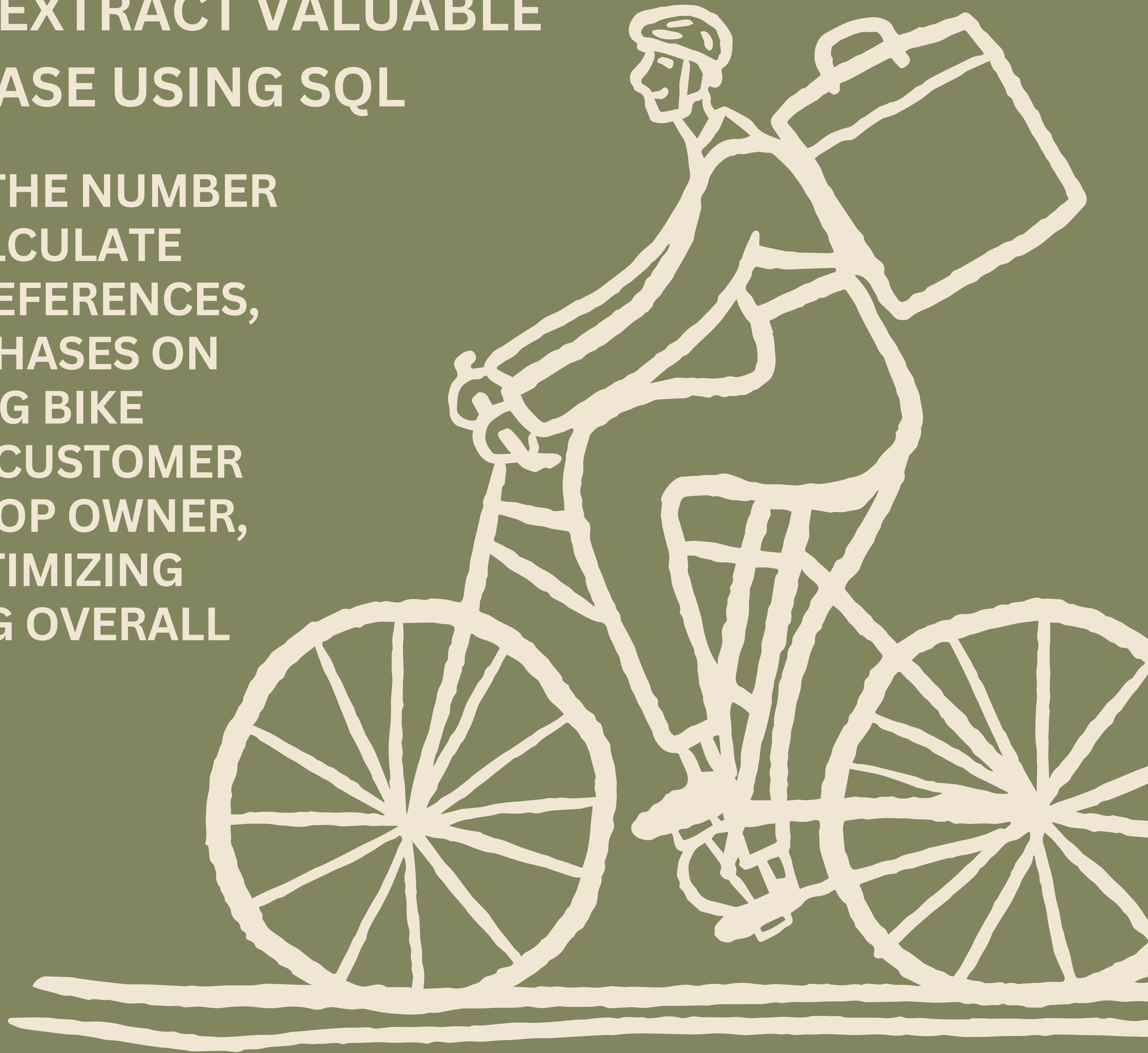
# Bike Rental Shop Analysis

SQL PROJECT

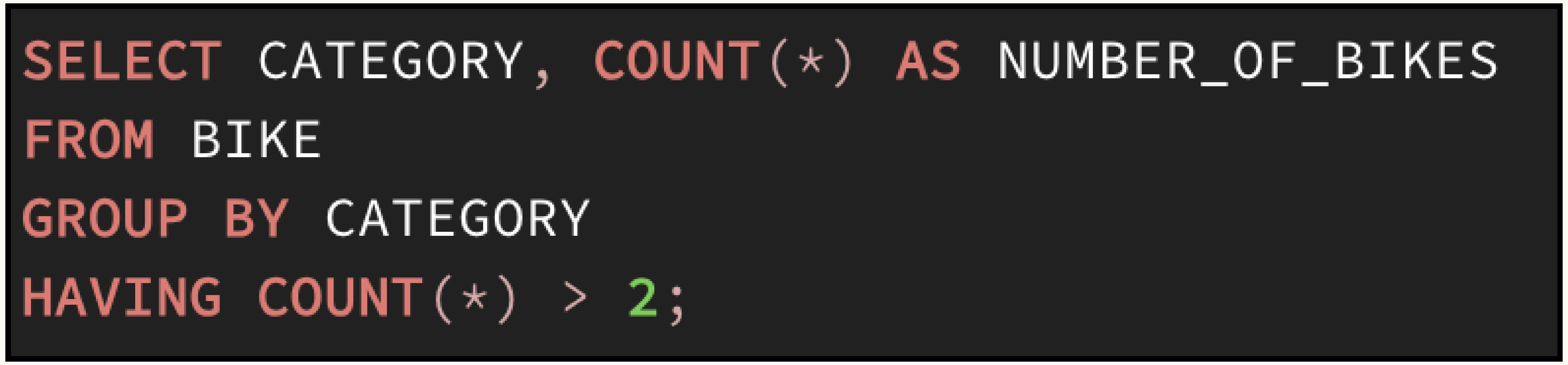
# OBJECTIVE

**THE OBJECTIVE OF THIS PROJECT IS TO ANALYZE AND EXTRACT VALUABLE INSIGHTS FROM THE BIKE RENTAL SHOP'S DATABASE USING SQL**

