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 items; | How many records you expect to display:  10 records |  |

2. Find the total sales amount for each product.

|  |  |  |
| --- | --- | --- |
| select item\_id, SUM(total) from orders GROUP BY item\_id; | Guess:  Item 1 = 44  Item 2 = 78 |  |

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

|  |  |  |
| --- | --- | --- |
| SELECT u.user\_id, u.firstname, u.lastname, o.order\_date FROM users u  JOIN orders o ON u.user\_id = o.user\_id WHERE o.order\_date = '2023-05-03'; | From the order date, find out which user made a purchase on 3rd May 2023 |  |

4. Find the top 5 costing items.

|  |  |  |
| --- | --- | --- |
| SELECT item\_id, item\_price from items ORDER BY item\_price DESC LIMIT 5; | Select the 5 more expensive items among a list of 5, organised in descendant order  5 records |  |

5. List all items and who purchased those items.

|  |  |  |
| --- | --- | --- |
| SELECT o.item\_id, i.item\_name, u.firstname, u.lastname from orders o inner join users u on u.user\_id = o.user\_id inner join items i on i.item\_id = o.item\_id ; | List of all sold items and showing users’ identity for each purchase  A same customer could be buying different items  Max 10 records |  |

6. Find the total order value for each user.

|  |  |  |
| --- | --- | --- |
| SELECT u.user\_id, firstname, lastname, *SUM*(o.total) FROM users u inner join orders o on o.user\_id = u.user\_id GROUP BY u.user\_id; | Adding the order values of each user to find their total order value  Max 10 records if all users made a purchase |  |

7. List all products with their corresponding orders.

|  |  |  |
| --- | --- | --- |
| select i.item\_id , i.item\_name, o.payment\_id AS Order\_number from items i inner join orders o on o.item\_id = i.item\_id ORDER BY i.item\_id ASC; | Selectin all items and their respective order number (originally payment\_id) |  |

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

|  |  |  |
| --- | --- | --- |
| SELECT u.user\_id, u.firstname, u.lastname, *SUM*(o.total) AS total\_spent FROM users u  INNER JOIN orders o ON o.user\_id = u.user\_id GROUP BY u.user\_id, u.firstname, u.lastname ORDER BY total\_spent DESC LIMIT 1; | Retrieving the name of the user having purchase the most in term of costs  1 record |  |

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

|  |  |  |
| --- | --- | --- |
| select i.item\_id , i.item\_name, *SUM*(o.total) AS Total\_sales from items i inner join orders o on o.item\_id = i.item\_id GROUP BY i.item\_id ORDER BY Total\_sales DESC LIMIT 3; | Organising the list of items so only the 3 highest total sales are shown  3 records |  |

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

|  |  |  |
| --- | --- | --- |
| SELECT o.item\_id, i.item\_name, u.firstname, u.lastname from orders o inner join users u on u.user\_id = o.user\_id inner join items i on i.item\_id = o.item\_id WHERE u.firstname LIKE 'John'; | Selecting all orders made by the same user by specifying his name  Max 10 records if the customer made a purchase of each item |  |

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

|  |  |  |
| --- | --- | --- |
| SELECT user\_id, *COUNT*(\*) AS number\_of\_orders FROM orders WHERE user\_id = 2; | Counts the number of row (orders) where user\_id =2 appears  1 record |  |

13. List all items with their respective quantities sold.

|  |  |  |
| --- | --- | --- |
| select i.item\_id , i.item\_name, *SUM*(o.quantity) AS Total\_quantity from items i  inner join orders o on o.item\_id = i.item\_id GROUP BY i.item\_id, i.item\_name ORDER BY i.item\_id ASC; | Getting a total of quantity for the same product  Max 10 records |  |

14. Find the total sales made by each user.

|  |  |  |
| --- | --- | --- |
| SELECT u.user\_id, u.firstname, u.lastname, *SUM*(o.total) AS total\_spent FROM users u  INNER JOIN orders o ON o.user\_id = u.user\_id GROUP BY u.user\_id, u.firstname, u.lastname; | Same query as number 6?  Max 10 records if each user has made a purchase |  |