E-COMMERCE TRENDS

Presented By
Shahbaz Khan

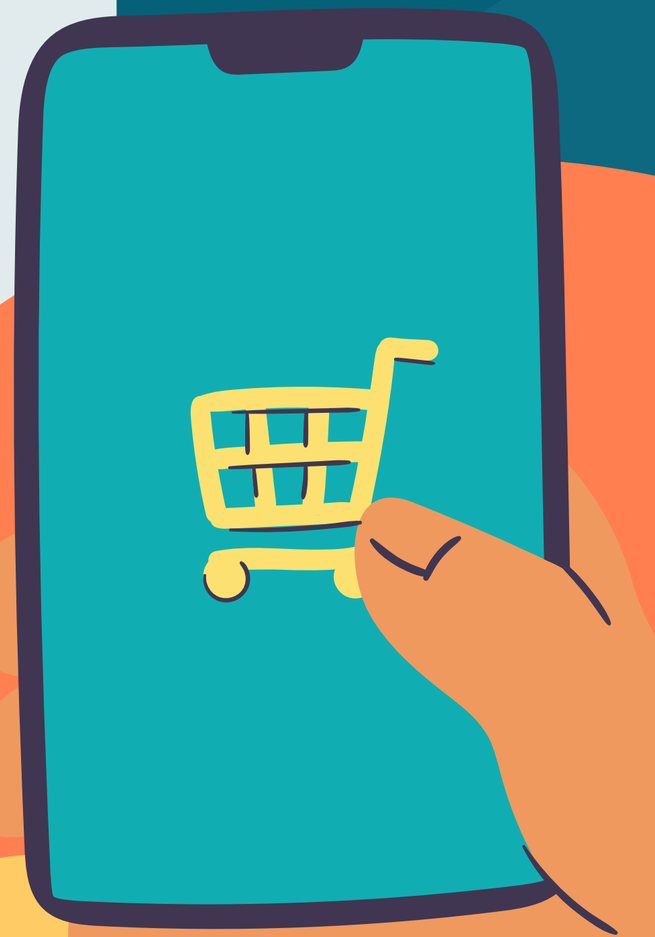
INTRODUCTION

In this project, I created an E-Commerce Database using MySQL to analyze key business insights. The system includes tables for users, orders, products, categories, and order items. I designed relational tables to capture customer orders, product details, and transactions. I used SQL to calculate total revenue, analyze purchasing patterns, and identify top-selling products and frequent items in orders. The project demonstrates my skills in database design, SQL querying, and data analysis for e-commerce applications.

FETCH THE NAMES OF USERS ALONG WITH THE TOTAL AMOUNT THEY HAVE SPENT ON THEIR ORDERS

```sql
SELECT users.name,
SUM(order_items.price * order_items.quantity) AS total_amount
FROM users
JOIN orders ON users.user_id = orders.user_id
JOIN order_items ON order_items.order_id = orders.order_id
GROUP BY users.name
ORDER BY total_amount DESC;
```

name	total_amount
John Doe	860.00
Jane Smith	600.00



LIST ALL ORDERS, INCLUDING THE PRODUCTS AND THEIR QUANTITIES, FOR EACH ORDER.

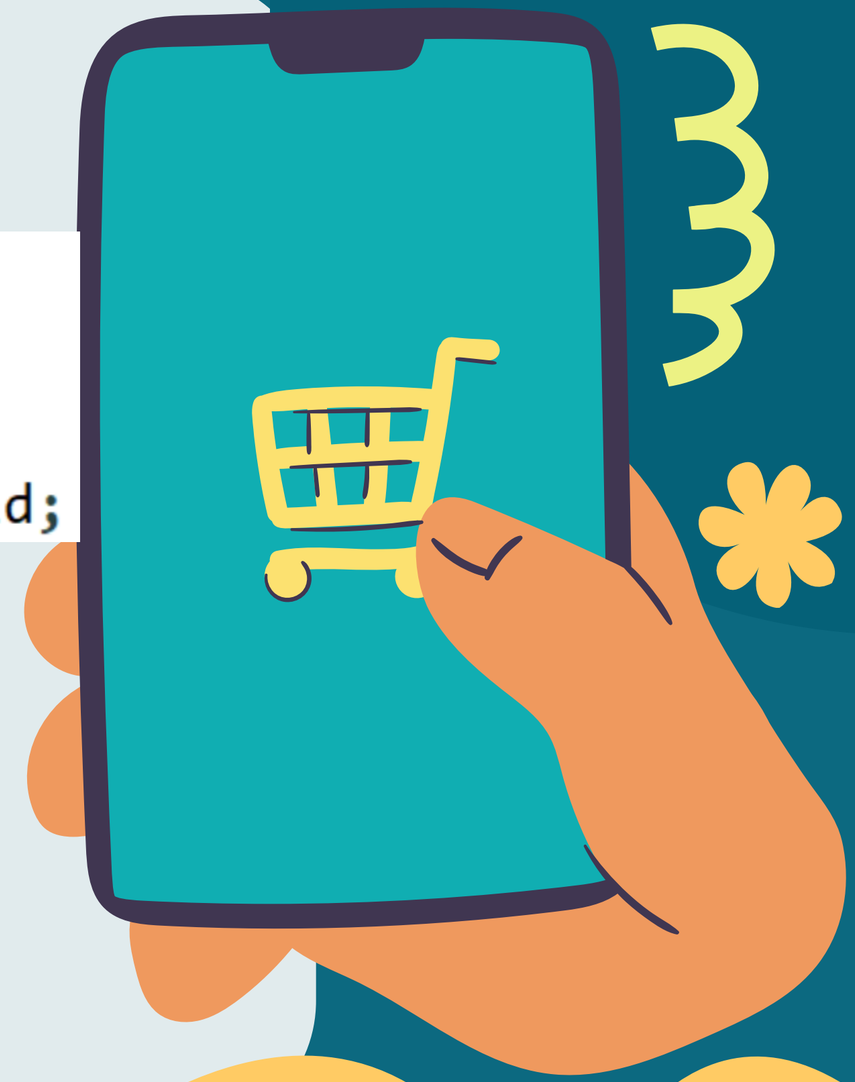
```sql
SELECT orders.order_id,products.name AS product_name,
order_items.quantity
FROM orders
JOIN order_items ON orders.order_id = order_items.order_id
JOIN products ON order_items.product_id = products.product_id
ORDER BY orders.order_id;
```

