Customer Behavior (SQL code)

```sql
--1. Revenue by Gender
Select gender, sum(purchase_amount) as Revenue
from customer
GROUP BY 1
ORDER BY 2 DESC;

--2. High-Spending Discount Users
SELECT customer_id, purchase_amount
from customer
where purchase_amount >= (SELECT  avg(purchase_amount) as avg
FROM customer) and discount_applied like 'Yes'

--3. Top 5 Products by Rating
SELECT item_purchased, round(avg(review_rating::numeric),2) as average
FROM customer
group by 1
order by 2 desc Limit 5;

--4. Shipping Type Comparison
SELECT shipping_type, round(avg(purchase_amount::numeric),2) as average
from customer
where shipping_type in ('Standard','Express')
group by 1;

--5. Subscribers vs. Non-Subscribers
SELECT subscription_status, count(customer_id) as customers,
round(avg(purchase_amount::numeric),2) as avg, sum(purchase_amount) as amount
from customer
group by 1;

--6. Discount-Dependent Products
SELECT item_purchased, ROUND(100.0 * SUM(CASE WHEN discount_applied = 'Yes' THEN
1 ELSE 0 END)/COUNT(*),2) AS discount_rate
FROM customer
group by 1
order by 2 desc limit 5

--7. Customer Segmentation
SELECT customer_segment, count(*) as number
from (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_segment
```

Customer Behavior (SQL code)

```sql
                            from customer)
group by 1
order by 2 desc


--8. Top 3 Products per Category
SELECT *
FROM (SELECT Row_number() OVER (partition by category ORDER BY
count(item_purchased) desc ) AS ranks, category, item_purchased, count(item_purchased) as
total
from customer
group by 2,3)
WHERE ranks <=3


--9. Repeat Buyers & Subscriptions
SELECT subscription_status, count(previous_purchases)
from customer
where previous_purchases >5
group by 1


--10. Revenue by Age Group
SELECT age_group, sum(purchase_amount) as amount
from customer
group by 1
order by 2 desc
```