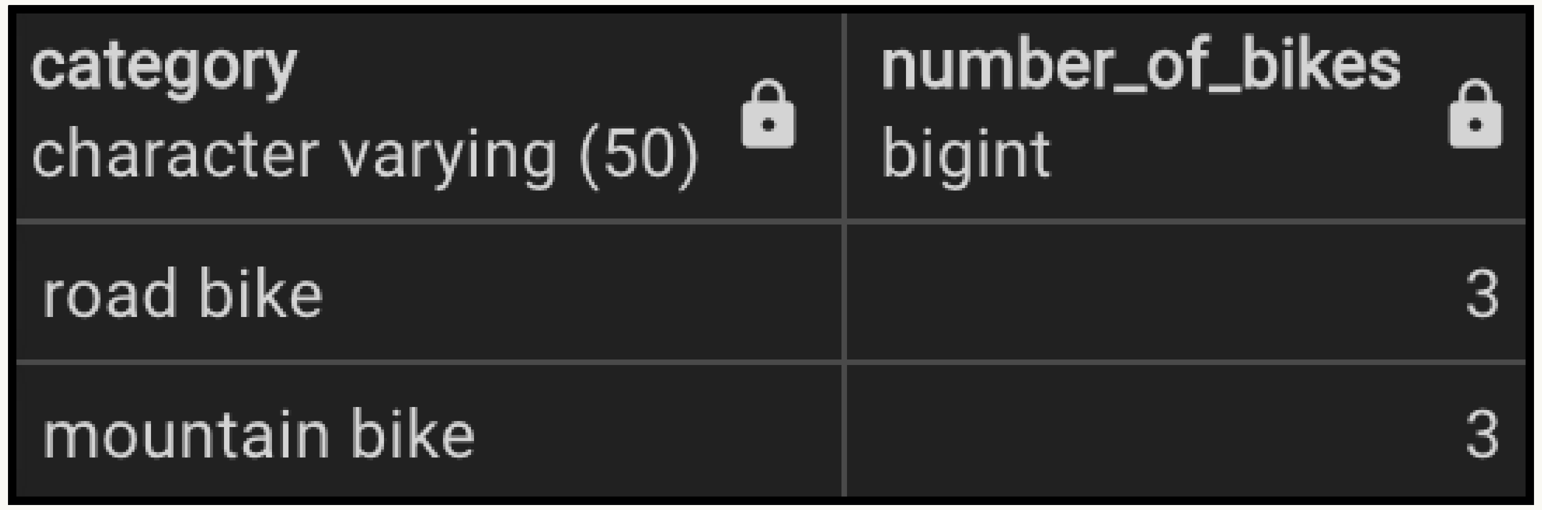
BY QUERYING THE DATA, THE GOAL IS TO DETERMINE THE NUMBER OF BIKES OWNED BY THE SHOP BY CATEGORY, CALCULATE MONTHLY RENTAL REVENUES, ASSESS CUSTOMER PREFERENCES, AND EVALUATE THE IMPACT OF MEMBERSHIP PURCHASES ON RENTAL ACTIVITY. THIS WILL INVOLVE ANALYZING BIKE INVENTORY, RENTAL DURATIONS, PRICING MODELS, CUSTOMER BEHAVIOR, AND MEMBERSHIP DATA TO HELP THE SHOP OWNER, EMILY, MAKE INFORMED DECISIONS AIMED AT OPTIMIZING INVENTORY, INCREASING REVENUE, AND IMPROVING OVERALL BUSINESS OPERATIONS.



## EMILY WOULD LIKE TO KNOW HOW MANY BIKES THE SHOP OWNS BY CATEGORY WHERE THE NUMBER OF BIKES IS MORE THAN TWO

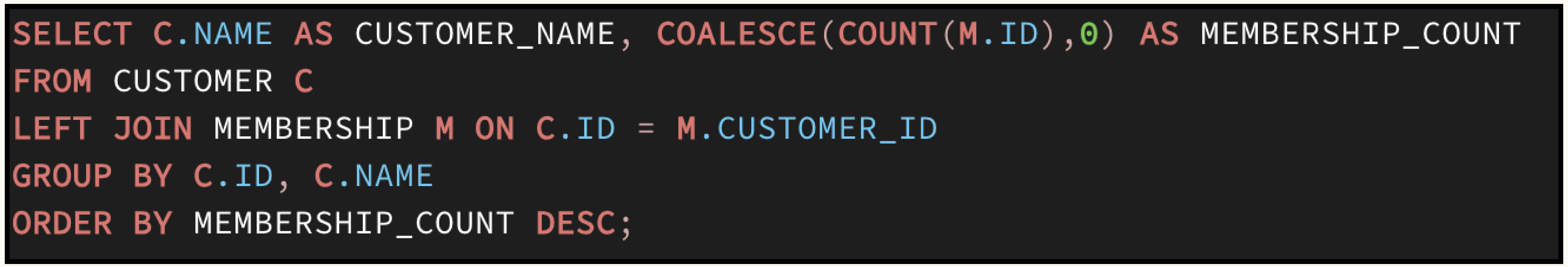


```sql
SELECT CATEGORY, COUNT(*) AS NUMBER_OF_BIKES
FROM BIKE
GROUP BY CATEGORY
HAVING COUNT(*) > 2;
```

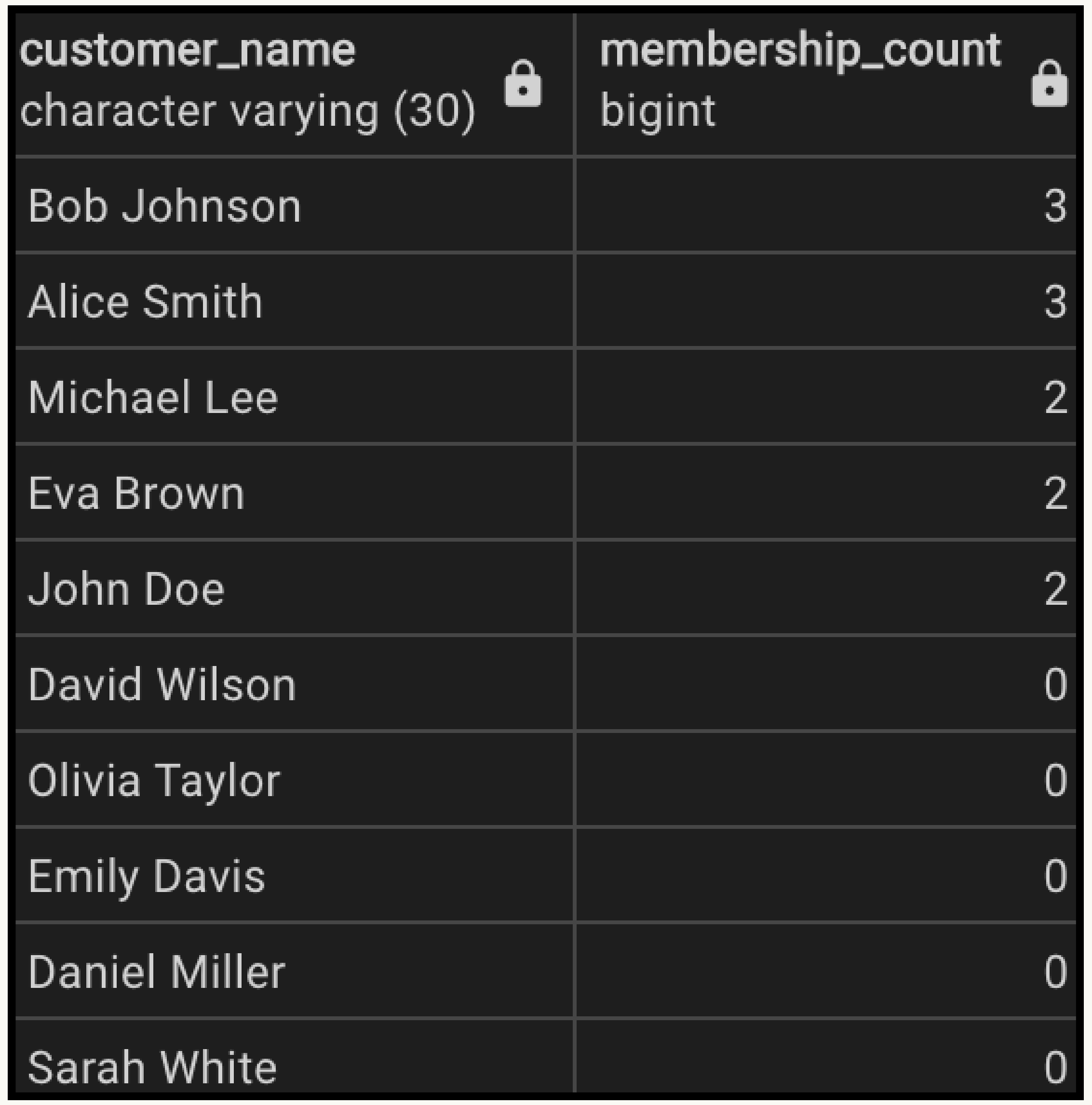


| category<br>character varying (50) | number_of_bikes<br>bigint |
|---|---|
| road bike | 3 |
| mountain bike | 3 |

