**SQL\_ASSIGNMENT\_3**

**Dataset: Sales Information**

You have been given a dataset containing information about sales transactions. The dataset includes the following columns:

* order\_id (integer): Unique identifier for each order.
* customer\_id (integer): Unique identifier for each customer.
* product\_id (integer): Unique identifier for each product.
* product\_name (string): Name of the product.
* quantity (integer): The quantity of the product sold.
* unit\_price (decimal): The unit price of the product.
* order\_date (date): The date when the order was placed.

**Table Structure:**

Create a table named sales with the following structure:

CREATE OR REPLACE TABLE SALES

(

ORDER\_ID INT PRIMARY KEY,

CUSTOMER\_ID INT,

PRODUCT\_ID INT,

PRODUCT\_NAME VARCHAR(50),

QUANTITY INT,

UNIT\_PRICE DECIMAL(10, 2),

ORDER\_DATE DATE

);

**Insert Data:**

Insert the following sample data into the sales table:

INSERT INTO SALES (ORDER\_ID, CUSTOMER\_ID, PRODUCT\_ID, PRODUCT\_NAME, QUANTITY, UNIT\_PRICE, ORDER\_DATE)

VALUES

(1, 101, 1, 'Widget A', 5, 10.00, '2023-01-15'),

(2, 102, 2, 'Widget B', 2, 12.50, '2023-01-16'),

(3, 103, 1, 'Widget A', 3, 10.00, '2023-01-16'),

(4, 104, 3, 'Widget C', 1, 15.75, '2023-01-17'),

(5, 105, 2, 'Widget B', 4, 12.50, '2023-01-17'),

(6, 106, 1, 'Widget A', 2, 10.00, '2023-01-18'),

(7, 107, 4, 'Widget D', 3, 20.00, '2023-01-18'),

(8, 108, 2, 'Widget B', 5, 12.50, '2023-01-19'),

(9, 109, 1, 'Widget A', 1, 10.00, '2023-01-19'),

(10, 101, 3, 'Widget C', 2, 15.75, '2023-01-20');

**Instructions:**

Write SQL queries to answer the following questions using the sales table:

1. **Retrieve the total sales quantity and revenue for each product.**

SELECT PRODUCT\_NAME,

SUM(QUANTITY) AS TOT\_SALES\_QUANTITY,

SUM(UNIT\_PRICE \* QUANTITY) AS REVENUE

FROM EMP\_DATABASE.PUBLIC.SALES

GROUP BY 1

ORDER BY 1;

A screenshot of a computer screen

Description automatically generated

1. **Find the total revenue for each customer.**

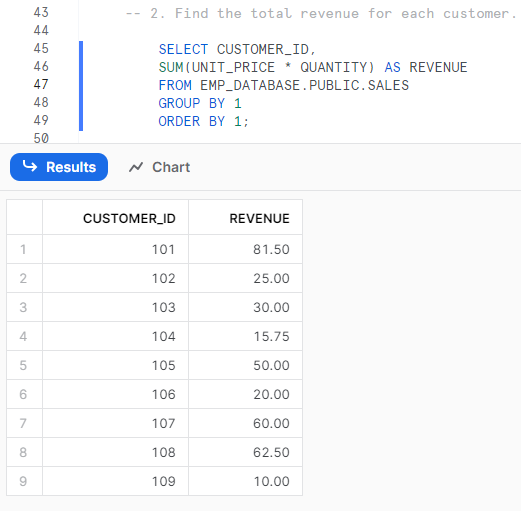
SELECT CUSTOMER\_ID,

SUM(UNIT\_PRICE \* QUANTITY) AS REVENUE

FROM EMP\_DATABASE.PUBLIC.SALES

GROUP BY 1

ORDER BY 1;



1. **Get the products with more than 10 units sold in a single order.**

SELECT PRODUCT\_NAME, ORDER\_ID,

SUM(QUANTITY) AS SOLD\_QTY

FROM EMP\_DATABASE.PUBLIC.SALES

GROUP BY 1,2

HAVING SOLD\_QTY > 10;

**So there is none such a record.**

A screenshot of a computer

Description automatically generated

1. **List the customers who have placed orders on at least three different dates.**

SELECT CUSTOMER\_ID, ORDER\_ID,

COUNT(DISTINCT ORDER\_DATE) AS TOT\_DIFF\_DATE\_ORDERS

FROM EMP\_DATABASE.PUBLIC.SALES

GROUP BY 1,2

HAVING TOT\_DIFF\_DATE\_ORDERS >= 3;

**So there is none such a record.**

A white screen with black text

Description automatically generated

1. **Calculate the average unit price of products.**

SELECT PRODUCT\_NAME,

AVG(UNIT\_PRICE) AS AVG\_UNIT\_PRICE

FROM EMP\_DATABASE.PUBLIC.SALES

GROUP BY 1

ORDER BY 1;

A screenshot of a computer

Description automatically generated

1. **Find the products with an average unit price greater than $12.00.**

SELECT PRODUCT\_NAME,

AVG(UNIT\_PRICE) AS AVG\_UNIT\_PRICE

FROM EMP\_DATABASE.PUBLIC.SALES

GROUP BY 1

HAVING AVG\_UNIT\_PRICE > 12.00

ORDER BY 1;

A screenshot of a computer

Description automatically generated

1. **Retrieve the customers who have spent more than $100.00 in total.**

SELECT CUSTOMER\_ID,

SUM(QUANTITY \* UNIT\_PRICE) AS TOT\_MONEY\_SPENT

FROM EMP\_DATABASE.PUBLIC.SALES

GROUP BY 1

HAVING TOT\_MONEY\_SPENT > 100.00

ORDER BY 1;

**So there is none such a record.**A screenshot of a computer

Description automatically generated

1. **List the customers who have purchased 'Widget B' and 'Widget A' in the same order.**

SELECT CUSTOMER\_ID,

COUNT(ORDER\_ID) AS TOT\_ORDERS

FROM EMP\_DATABASE.PUBLIC.SALES

WHERE PRODUCT\_NAME ='Widget B' AND PRODUCT\_NAME ='Widget A'

GROUP BY 1

HAVING TOT\_ORDERS >= 2

ORDER BY 1;

**So there is none such a record.**

A screenshot of a computer

Description automatically generated

**\*\*\*\*\*\*\*\*\*\*\*\*\*\* THANK YOU \*\*\*\*\*\*\*\*\*\*\*\*\*\*\***