Week04 – SQL - QUESTIONS

These questions and queries cover a wide range of scenarios commonly encountered in a MKTIME database, utilising joins, subqueries, and aggregate functions to extract meaningful output from the database.

1. List all products.

|  |  |  |
| --- | --- | --- |
| Code | Expected Answer | Actual Answer |
| SELECT \* FROM view\_items; | How many records you expect to display:  10 | 10 |

A screenshot of a computer

Description automatically generated

2. Find the total sales amount for each product.

SELECT view\_items.item\_name AS ITEM, *SUM*(view\_orders.quantity) AS QUANTITY FROM view\_orders, view\_items  
WHERE view\_orders.item\_id = view\_items.item\_id  
GROUP BY view\_orders.item\_id

A screenshot of a black screen

Description automatically generated

3. List all users who made purchase on 3rd May 2023.

SELECT view\_users.firstname AS NAME FROM view\_users, view\_orders  
WHERE view\_users.user\_id = view\_orders.user\_id AND view\_orders.order\_date ='2023-05-03'

A black screen with white text

Description automatically generated

4. Find the top 5 costing items.

SELECT view\_items.item\_name AS NAME, view\_items.item\_price AS PRICE FROM view\_items  
ORDER BY view\_items.item\_price DESC LIMIT 5

A screenshot of a black screen

Description automatically generated

5. List all items and who purchased those items.

SELECT view\_items.item\_name AS NAME, view\_users.firstname AS CUSTOMER FROM view\_items, view\_users, view\_orders  
WHERE view\_items.item\_id = view\_orders.item\_id AND view\_users.user\_id = view\_orders.user\_id

A screenshot of a computer

Description automatically generated

6. Find the total order value for each user.

SELECT view\_users.firstname AS CUSTOMER, *COUNT*(view\_orders.user\_id) AS ORDERS FROM view\_users, view\_orders  
WHERE view\_users.user\_id = view\_orders.user\_id  
GROUP BY view\_orders.user\_id

A screenshot of a computer

Description automatically generated

7. List all products with their corresponding orders.

SELECT view\_items.item\_name AS NAME, *COUNT*(view\_orders.item\_id) AS ORDERS FROM view\_items, view\_orders  
WHERE view\_items.item\_id = view\_orders.item\_id  
GROUP BY view\_orders.item\_id

A screenshot of a black menu

Description automatically generated

8. Find the customer who spent the most in total.

SELECT view\_users.firstname AS CUSTOMER, *SUM*(view\_orders.total) AS TOTAL FROM view\_users, view\_orders  
WHERE view\_users.user\_id= view\_orders.user\_id  
GROUP BY view\_users.firstname  
LIMIT 1

A screenshot of a computer

Description automatically generated

9. Find the top 3 categories with the highest total sales.

We do not have categories

11. List all orders made by a specific customer (e.g., John Doe).

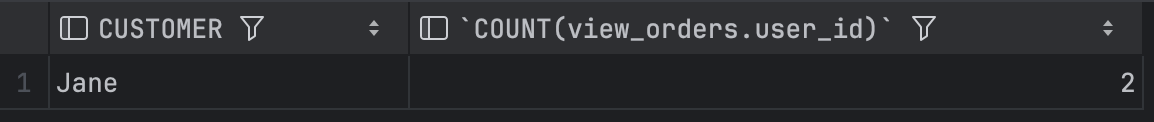
SELECT view\_users.firstname AS CUSTOMER,view\_orders.total, view\_orders.payment\_id  
FROM view\_users, view\_orders  
WHERE view\_users.user\_id= view\_orders.user\_id AND view\_users.firstname = 'John'

A screenshot of a graph

Description automatically generated

12. Find the number of orders placed by user\_id = 2.

SELECT view\_users.firstname AS CUSTOMER, *COUNT*(view\_orders.user\_id)  
FROM view\_users, view\_orders  
WHERE view\_users.user\_id= view\_orders.user\_id AND view\_users.user\_id = 2



13. List all items with their respective quantities sold.

SELECT view\_items.item\_name AS ITEM\_NAME, *SUM*(view\_orders.quantity)  
FROM view\_items, view\_orders  
WHERE view\_items.item\_id= view\_orders.item\_id  
GROUP BY view\_items.item\_name

A screenshot of a computer

Description automatically generated

14. Find the total sales made by each user.

SELECT view\_users.firstname AS CUSTOMER, *COUNT*(view\_orders.user\_id) AS SALES  
FROM view\_users, view\_orders  
WHERE view\_users.user\_id= view\_orders.user\_id  
GROUP BY view\_users.firstname

A screenshot of a black screen

Description automatically generated