PROJECT

E-COMMERCE

MySQL

DATA ANALYST

sufiyanfarzand150@gmail.com 03085235952

check Now

INTRODUCTION

This MySQL database project is designed to power a complete online store, managing products, customers, orders, payments, and inventory efficiently. Built with scalability and performance in mind, it ensures smooth handling of high-volume transactions while maintaining data integrity.

QUESTION:1 Find all customers from New York

```sql
SELECT
    *
FROM
    customers
WHERE
    city = 'new york';
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

customer_id	first_name	last_name	email	phone	address	city	state	zip_code	registration_date
1	John	Doe	john.doe@email.com	555-0101	123 Main St	New York	NY	10001	2022-01-15
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

QUESTION:2 LIST ALL PRODUCTS IN THE ELECTRONICS CATEGORY

```sql
SELECT
    *
FROM
    products
WHERE
    category = 'electronics';
```

product_id	product_name	category	price	description	stock_quantity	supplier_id	created_at
101	Wireless Headphones	Electronics	99.99	Noise-cancelling wireless headphones with 30hr ...	50	1	2022-01-01
102	Smart Watch	Electronics	199.99	Fitness tracker with heart rate monitor	30	1	2022-01-10
107	Bluetooth Speaker	Electronics	79.99	Portable speaker with 20hr playtime	35	1	2022-04-01
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

QUESTION:3 HOW TO CREATE VIEW OF CUSTOMER NAME,CITY AND ADDRESS

```
    CREATE VIEW nessary AS
    SELECT
        first_name, last_name, city, address
    FROM
        customers;
SELECT
    *
FROM
    nessary;
```

first_name	last_name	city	address
John	Doe	New York	123 Main St
Jane	Smith	Los Angeles	456 Oak Ave
Robert	Johnson	Chicago	789 Pine Rd
Emily	Williams	Houston	321 Elm St
Michael	Brown	Phoenix	654 Maple Dr

QUESTION:4

CALCULATE TOTAL SALES FOR EACH CUSTOMER

```sql
SELECT
    first_name, SUM(total_amount) AS sales
FROM
    customers AS c
        JOIN
    orders AS o ON c.customer_id = o.customer_id
GROUP BY first_name
ORDER BY sales DESC;
```

first_name	sales
John	259.96
Robert	229.97
Emily	129.98
Jane	89.99
Michael	49.99

QUESTION:5 FIND THE MOST POPULAR PRODUCT (BY QUANTITY SOLD)

```sql
SELECT
    product_name, category, quantity
FROM
    products AS p
        JOIN
    order_items AS oi ON p.product_id = oi.product_id
ORDER BY quantity DESC
LIMIT 1;
```

	product_name	category	quantity
▶	Cotton T-Shirt	Clothing	3

QUESTION:6 LIST ORDERS WITH THEIR CUSTOMER NAMES AND ORDER DATES

```
SELECT
    first_name, order_date
FROM
    customers AS c
        JOIN
    orders AS o ON c.customer_id = o.customer_id;
```

	first_name	order_date
▶	John	2022-06-01
	John	2022-06-15
	Jane	2022-06-05
	Robert	2022-06-10
	Emily	2022-06-20
	Michael	2022-06-25

QUESTION:7 FIND PRODUCTS WITH LOW STOCK (LESS THAN 40)

```sql
SELECT
    product_name, stock_quantity
FROM
    products
WHERE
    stock_quantity < 40;
```

	product_name	stock_quantity
▶	Smart Watch	30
	Coffee Maker	25
	Bluetooth Speaker	35

QUESTION:8 CALCULATE TOTAL REVENUE BY PRODUCT CATEGORY

```sql
SELECT
    category,
    COUNT(quantity),
    SUM(unit_price * quantity) AS revenue
FROM
    products AS p
        JOIN
    order_items AS o ON p.product_id = o.product_id
GROUP BY category;
```

category	COUNT(quantity)	revenue
Electronics	3	479.96
Clothing	1	59.97
Footwear	1	89.99
Home	1	49.99
Accessories	1	39.99
Fitness	1	29.99

QUESTION:9 FIND CUSTOMERS WHO HAVEN'T PLACED ANY ORDERS

```sql
SELECT
    *
FROM
    customers AS c
        JOIN
    orders AS o ON c.customer_id = o.customer_id
WHERE
    order_date IS NULL;
```

IN MY CASE, EVERY CUSTOMER PLASED ORDER

Result Grid | Filter Rows: | Export: | Wrap Cell

customer_id	first_name	last_name	email	phone	address	city

QUESTION:10 LIST ALL ORDERS WITH THEIR ITEMS AND SUBTOTALS

```sql
SELECT
    product_name AS item_name, subtotal
FROM
    products AS p
        JOIN
    order_items AS oi ON p.product_id = oi.product_id;
```

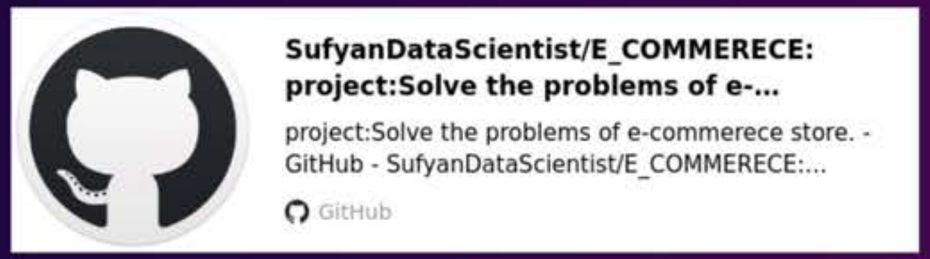
item_name	subtotal
Wireless Headphones	199.98
Smart Watch	199.99
Cotton T-Shirt	59.97
Running Shoes	89.99
Coffee Maker	49.99
Backpack	39.99
Bluetooth Speaker	79.99
Yoga Mat	29.99

QUESTION:11 FIND THE AVERAGE ORDER VALUE

```sql
SELECT
    ROUND(AVG(total_amount), 2) AS average_order_value
FROM
    orders;
```

average_order_value
126.65

"SEARCH THIS LINK TO GET THIS CODE."



GIT REMOTE ADD ORIGIN HTTPS://GITHUB.COM/SUFYANDATASCIENTIST/E_COMMERECE.GIT


GOOD
byes