## EMILY NEEDS A LIST OF CUSTOMER NAMES WITH THE TOTAL NUMBER OF MEMBERSHIPS PURCHASED BY EACH



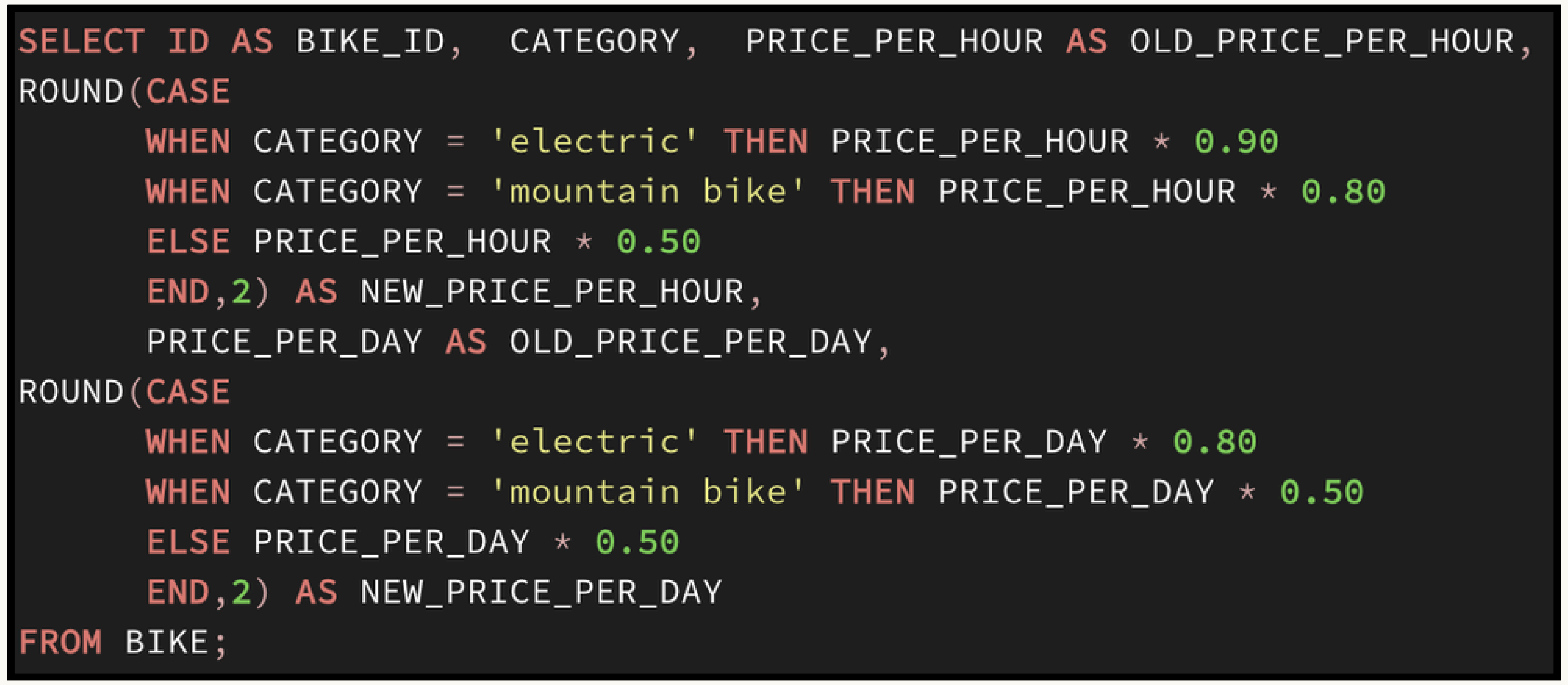
```sql
SELECT C.NAME AS CUSTOMER_NAME, COALESCE(COUNT(M.ID),0) AS MEMBERSHIP_COUNT
FROM CUSTOMER C
LEFT JOIN MEMBERSHIP M ON C.ID = M.CUSTOMER_ID
GROUP BY C.ID, C.NAME
ORDER BY MEMBERSHIP_COUNT DESC;
```



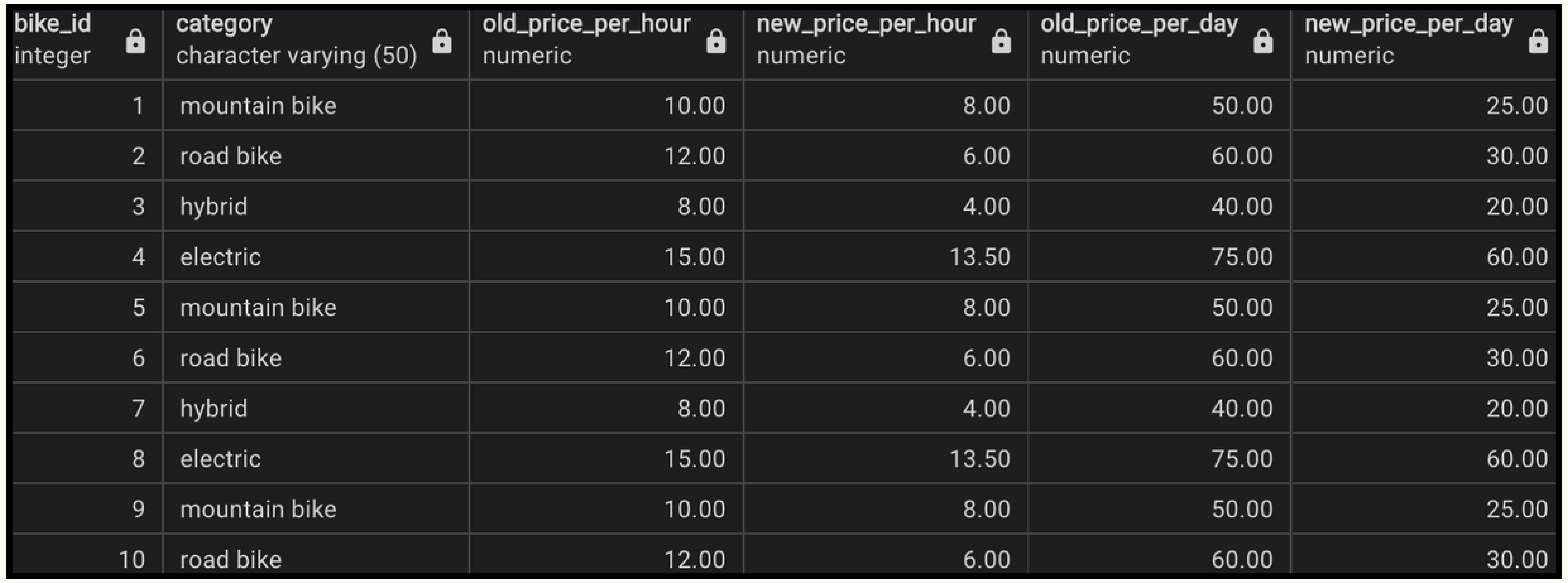
| customer_name character varying (30) | membership_count bigint |
|---|---|
| Bob Johnson | 3 |
| Alice Smith | 3 |
| Michael Lee | 2 |
| Eva Brown | 2 |
| John Doe | 2 |
| David Wilson | 0 |
| Olivia Taylor | 0 |
| Emily Davis | 0 |
| Daniel Miller | 0 |
| Sarah White | 0 |

# EMILY IS WORKING ON A SPECIAL OFFER FOR THE WINTER MONTHS

Electric bikes should have a 10% discount for hourly rentals and a 20% discount for daily rentals. Mountain bikes should have a 20% discount for hourly rentals and a 50% discount for daily rentals. All other bikes should have a 50% discount for all types of rentals.

