**PROJECT 1**

**ShopFast**

**STEP 1: Creating a new Database,**

create database ShopFast;

**STEP 2: using the new database,**

use ShopFast;

**STEP 3: Inserting the values through given csv files,**

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**STEP 4: Check weather all the data’s are inserted properly,**

select \* from CUSTOMERS;

select \* from ORDER\_ITEMS;

select \* from ORDERS;

select \* from PRODUCTS;

A screenshot of a computer

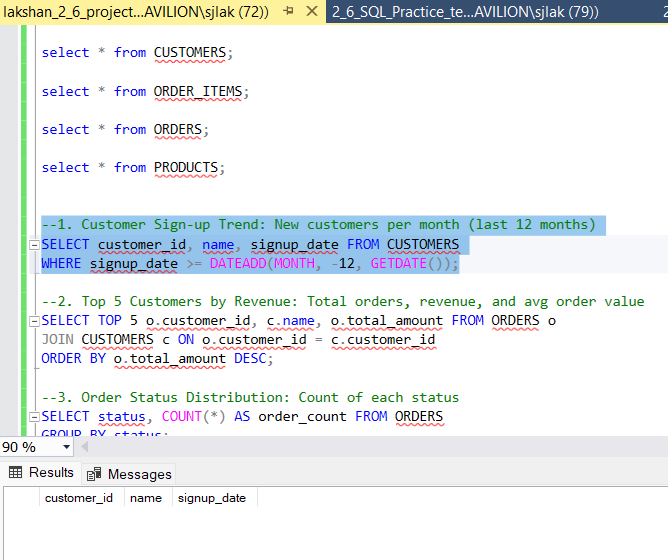
AI-generated content may be incorrect.

**STEP 5: Answers for all Problem Set**

--1. Customer Sign-up Trend: New customers per month (last 12 months)

SELECT customer\_id, name, signup\_date FROM CUSTOMERS

WHERE signup\_date >= DATEADD(MONTH, -12, GETDATE());



--2. Top 5 Customers by Revenue: Total orders, revenue, and avg order value

SELECT TOP 5 o.customer\_id, c.name, o.total\_amount FROM ORDERS o

JOIN CUSTOMERS c ON o.customer\_id = c.customer\_id

ORDER BY o.total\_amount DESC;

A screenshot of a computer

AI-generated content may be incorrect.

--3. Order Status Distribution: Count of each status

SELECT status, COUNT(\*) AS order\_count FROM ORDERS

GROUP BY status;

A screenshot of a computer

AI-generated content may be incorrect.

--4. Revenue by Category: Total revenue by category

SELECT o.product\_id, p.product\_name, o.quantity, o.price\_per\_unit

FROM ORDER\_ITEMS o

JOIN PRODUCTS p ON o.product\_id = p.product\_id;

A screenshot of a computer

AI-generated content may be incorrect.

--5. Best-Selling Products: Top 5 by quantity sold

SELECT o.product\_id, p.product\_name, o.quantity

FROM ORDER\_ITEMS o

JOIN PRODUCTS p ON o.product\_id = p.product\_id

ORDER BY o.quantity DESC;

A screenshot of a computer

AI-generated content may be incorrect.

--6. Low-Stock Products: Products with <10% stock using CASE

SELECT product\_name, stock\_quantity, CASE WHEN stock\_quantity < 10 THEN 'Low Stock' ELSE 'OK' END AS stock\_status

FROM PRODUCTS;

A screenshot of a computer

AI-generated content may be incorrect.

--7. Avg Delivery Time per Month

SELECT FORMAT(order\_date, 'yyyy-MM') AS order\_month, AVG(DATEDIFF(DAY, order\_date, delivery\_date)) AS avg\_delivery\_days

FROM ORDERS

WHERE delivery\_date IS NOT NULL

GROUP BY FORMAT(order\_date, 'yyyy-MM')

ORDER BY order\_month;

A screenshot of a computer

AI-generated content may be incorrect.

--8. Orders with Delivery >7 days

SELECT order\_id, order\_date, delivery\_date

FROM ORDERS

WHERE DATEDIFF(DAY, order\_date, delivery\_date) > 7;

A screenshot of a computer program

AI-generated content may be incorrect.

--9. Repeat Customers: More than 1 order

SELECT customer\_id FROM ORDERS GROUP BY customer\_id

HAVING COUNT(\*) > 1;

A screenshot of a computer

AI-generated content may be incorrect.

--10. Monthly Revenue Growth with LAG()

SELECT

FORMAT(order\_date, 'yyyy-MM') AS order\_month,

SUM(total\_amount) AS monthly\_revenue,

SUM(total\_amount) - LAG(SUM(total\_amount)) OVER (ORDER BY FORMAT(order\_date, 'yyyy-MM')) AS revenue\_growth

FROM ORDERS

WHERE total\_amount IS NOT NULL

GROUP BY FORMAT(order\_date, 'yyyy-MM')

ORDER BY order\_month;

A screenshot of a computer

AI-generated content may be incorrect.

--11. Cohort Analysis using CTE (signup year)

SELECT customer\_id, name, YEAR(signup\_date) AS signup\_year

FROM CUSTOMERS;

A screenshot of a computer

AI-generated content may be incorrect.

--12. Cancelled/Returned Product Revenue Loss

SELECT order\_id, status, total\_amount FROM ORDERS

WHERE status = 'Cancelled' OR status = 'Returned';

A screenshot of a computer

AI-generated content may be incorrect.

--13. Customer City Heatmap

SELECT customer\_id, city FROM CUSTOMERS;

A screenshot of a computer

AI-generated content may be incorrect.

--14. First & Last Order per Customer with ROW\_NUMBER()

WITH RankedOrders AS (

SELECT

customer\_id,

order\_id,

order\_date,

ROW\_NUMBER() OVER (PARTITION BY customer\_id ORDER BY order\_date ASC) AS rn\_asc,

ROW\_NUMBER() OVER (PARTITION BY customer\_id ORDER BY order\_date DESC) AS rn\_desc

FROM ORDERS

)

SELECT customer\_id, order\_id, order\_date, 'First Order' AS order\_type

FROM RankedOrders WHERE rn\_asc = 1

UNION

SELECT customer\_id, order\_id, order\_date, 'Last Order'

FROM RankedOrders WHERE rn\_desc = 1;

A screenshot of a computer

AI-generated content may be incorrect.

--15. NULL Handling: Orders with missing delivery/amount

SELECT order\_id, delivery\_date, total\_amount FROM ORDERS

WHERE delivery\_date IS NULL OR total\_amount IS NULL;

A screenshot of a computer

AI-generated content may be incorrect.