

Code Glossary

Module 2: Data Analysis with SQL on a Single Table

1. **To find the starting date of the orders from the dataset:**

```
SELECT MIN(order_date) FROM orders;
```

2. **To find the ending date of the orders from the dataset:**

```
SELECT MAX(order_date) FROM orders;
```

3. **To find the number of orders in the month of June from the dataset:**

```
SELECT COUNT(Order_ID) FROM orders  
WHERE order_date >= '2022-06-01' AND order_date <= '2022-06-30';
```

4. **To find the number of orders in a month using BETWEEN from the dataset:**

```
SELECT COUNT(Order_ID) FROM orders  
WHERE order_date BETWEEN '2022-08-01' AND '2022-08-31';
```

5. **To find the number total number of orders in each month using GROUP BY from the dataset and provide alias using AS:**

```
SELECT MONTH(order_date) AS Order_Month, COUNT(order_id) as  
OrderQuantity  
FROM orders  
GROUP BY MONTH(order_date);
```

6. **To find the number total number of orders in each month using GROUP BY from the dataset and sort it:**

```
SELECT MONTH(order_date) AS Order_Month, COUNT(order_id) AS  
OrderQuantity  
FROM orders  
GROUP BY Order_Month  
ORDER BY OrderQuantity desc;
```

7. **To find the total discount over the months from the dataset:**

```
SELECT MONTH(order_date) as month, SUM(discount) as TotalDiscount,  
SUM(final_price) AS TotalRevenue  
FROM orders  
GROUP BY month ORDER BY month;
```

8. **To round off unwanted values after a decimal in a column using ROUND():**

```
SELECT MONTH(order_date) as month, SUM(discount) as TotalDiscount,  
ROUND(SUM(final_price),0) AS TotalRevenue
```

```
FROM orders  
GROUP BY month ORDER BY month;
```

- 9. To find the percentage change in discount over the months from the dataset:**

```
SELECT MONTH(order_date) as month, SUM(final_price) as TotalRevenue,  
SUM(discount)/SUM(final_price) as Discount_Sales_Ratio SUM(discount) as  
TotalDiscount, COUNT(order_id) as OrderCount  
FROM orders  
GROUP BY month ORDER BY month;
```

- 10. To find the total revenue based on the day of the week from the dataset:**

```
SELECT DAYOFWEEK(order_date) as WDAY, SUM(final_price) AS  
total_revenue, COUNT(order_id) AS order_count  
FROM orders  
GROUP BY WDAY order by WDAY;
```

- 11. To find the total revenue based on the day of the week from the dataset using CASE statements:**

```
SELECT SUM(final_price) AS total_revenue, COUNT(order_id) AS order_count,  
CASE  
    WHEN DAYOFWEEK(order_date) = 1 THEN 'Weekend'  
    WHEN DAYOFWEEK(order_date) = 7 THEN 'Weekend'  
    ELSE 'Weekday'  
END AS WeekDay  
FROM orders  
GROUP BY WeekDay;
```