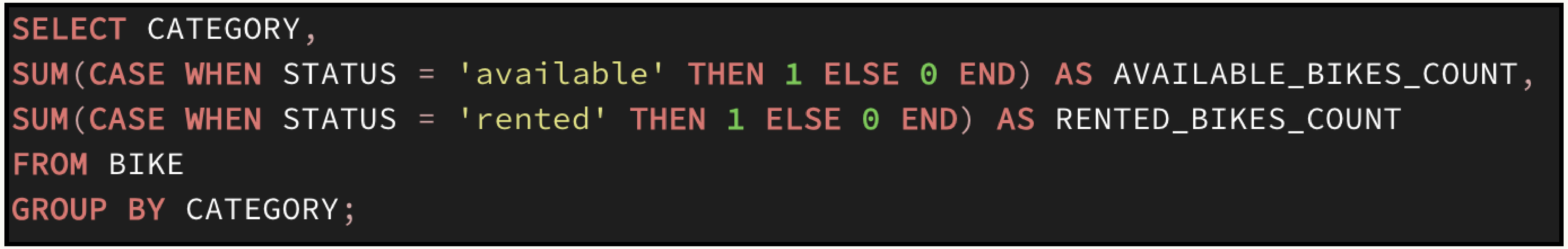


```
SELECT ID AS BIKE_ID,  CATEGORY,  PRICE_PER_HOUR AS OLD_PRICE_PER_HOUR,
ROUND(CASE
      WHEN CATEGORY = 'electric' THEN PRICE_PER_HOUR * 0.90
      WHEN CATEGORY = 'mountain bike' THEN PRICE_PER_HOUR * 0.80
      ELSE PRICE_PER_HOUR * 0.50
      END,2) AS NEW_PRICE_PER_HOUR,
      PRICE_PER_DAY AS OLD_PRICE_PER_DAY,
ROUND(CASE
      WHEN CATEGORY = 'electric' THEN PRICE_PER_DAY * 0.80
      WHEN CATEGORY = 'mountain bike' THEN PRICE_PER_DAY * 0.50
      ELSE PRICE_PER_DAY * 0.50
      END,2) AS NEW_PRICE_PER_DAY
FROM BIKE;
```

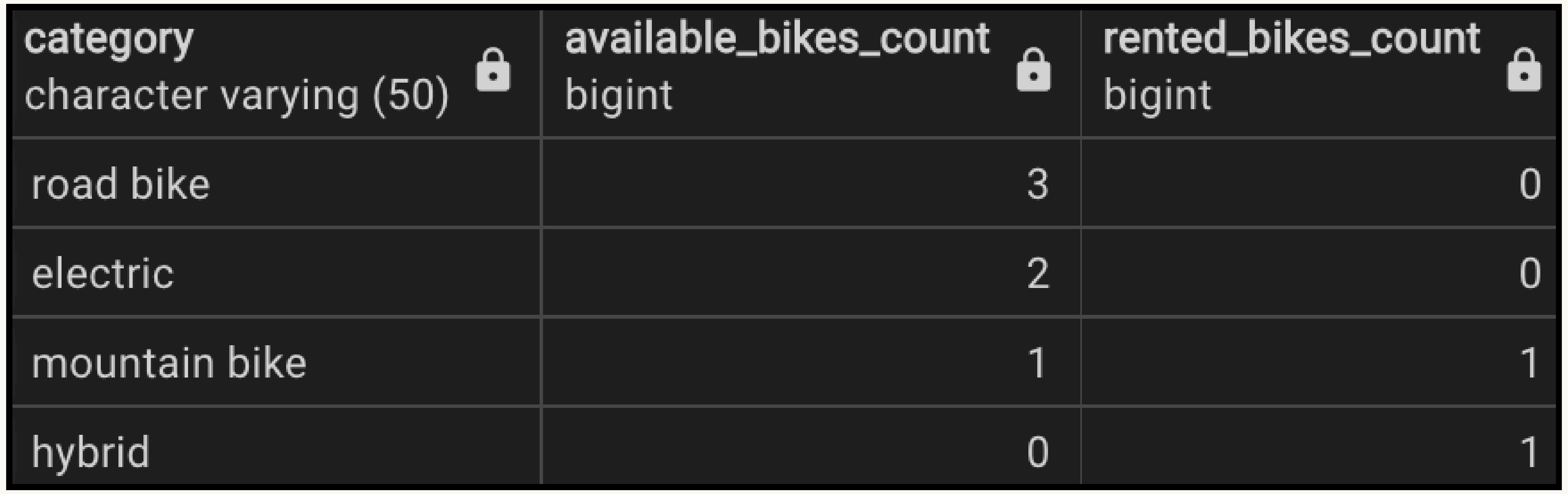


| bike_id integer | category character varying (50) | old_price_per_hour numeric | new_price_per_hour numeric | old_price_per_day numeric | new_price_per_day numeric |
|---|---|---|---|---|---|
| 1 | mountain bike | 10.00 | 8.00 | 50.00 | 25.00 |
| 2 | road bike | 12.00 | 6.00 | 60.00 | 30.00 |
| 3 | hybrid | 8.00 | 4.00 | 40.00 | 20.00 |
| 4 | electric | 15.00 | 13.50 | 75.00 | 60.00 |
| 5 | mountain bike | 10.00 | 8.00 | 50.00 | 25.00 |
| 6 | road bike | 12.00 | 6.00 | 60.00 | 30.00 |
| 7 | hybrid | 8.00 | 4.00 | 40.00 | 20.00 |
| 8 | electric | 15.00 | 13.50 | 75.00 | 60.00 |
| 9 | mountain bike | 10.00 | 8.00 | 50.00 | 25.00 |
| 10 | road bike | 12.00 | 6.00 | 60.00 | 30.00 |

# EMILY IS LOOKING FOR COUNTS OF THE RENTED BIKES AND OF THE AVAILABLE BIKES IN EACH CATEGORY

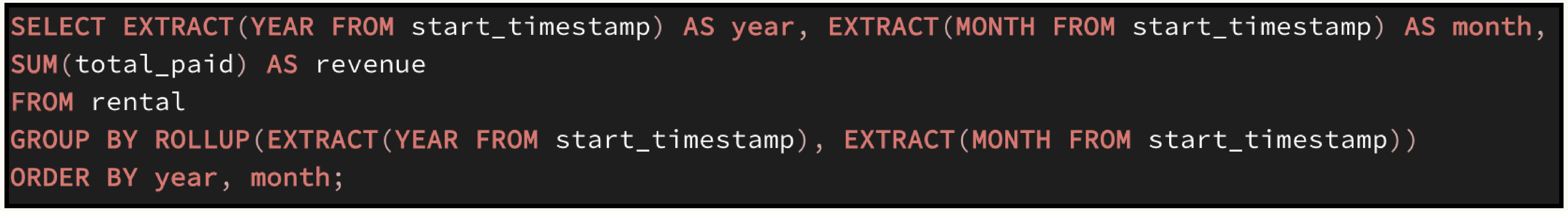


```sql
SELECT CATEGORY,
SUM(CASE WHEN STATUS = 'available' THEN 1 ELSE 0 END) AS AVAILABLE_BIKES_COUNT,
SUM(CASE WHEN STATUS = 'rented' THEN 1 ELSE 0 END) AS RENTED_BIKES_COUNT
FROM BIKE
GROUP BY CATEGORY;
```