order_id	product_name	quantity
1	Laptop	1
1	Fiction Book	2
2	Smartphone	1

RETRIEVE THE NAME OF THE CATEGORY ALONG WITH THE PRODUCT NAME FOR A PRODUCTS

```
select categories.name as category_name,
products.name as Product_name
from categories join products
on categories.category_id = products.category_id;
```

category_name	Product_name
Electronics	Laptop
Electronics	Smartphone
Books	Fiction Book
Clothing	T-Shirt



LIST ALL USERS WHO HAVE MADE A PAYMENT USING THE "CREDIT CARD" METHOD

```sql
SELECT users.name
FROM users
JOIN orders ON users.user_id = orders.user_id
JOIN payments
ON orders.order_id = payments.order_id
WHERE payments.method = 'Credit Card';
```

name
John Doe

RETRIEVE THE TOP 3 PRODUCTS WITH THE HIGHEST STOCK LEVELS ACROSS ALL CATEGORIES

```sql
select categories.name as category_name,
products.name as product_name,products.stock
from products join categories
on products.category_id = categories.category_id
order by products.stock desc limit 3;
```

category_name	product_name	stock
Clothing	T-Shirt	100
Books	Fiction Book	50
Electronics	Smartphone	15

FIND THE MOST EXPENSIVE PRODUCT ACROSS CATEGORIES

```sql
SELECT categories.name AS category_name,
products.name AS product_name,
products.price AS product_price
FROM products
JOIN categories
ON categories.category_id = products.category_id
WHERE products.price = (SELECT MAX(price)
FROM products WHERE
category_id = products.category_id);
```

category_name	product_name	product_price
Electronics	Laptop	800.00

CALCULATE THE TOTAL REVENUE GENERATED FROM COMPLETED ORDERS

```sql
SELECT orders.status as order_status,
SUM(order_items.price * order_items.quantity)
AS total_revenue
FROM orders
JOIN order_items
ON orders.order_id = order_items.order_id
WHERE orders.status = 'Completed';
```

order_status	total_revenue
Completed	860.00

LIST THE USERS WHO PLACED ORDERS FOR PRODUCTS FROM MULTIPLE CATEGORIES

```
select users.name as customer_name,
categories.name as Category_name
from categories join products
on categories.category_id = products.category_id
join order_items
on order_items.product_id = products.product_id
join orders
on orders.order_id = order_items.order_id
join users on users.user_id = orders.user_id;
```

customer_name	Category_name
John Doe	Electronics
John Doe	Books
Jane Smith	Electronics

DISPLAY THE AVERAGE ORDER VALUE (AMOUNT) FOR EACH USER

```sql
SELECT users.name AS customer_name,
round(AVG(order_items.price * order_items.quantity),0)
AS average_order_value
FROM users JOIN orders
ON users.user_id = orders.user_id
JOIN order_items
ON orders.order_id = order_items.order_id
GROUP BY users.user_id;
```

customer_name	average_order_value
John Doe	430
Jane Smith	600

IDENTIFY THE PRODUCT THAT APPEARS MOST FREQUENTLY ACROSS ALL ORDERS

```sql
SELECT products.name AS product_name,
COUNT(order_items.product_id) AS product_count
FROM order_items
JOIN products
ON order_items.product_id = products.product_id
GROUP BY products.product_id
ORDER BY product_count DESC
LIMIT 1;
```

product_name	product_count
Laptop	1

THANK YOU