```sql
SHOW VOLUMES IN workspace.default;


-- 'customer_id', 'age', 'gender', 'item_purchased', 'category',
-- 'purchase_amount', 'location', 'size', 'color', 'season',
-- 'review_rating', 'subscription_status', 'shipping_type',
-- 'discount_applied', 'previous_purchases', 'payment_method',
-- 'frequency_of_purchases', 'age_group', 'purchase_frequency_days'


SELECT * FROM customer_data LIMIT 20;


--Q1. What is the total revenue generated by male vs. female customers?

SELECT
gender,
sum(purchase_amount) FROM customer_data
WHERE gender = 'Male'
GROUP BY gender
UNION ALL
SELECT
gender,
sum(purchase_amount)
FROM customer_data
WHERE gender = 'Female'
GROUP BY gender;

SELECT
gender,
sum(purchase_amount) FROM customer_data
GROUP BY gender;

--Q2. Which customers used a discount but still spent more than the average
purchase amount?

SELECT
customer_id
FROM customer_data
WHERE discount_applied = 'Yes'
AND
```

```sql
purchase_amount > (SELECT avg(purchase_amount) FROM customer_data);

--Q3. Which are the top 5 products with the highest average review rating?

SELECT
item_purchased ,
avg(review_rating) as avg_review_rating
FROM customer_data
GROUP BY item_purchased
ORDER BY avg(review_rating) DESC LIMIT 5;

--Q4. Compare the average Purchase Amounts between Standard and Express
Shipping.

select shipping_type, avg(purchase_amount) FROM customer_data  WHERE
shipping_type = 'Standard' GROUP BY shipping_type UNION ALL SELECT
shipping_type, avg(purchase_amount) FROM customer_data WHERE shipping_type =
'Express' GROUP BY shipping_type;

select
shipping_type,
avg(purchase_amount)
FROM customer_data
-- WHERE shipping_type = 'Standard' OR shipping_type = 'Express'
WHERE shipping_type IN ('Standard', 'Express')
GROUP BY shipping_type;

--Q5. Do subscribed customers spend more? Compare average spend and total
revenue between subscribers and non-subscribers.

-- subscription type Yes/No

select
subscription_status,
count(distinct customer_id) as total_customers,
avg(purchase_amount) as avg_purchase_amount,
sum(purchase_amount) as total_revenue
FROM customer_data
GROUP BY subscription_status;
```

```sql
--Q6. Which 5 products have the highest percentage of purchases with
discounts applied?

SELECT
   item_purchased,
   COUNT(CASE WHEN discount_applied = 'Yes' THEN 1 END) AS
discounted_purchases,
   COUNT(*) AS total_purchases,
   100.0 * COUNT(CASE WHEN discount_applied = 'Yes' THEN 1 END) / COUNT(*) AS
discount_percentage
FROM customer_data
GROUP BY item_purchased
ORDER BY discount_percentage DESC
LIMIT 5;

--Q7. Segment customers into New, Returning, and Loyal based on their total
number of previous purchases, and show the count of each segment.

SELECT
purchase_frequency_days
FROM customer_data
GROUP BY purchase_frequency_days;
-- 90, 14, 365, 7, 30

-- previous_purchases number of previous purchases

with customer_type as (
SELECT
customer_id, previous_purchases,
CASE
WHEN previous_purchases = 1 THEN 'New'
WHEN previous_purchases BETWEEN 2 AND 10 THEN 'Returning'
ELSE 'LOYAL'
END AS customer_type
FROM customer_data)
SELECT
customer_type,
count(customer_id) as total_customers
FROM customer_type
GROUP BY customer_type;
```

```sql
--Q8. What are the top 3 most purchased products within each category?

-- 'item_purchased', 'category',
SELECT
   category,
   item_purchased,
   cnt,
   row_number() OVER (PARTITION BY category ORDER BY cnt DESC) AS
rank_in_category
FROM (
   SELECT
       category,
       item_purchased,
       COUNT(*) AS cnt
   FROM customer_data
   GROUP BY category, item_purchased
) as t;


--Q9. Are customers who are repeat buyers (more than 5 previous purchases)
also likely to subscribe?

--  repeat_buyers != buying ->  subsciption -> subscrption !=
valuableForCustomer

SELECT
subscription_status,
COUNT(DISTINCT customer_id) AS unique_customers
FROM customer_data
WHERE previous_purchases > 5
GROUP BY subscription_status;

--Q10. What is the revenue contribution of each age group?
-- age_group

SELECT
   age_group,
   SUM(purchase_amount) AS purchase_amount
FROM customer_data
GROUP BY age_group
```

```
ORDER BY purchase_amount DESC;
```