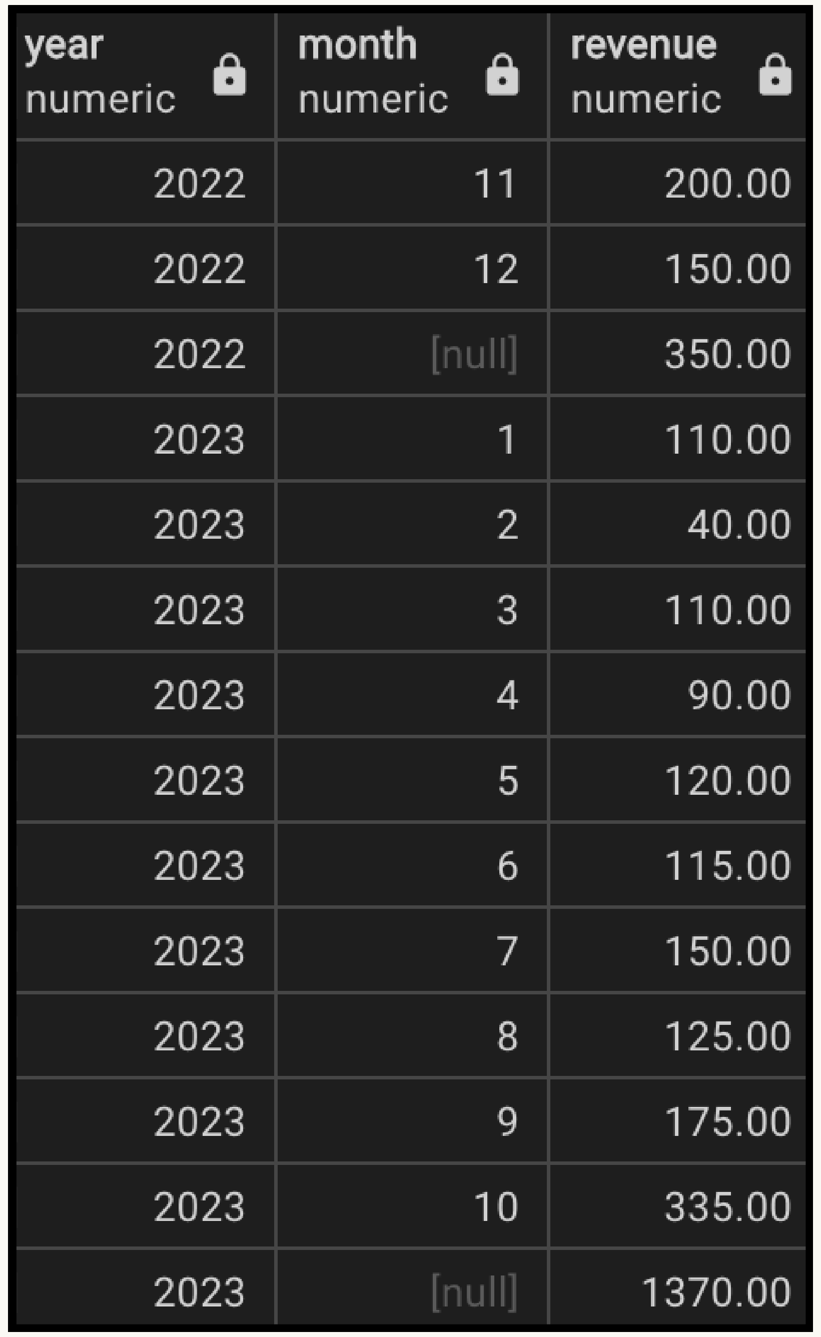


| category<br>character varying (50) | available_bikes_count<br>bigint | rented_bikes_count<br>bigint |
|---|---:|---:|
| road bike | 3 | 0 |
| electric | 2 | 0 |
| mountain bike | 1 | 1 |
| hybrid | 0 | 1 |

# EMILY IS PREPARING A SALES REPORT. SHE NEEDS TO KNOW THE TOTAL REVENUE FROM RENTALS BY MONTH, THE TOTAL BY YEAR, AND THE ALL-TIME ACROSS ALL THE YEARS

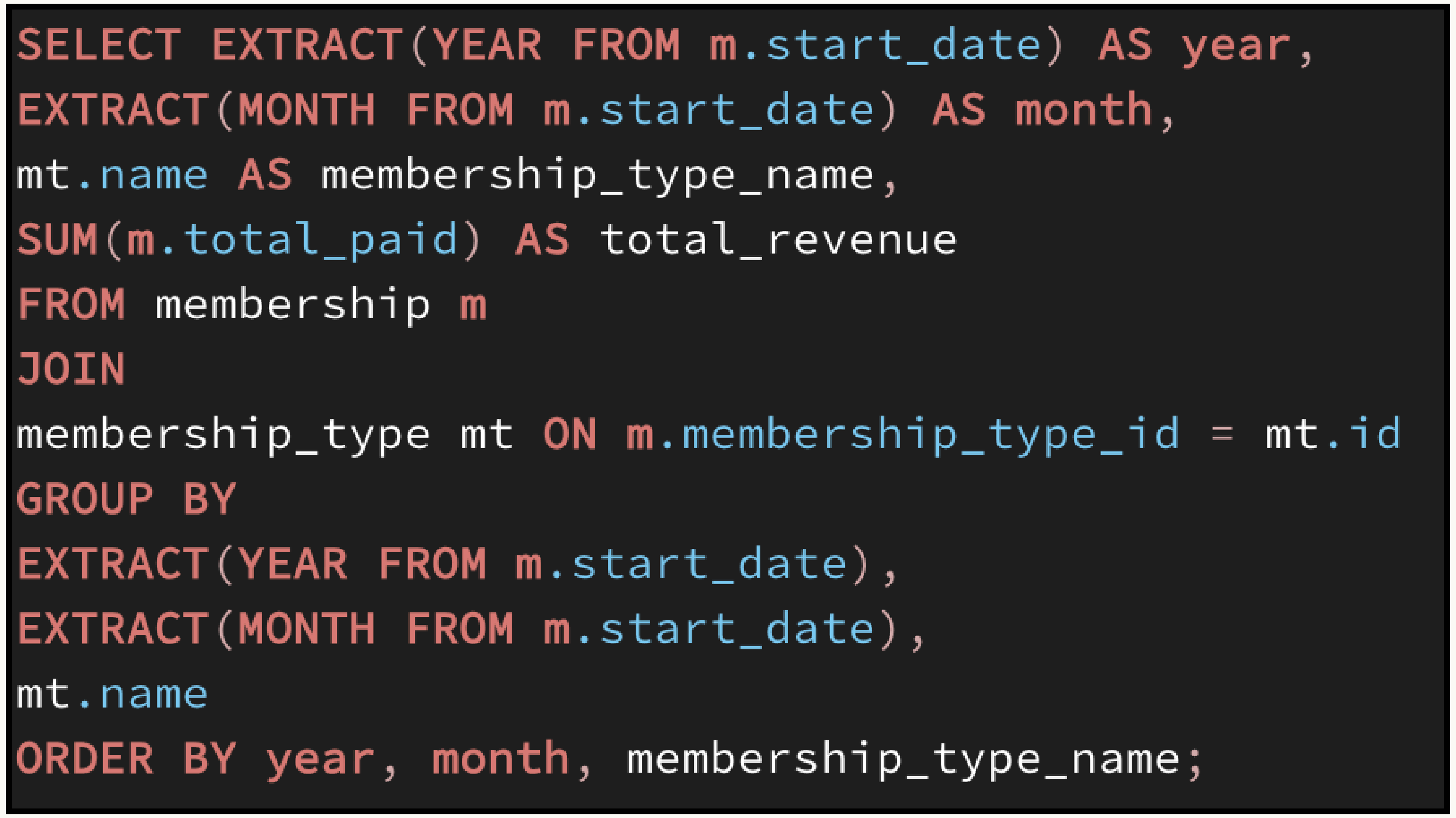


```sql
SELECT EXTRACT(YEAR FROM start_timestamp) AS year, EXTRACT(MONTH FROM start_timestamp) AS month,
SUM(total_paid) AS revenue
FROM rental
GROUP BY ROLLUP(EXTRACT(YEAR FROM start_timestamp), EXTRACT(MONTH FROM start_timestamp))
ORDER BY year, month;
```

