**Activity**

1. Write a query to find all customers who live in the zip code 20260.

**SELECT customer\_unique\_id, customer\_zip\_code\_prefix**

**FROM transactions**

**WHERE customer\_zip\_code\_prefix='20260';**

1. Write a select query to find all orders with a price greater than 50.

**SELECT order\_id, product\_id, price**

**FROM transactions**

**WHERE price > 50;**

1. Group the orders by payment\_type and count how many orders belong to each type.

**SELECT payment\_type, COUNT(\*) AS total\_orders**

**FROM transactions**

**GROUP BY payment\_type;**

1. Write a query to find all products that have been ordered more than once.

**SELECT product\_id, COUNT(\*) AS order\_count**

**FROM transactions**

**GROUP BY product\_id**

**HAVING COUNT(\*) > 1;**

1. Find the total freight\_value for each seller\_id.

**SELECT seller\_id, SUM(freight\_value) AS total\_freigh**

**FROM transactions**

**GROUP BY seller\_id;**

1. Write a query to find all products where the product\_name\_length is greater than 10.

**SELECT product\_id, product\_name\_lenght**

**FROM transactions**

**WHERE product\_name\_lenght > 10;**

1. Write a query to find the average price for each product\_id.

**SELECT product\_category\_name, AVG(price) AS average\_price**

**FROM transactions**

**GROUP BY product\_category\_name;**

1. Find all products with a price lower than the average price.

**SELECT product\_category\_name, price**

**FROM transactions**

**WHERE price < (SELECT AVG(price) FROM transactions);**

1. Count how many orders were made for each payment\_installments value.

**SELECT payment\_installments, COUNT(\*) AS order\_count**

**FROM transactions**

**GROUP BY payment\_installments;**

1. Write a query to retrieve a list of unique customer\_state values.

**SELECT DISTINCT customer\_state**

**FROM transactions;**

11. Write a query to calculate the total payment\_value for each customer

(customer\_unique\_id).

**SELECT customer\_unique\_id, SUM(payment\_value) AS total\_paymen**

**FROM transactions**

**GROUP BY customer\_unique\_id;**

1. Write a query to find seller\_id values with more than 10 orders.

**SELECT seller\_id, COUNT(order\_id) AS order\_count**

**FROM transactions**

**GROUP BY seller\_id**

**HAVING COUNT(order\_id) > 10;**

1. Find the total payment\_value for each customer\_unique\_id and order the results by the total payment in descending order.

**SELECT customer\_unique\_id, SUM(payment\_value) AS total\_payment\_value**

**FROM olist\_transactions**

**GROUP BY customer\_unique\_id**

**ORDER BY total\_payment\_value DESC;**

1. Write a query to find orders where the price is greater than 50 and the freight\_value is less than 10.

**SELECT order\_id, price, freight\_value**

**FROM transactions**

**WHERE price > 50 AND freight\_value < 10;**

1. Write a query to find sellers whose average price is greater than 100.

**SELECT seller\_id, AVG(price) AS average\_price**

**FROM transactions**

**GROUP BY seller\_id**

**HAVING AVG(price) > 100;**