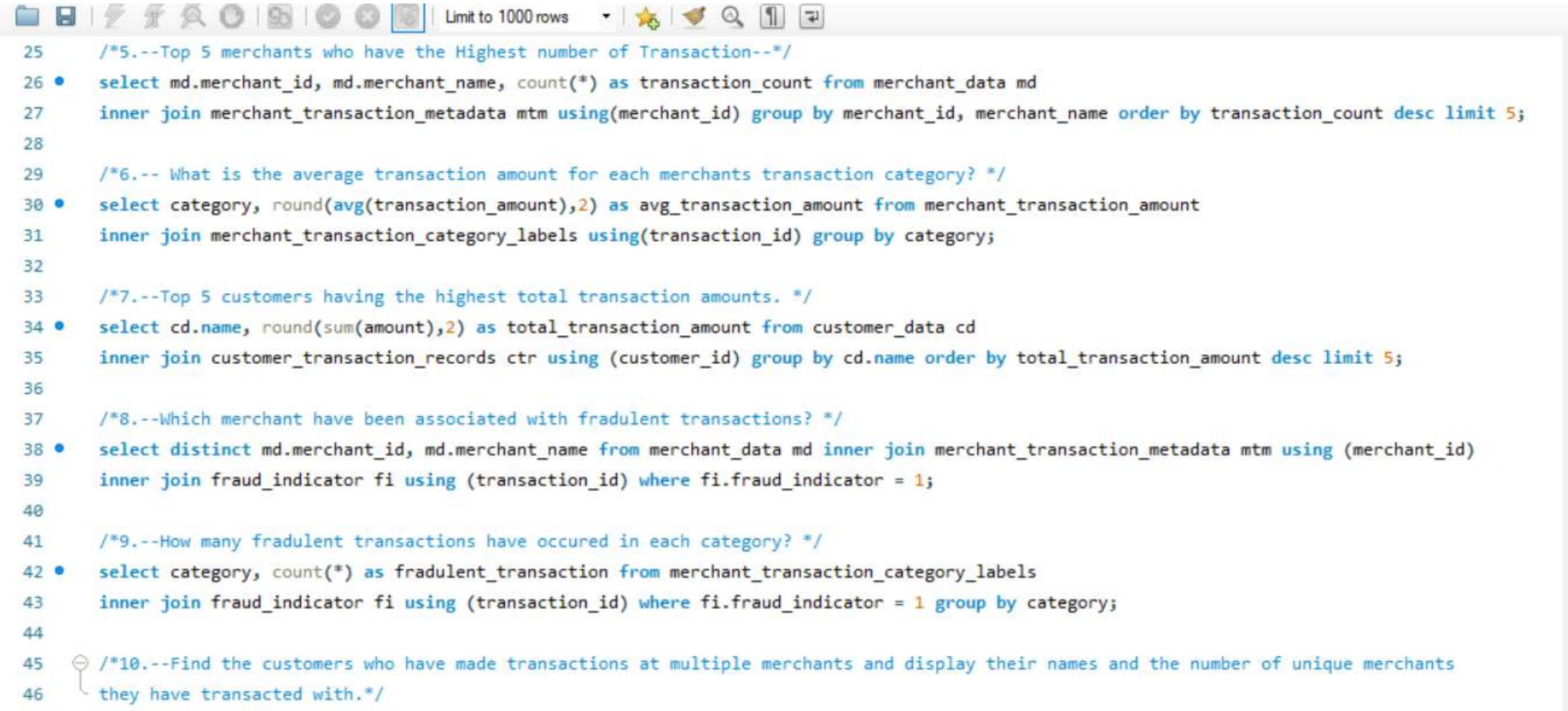

```sql
create database fraud_detection;
use fraud_detection;
/*1.--How many Unique Customers are there in the dataset? */
select count(distinct customer_id) as unique_customers from customer_data;
select distinct customer_id as unique_customers from customer_data;

/*2.--Which customer have the highest & lowest account balance? */
select customer_id, account_balance from customer_account_activity
where account_balance = (select max(account_balance) from customer_account_activity);
select customer_id, account_balance from customer_account_activity
where account_balance = (select min(account_balance) from customer_account_activity);

/*3.--What is the distribution of customer ages in the dataset? */
select
    case when age between 18 and 30 then '18-30'
         when age between 31 and 45 then '31-45'
         when age between 46 and 60 then '43-60'
         else '61+'
         end as age_group, count(*) as customer_count from customer_data group by age_group order by age_group;

/*4.--How many customer are engaged in suspicious activity? */
select count(distinct customer_id) as suspicious_transactions from customer_suspisious_activity
where suspisious_flag = '1';

```


Limit to 1000 rows
/*5.--Top 5 merchants who have the Highest number of Transaction--*/
select md.merchant_id, md.merchant_name, count(*) as transaction_count from merchant_data md
inner join merchant_transaction_metadata mtm using(merchant_id) group by merchant_id, merchant_name order by transaction_count desc limit 5;
/*6.-- What is the average transaction amount for each merchants transaction category? */
select category, round(avg(transaction_amount),2) as avg_transaction_amount from merchant_transaction_amount
inner join merchant_transaction_category_labels using(transaction_id) group by category;
/*7.--Top 5 customers having the highest total transaction amounts. */
select cd.name, round(sum(amount),2) as total_transaction_amount from customer_data cd
inner join customer_transaction_records ctr using (customer_id) group by cd.name order by total_transaction_amount desc limit 5;
/*8.--Which merchant have been associated with fradulent transactions? */
select distinct md.merchant_id, md.merchant_name from merchant_data md inner join merchant_transaction_metadata mtm using (merchant_id)
inner join fraud_indicator fi using (transaction_id) where fi.fraud_indicator = 1;
/*9.--How many fradulent transactions have occured in each category? */
select category, count(*) as fradulent_transaction from merchant_transaction_category_labels
inner join fraud_indicator fi using (transaction_id) where fi.fraud_indicator = 1 group by category;
/*10.--Find the customers who have made transactions at multiple merchants and display their names and the number of unique merchants
they have transacted with.*/


```
43  inner join fraud_indicator fi using (transaction_id) where fi.fraud_indicator = 1 group by category;
44
45  /*10.--Find the customers who have made transactions at multiple merchants and display their names and the number of unique merchants
46  they have transacted with.*/
47  select cd.name as customer_name, count(distinct md.merchant_id) as unique_merchant from customer_data cd inner join customer_transaction_record
48  using (customer_id) inner join merchant_transaction_metadata mtm using (transaction_id) inner join merchant_data md using (merchant_id)
49  group by cd.name having unique_merchant >1;
50
51  /*11.--What is the average transaction amount for fraudulent transactions compared to non-fraudulent transactions? */
52  select fi.fraud_indicator, round(avg(amount),2) as average_transaction_amount
53  from fraud_indicator fi
54  inner join customer_transaction_records ctr using (transaction_id)
55  group by fraud_indicator;
56
57  /*12.--Are there any regional patterns in fraudulent transactions? */
58  select md.location, count(*) as fraudulent_transaction from merchant_data md
59  inner join merchant_transaction_metadata mtm using (merchant_id)
60  inner join fraud_indicator fi using (transaction_id)
61  where fraud_indicator = 1
62  group by md.location
63  order by fraudulent_transaction desc;
64
65
66
```