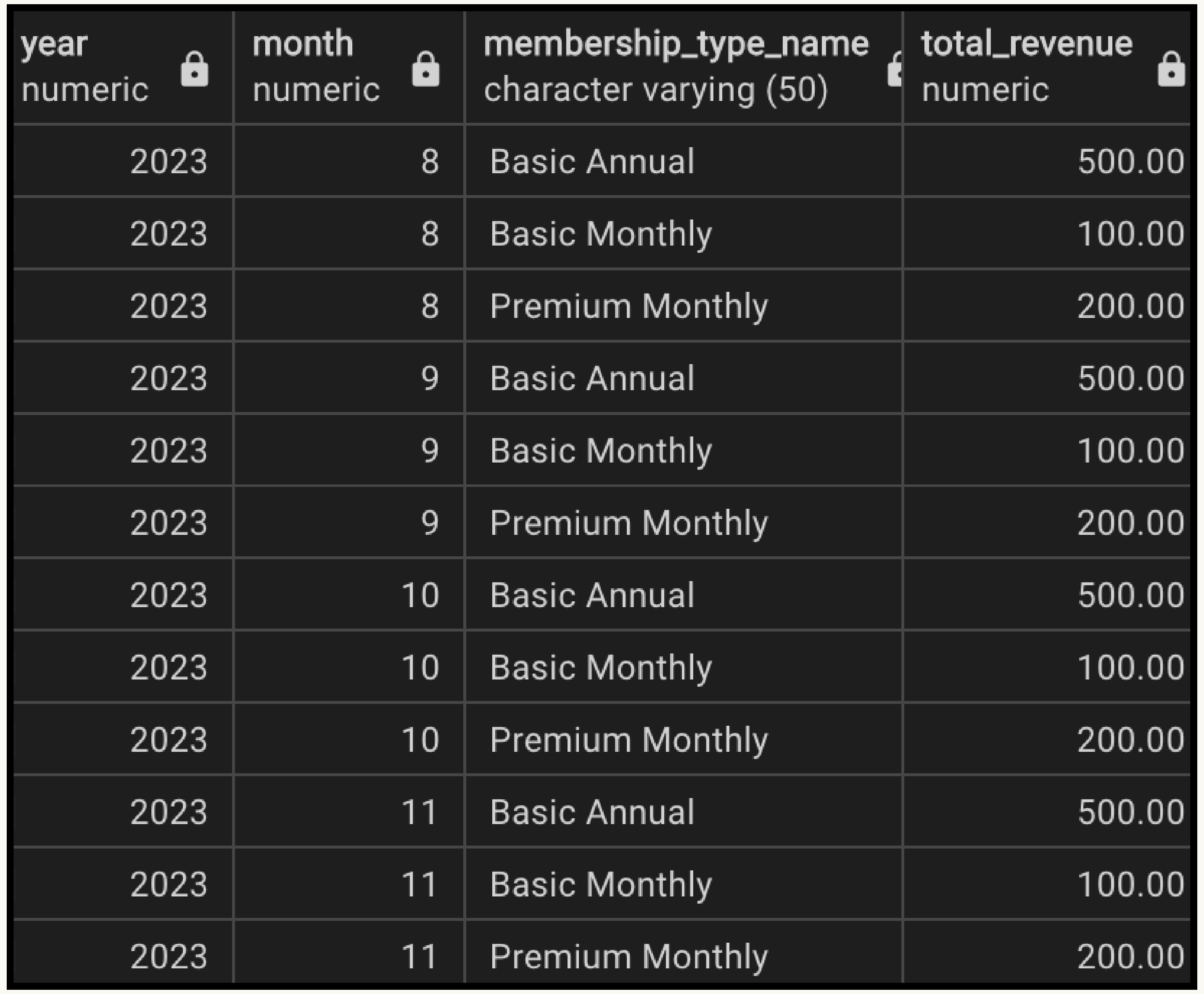


| year numeric | month numeric | revenue numeric |
|---|---|---|
| 2022 | 11 | 200.00 |
| 2022 | 12 | 150.00 |
| 2022 | [null] | 350.00 |
| 2023 | 1 | 110.00 |
| 2023 | 2 | 40.00 |
| 2023 | 3 | 110.00 |
| 2023 | 4 | 90.00 |
| 2023 | 5 | 120.00 |
| 2023 | 6 | 115.00 |
| 2023 | 7 | 150.00 |
| 2023 | 8 | 125.00 |
| 2023 | 9 | 175.00 |
| 2023 | 10 | 335.00 |
| 2023 | [null] | 1370.00 |

# EMILY HAS ASKED YOU TO GET THE TOTAL REVENUE FROM MEMBERSHIPS FOR EACH COMBINATION OF YEAR, MONTH, AND MEMBERSHIP TYPE.

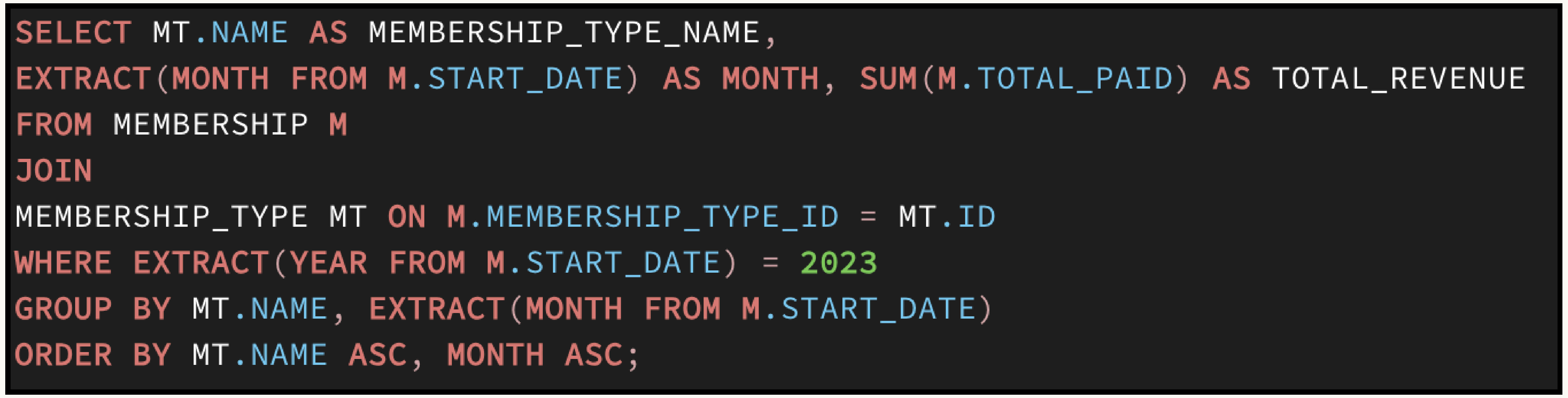


```sql
SELECT EXTRACT(YEAR FROM m.start_date) AS year,
EXTRACT(MONTH FROM m.start_date) AS month,
mt.name AS membership_type_name,
SUM(m.total_paid) AS total_revenue
FROM membership m
JOIN
membership_type mt ON m.membership_type_id = mt.id
GROUP BY
EXTRACT(YEAR FROM m.start_date),
EXTRACT(MONTH FROM m.start_date),
mt.name
ORDER BY year, month, membership_type_name;
```

