, ROUND(AVG(o.total_amount), 0) AS Average_Order_Value
FROM
    orders AS o
    JOIN
    customers AS c ON o.customer_id = c.customer_id
GROUP BY c.city;
```



	city	Average_Order_Value
▶	Delhi	20999
	Mumbai	5798
	Bangalore	1699
	Kolkata	1598
	Hyderabad	2499
	Chennai	1299
	Pune	1499
	Ahmedabad	5249
	Lucknow	1797
	Surat	699

Result 23 x

## 4. Best-Selling Products

Which 5 products have been ordered the most by quantity?

```
-- Best-Selling Products
-- Which 5 products have been ordered the most by quantity?

SELECT
    p.name, SUM(oi.quantity) AS total_quantity_ordered
FROM
    products AS p
    JOIN
    order_items AS oi ON p.product_id = oi.product_id
GROUP BY p.name
ORDER BY total_quantity_ordered DESC
LIMIT 5;
```

Result Grid			Filter Rows:
	name	total_quantity_ordered	
▶	Wireless Mouse	3	
	Running Shoes	2	
	Water Bottle	2	
	Leather Wallet	2	
	Induction Cooktop	2	



## 5. Low Stock Alert

List all products with a stock quantity less than 100.

```
-- Low Stock Alert
-- List all products with a stock quantity less than 100.

SELECT
    name, category, stock
FROM
    products
WHERE
    stock < 100;
```

Result Grid				Filter Rows:	Expo
	name	category	stock		
▶	Laptop Bag	Accessories	90		
	LED Monitor	Electronics	80		
	Mixer Grinder	Home Appliances	75		
	Induction Cooktop	Home Appliances	60		



# 6. Customer Purchase Summary

For each customer, show total number of orders and total items purchased.

```
-- Customer Purchase Summary
-- For each customer, show total number of orders and total items purchased.
SELECT
  c.name,
  c.city,
  c.email,
  c.phone,
  COUNT(DISTINCT o.order_id) AS total_no_of_orders,
  SUM(oi.quantity) AS total_items_purchased
FROM
  customers AS c
  JOIN
  orders AS o ON c.customer_id = o.customer_id
  JOIN
  order_items AS oi ON o.order_id = oi.order_id
GROUP BY c.name , c.city , c.email , c.phone;
```

Result Grid						
Filter Rows:		Export:		Wrap Cell Content:		
	name	city	email	phone	total_no_of_orders	total_items_purchased
▶	Aarti Rawat	Meerut	aartirawat@example.com	9876543245	1	1
	Aditya Joshi	Jodhpur	adityajoshi@example.com	9876543236	1	1
	Alok Tiwari	Thane	aloktiwari@example.com	9876543229	1	1
	Aman Singh	Chennai	amansingh@example.com	9876543215	1	1
	Ananya Joshi	Bangalore	ananyajoshi@example.com	9876543212	1	1
	Arjun Nair	Bhopal	arjunnair@example.com	9876543222	1	2
	Bhavna Shah	Vadodara	bhavnashah@example.com	9876543230	1	1
	Deepak Mehta	Surat	deepakmehta@example.com	9876543219	1	1
	Dev Sharma	Tiruchirappalli	devsharma@example.com	9876543248	1	1
	Divya Bhatt	Jaipur	divyabhatt@example.com	9876543223	1	1



# 7. Product Performance Report

For each product, show total revenue generated and total quantity sold.

```
-- Product Performance Report
-- For each product, show total revenue generated and total quantity sold.
SELECT
  p.product_id,
  p.name,
  p.category,
  SUM(oi.quantity) AS total_quantity_sold,
  SUM(oi.quantity * oi.price) AS total_revenue
FROM
  products AS p
  JOIN
  order_items AS oi ON p.product_id = oi.product_id
GROUP BY p.product_id , p.name , p.category
```

product_id	name	category	total_quantity_sold	total_revenue
101	Smartphone	Electronics	1	20999.00
102	Running Shoes	Footwear	2	5798.00
103	Bluetooth Speaker	Electronics	1	1699.00
104	Leather Wallet	Accessories	2	1598.00
105	Digital Watch	Wearables	1	2499.00
106	Laptop Bag	Accessories	1	1299.00
107	Backpack	Accessories	1	1499.00
108	LED Monitor	Electronics	1	8999.00
109	Wireless Mouse	Electronics	3	1797.00
110	Keyboard	Electronics	1	699.00





## Conclusion

**This project helped reinforce core SQL concepts through hands-on practice with realistic business scenarios. By simulating Blinkit's operations, I improved my skills in data modeling, query writing, and deriving insights—building a strong foundation for real-world data analysis tasks.**



