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 items.

|  |
| --- |
| **select** \* **from** orders; |
| All records display (10 records) |
|  |

2. Find the total sales amount for each product.

|  |
| --- |
| **select** items.item\_id, items.item\_name, **sum**(items.item\_price)  **from**  orders **join** items **on** orders.item\_id = items.item\_id  **group** **by** items.item\_id  **order** **by** items.item\_id; |
| One record per item with item name and the sum of the item’s price over all the orders that contain that item. |
|  |

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

|  |
| --- |
| **select** users.user\_id, users.firstname, users.lastname  **from** users **join** payment **on** users.payment\_id = payment.payment\_id  **join** orders **on** payment.payment\_id = orders.payment\_id  **where** date(orders.order\_date) = '2023-05-03'; |
| The id, name and surname of the user(s) who made a purchase on 2023-05-03 |
|  |

4. Find the top 5 costing items.

|  |
| --- |
| **select** \* **from** items  **order** **by** item\_price **desc**  **limit** 5; |
| The records of the 5 most expensive items ordered by their price in descending order (the most expensive first) |
|  |

5. List all items and who purchased those items.

|  |
| --- |
| **select** items.item\_name, date(orders.order\_date) **as** "purchased on", concat(users.firstname, ' ', users.lastname) **as** "purchased from"  **from**      items **join** orders **on** items.item\_id = orders.item\_id  **join** users **on** orders.user\_id = users.user\_id  **order** **by** item\_name; |
| A list with the item’s name, date of purchase and user who purchased that item. |
|  |

6. Find the total order value for each user.

|  |
| --- |
| **select** users.user\_id, users.firstname, users.lastname, **sum**(orders.total) **as** 'total order'  **from**      users **join** orders **on** users.user\_id = orders.user\_id  **group** **by** users.user\_id  **order** **by** users.user\_id; |
| For each user, the sum of their order total over all the orders they made. |
|  |

7. List all products with their corresponding orders.

|  |
| --- |
| **select** items.item\_id, items.item\_name, date(orders.order\_date) **as** 'ordered on'  **from** items **join** orders **on** items.item\_id = orders.item\_id; |
| For each item, its name and the date it was ordered. |
|  |

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

|  |
| --- |
| **select** users.user\_id, users.firstname, users.lastname, **sum**(payment.payment\_amount) **as** 'total spent'  **from**      users **join** orders **on** users.user\_id = orders.user\_id  **join** payment **on** orders.payment\_id = payment.payment\_id  **group** **by** users.user\_id  **order** **by** **sum**(payment\_amount) **desc**  **limit** 1; |
| The user with the highest total payment amount. |
|  |

9. Find the top 3 categories with the highest total sales. (NOT SURE HOW TO DO IT HAS THERE IS NO CATEGORY COLUMN IN ANY OF THE TABLES)

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

|  |
| --- |
| **select** orders.order\_date, orders.total, orders.quantity  **from** orders **join** users **on** orders.user\_id = users.user\_id  **where** firstname = 'John' **and** lastname = 'Doe'; |
| The date, order total and order quantity of each of the orders made by the customer ‘John Doe’. |
|  |

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

|  |
| --- |
| **select** **count**(\*)  **from** orders **join** users **on** orders.user\_id = users.user\_id  **where** users.user\_id=2; |
| The number of orders made by that specific user. |
|  |

12. List all items with their respective quantities sold.

|  |
| --- |
| **select** items.item\_name, **sum**(orders.quantity) **as** 'Quantity sold' **from** items **join** orders **on** items.item\_id = orders.item\_id **group** **by** items.item\_name; |
| For each item, the aggregated total of they quantities sold in all orders. |
|  |

13. Find the total sales made by each user.

|  |
| --- |
| **select** users.user\_id, **sum**(payment.payment\_amount) **from**  users **join** orders **on** users.user\_id = orders.user\_id **join** payment **on** orders.payment\_id = payment.payment\_id **group** **by** users.user\_id; |
| For each user, their user\_id and the sum of all the payments made over all the orders they made *(not sure if this is what is meant by ‘total sales’)*. |
|  |