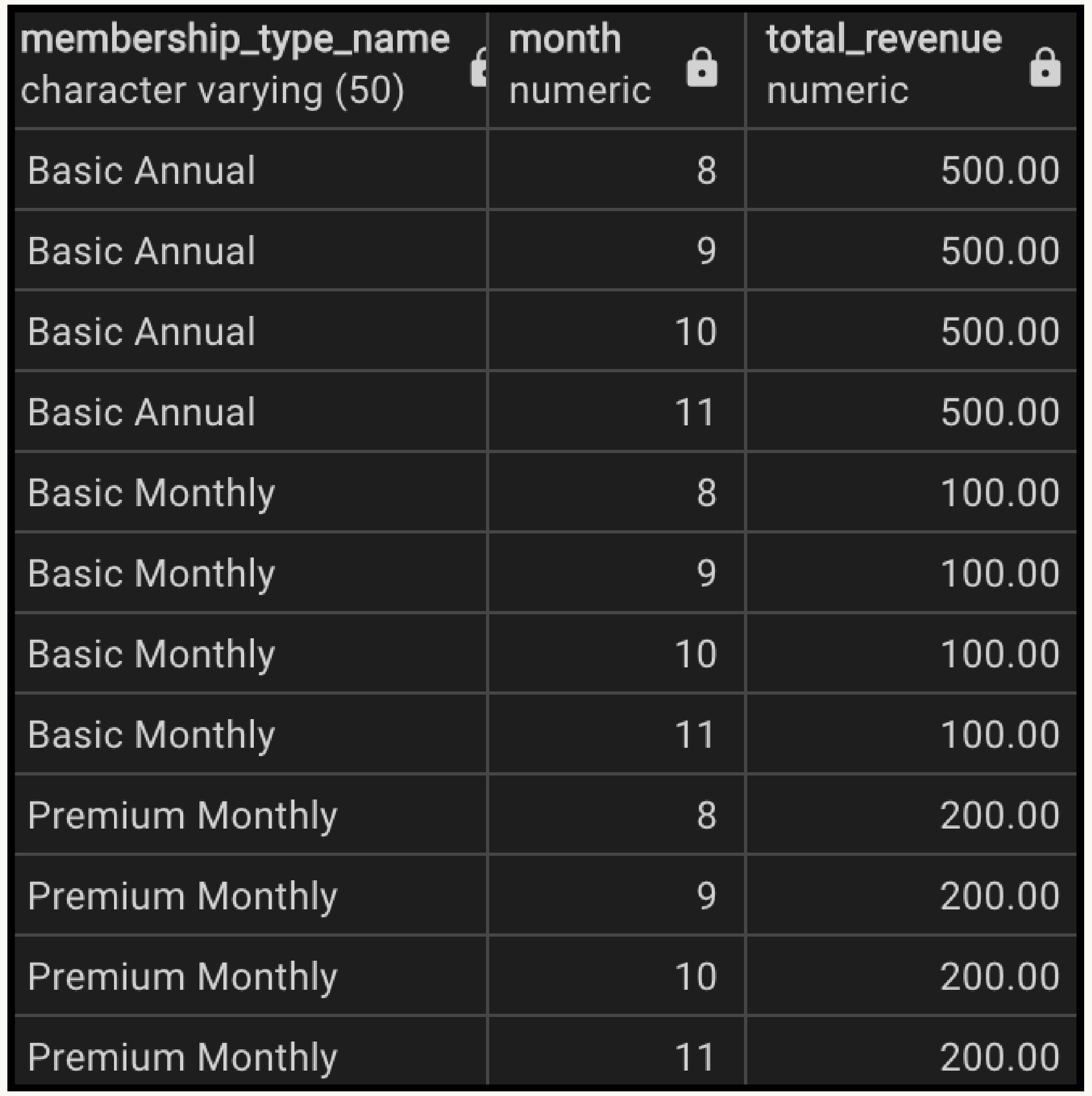


| year numeric | month numeric | membership_type_name character varying (50) | total_revenue numeric |
|---|---|---|---|
| 2023 | 8 | Basic Annual | 500.00 |
| 2023 | 8 | Basic Monthly | 100.00 |
| 2023 | 8 | Premium Monthly | 200.00 |
| 2023 | 9 | Basic Annual | 500.00 |
| 2023 | 9 | Basic Monthly | 100.00 |
| 2023 | 9 | Premium Monthly | 200.00 |
| 2023 | 10 | Basic Annual | 500.00 |
| 2023 | 10 | Basic Monthly | 100.00 |
| 2023 | 10 | Premium Monthly | 200.00 |
| 2023 | 11 | Basic Annual | 500.00 |
| 2023 | 11 | Basic Monthly | 100.00 |
| 2023 | 11 | Premium Monthly | 200.00 |

## EMILY WOULD LIKE DATA ABOUT MEMBERSHIPS PURCHASED IN 2023, WITH SUBTOTALS AND GRAND TOTALS FOR ALL THE DIFFERENT COMBINATIONS OF MEMBERSHIP TYPES AND MONTHS.

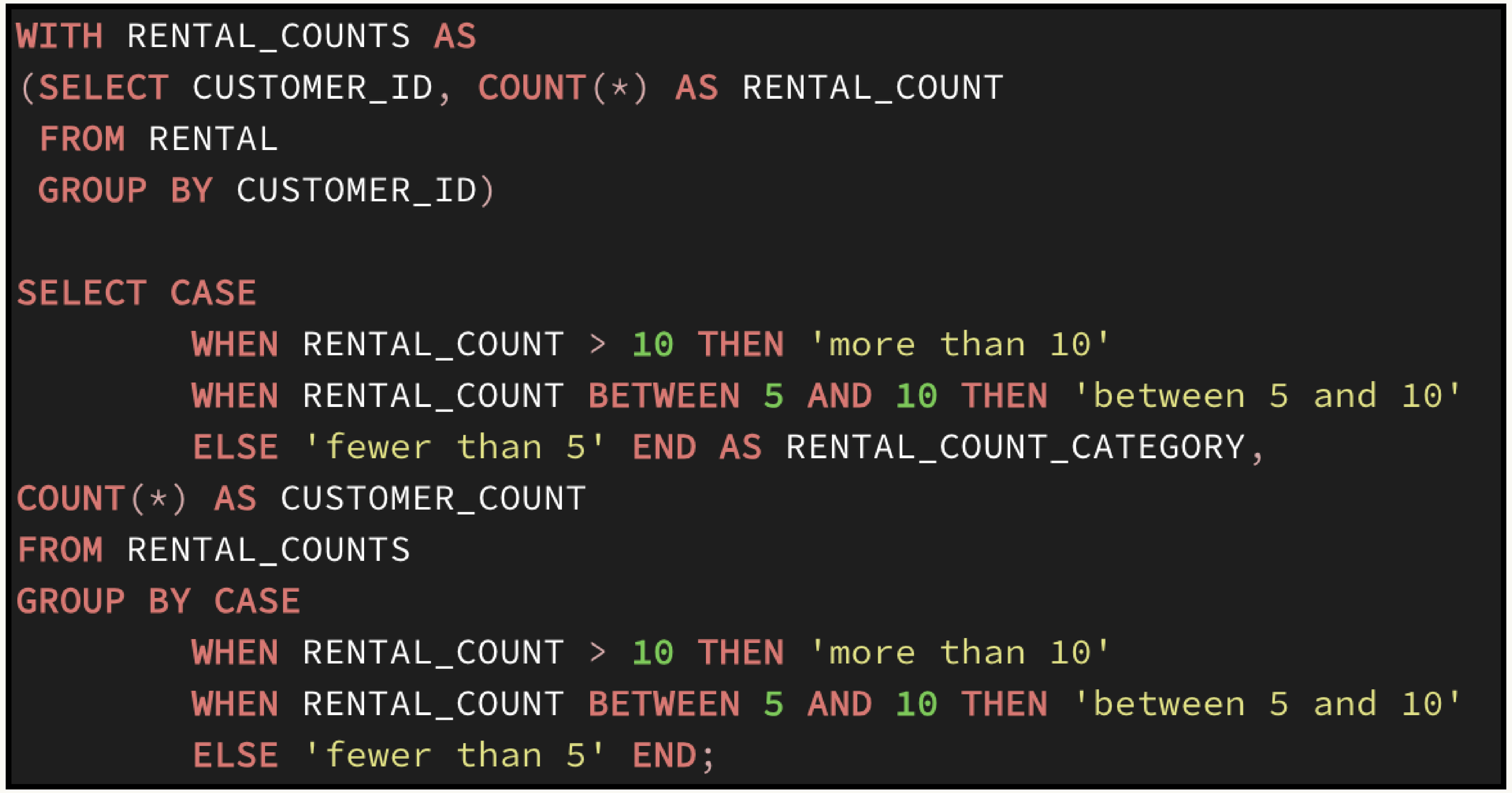


```sql
SELECT MT.NAME AS MEMBERSHIP_TYPE_NAME,
EXTRACT(MONTH FROM M.START_DATE) AS MONTH, SUM(M.TOTAL_PAID) AS TOTAL_REVENUE
FROM MEMBERSHIP M
JOIN
MEMBERSHIP_TYPE MT ON M.MEMBERSHIP_TYPE_ID = MT.ID
WHERE EXTRACT(YEAR FROM M.START_DATE) = 2023
GROUP BY MT.NAME, EXTRACT(MONTH FROM M.START_DATE)
ORDER BY MT.NAME ASC, MONTH ASC;
```



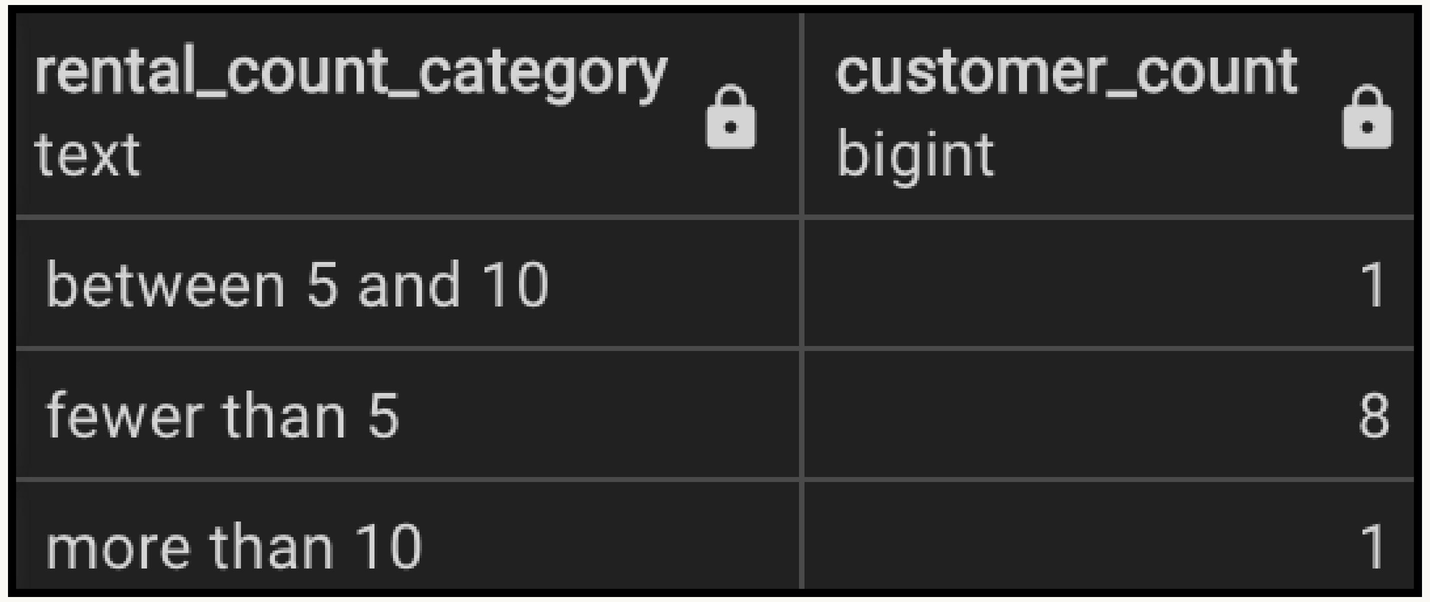
| membership_type_name character varying (50) | month numeric | total_revenue numeric |
|---|---|---|
| Basic Annual | 8 | 500.00 |
| Basic Annual | 9 | 500.00 |
| Basic Annual | 10 | 500.00 |
| Basic Annual | 11 | 500.00 |
| Basic Monthly | 8 | 100.00 |
| Basic Monthly | 9 | 100.00 |
| Basic Monthly | 10 | 100.00 |
| Basic Monthly | 11 | 100.00 |
| Premium Monthly | 8 | 200.00 |
| Premium Monthly | 9 | 200.00 |
| Premium Monthly | 10 | 200.00 |
| Premium Monthly | 11 | 200.00 |

## EMILY WANTS TO SEGMENT CUSTOMERS BASED ON THE NUMBER OF RENTALS AND SEE THE COUNT OF CUSTOMERS IN EACH SEGMENT.



```sql
WITH RENTAL_COUNTS AS
(SELECT CUSTOMER_ID, COUNT(*) AS RENTAL_COUNT
 FROM RENTAL
 GROUP BY CUSTOMER_ID)

SELECT CASE
        WHEN RENTAL_COUNT > 10 THEN 'more than 10'
        WHEN RENTAL_COUNT BETWEEN 5 AND 10 THEN 'between 5 and 10'
        ELSE 'fewer than 5' END AS RENTAL_COUNT_CATEGORY,
COUNT(*) AS CUSTOMER_COUNT
FROM RENTAL_COUNTS
GROUP BY CASE
        WHEN RENTAL_COUNT > 10 THEN 'more than 10'
        WHEN RENTAL_COUNT BETWEEN 5 AND 10 THEN 'between 5 and 10'
        ELSE 'fewer than 5' END;
```



| rental_count_category<br>text | customer_count<br>bigint |
|---|---|
| between 5 and 10 | 1 |
| fewer than 5 | 8 |
| more than 10 | 1 |

THANK YOU!