Appendix: Case of Study 1 (Cyclistic)

SQL queries

Prepare

Organization

Headers description

```
SELECT *
FROM rides
LIMIT 50;
```

Data credibility

Graph 1: Rides duration distribution

```
SELECT (ended_at - started_at) AS duration, COUNT(1) AS count
FROM rides
GROUP BY duration
ORDER BY duration;
```

Graph 2: Rides duration distribution between 1 and 60 minutes

```
SELECT (ended_at - started_at) AS duration, COUNT(1) AS count
FROM rides
WHERE  (EXTRACT(epoch  FROM  ended_at  -  started_at)  /  60.0)  >=  1.0  AND  (EXTRACT(epoch  FROM  ended_at  -
started_at) / 60.0) <= 60.0
GROUP BY duration
ORDER BY duration;
```

Table 1: Percentage of rides between 1 and 60 minutes

```
WITH
range_rides AS (SELECT COUNT(1) AS count
                FROM rides
                 WHERE  (EXTRACT(epoch FROM ended_at - started_at) / 60.0) >= 1.0 AND (EXTRACT(epoch FROM
ended_at - started_at) / 60.0) <= (60.0)),
total_rides AS (SELECT COUNT(*) AS total
                FROM rides)
SELECT range_rides.count,
       total_rides.total,
       ROUND((range_rides.count/total_rides.total::numeric) * 100, 2) AS percentage
FROM range_rides, total_rides;
```

Data integrity

Table 2: Total amount of missing cells

```
SELECT SUM(null_cells.count) AS missing_cells
FROM table_columns_missing AS null_cells;
```

Table 3: Table with missing cells per variable
VIEW Table:

```
CREATE OR REPLACE VIEW table_columns_missing AS
        SELECT 'ride_id' AS variable, COUNT(*)
        FROM rides
        WHERE ride_id IS NULL
UNION
        SELECT 'rideable_type' AS variable, COUNT(*)
        FROM rides
        WHERE rideable_type IS NULL
UNION
        SELECT 'started_at' AS variable, COUNT(*)
        FROM rides
        WHERE started_at IS NULL
UNION
        SELECT 'ended_at' AS variable, COUNT(*)
        FROM rides
        WHERE ended_at IS NULL
UNION
        SELECT 'start_station_name' AS variable, COUNT(*)
        FROM rides
        WHERE start_station_name IS NULL
UNION
        SELECT 'start_station_id' AS variable, COUNT(*)
        FROM rides
        WHERE start_station_id IS NULL
UNION
        SELECT 'end_station_name' AS variable, COUNT(*)
        FROM rides
        WHERE end_station_name IS NULL
UNION
        SELECT 'end_station_id' AS variable, COUNT(*)
        FROM rides
        WHERE end_station_id IS NULL
UNION
        SELECT 'start_lat' AS variable, COUNT(*)
        FROM rides
        WHERE start_lat IS NULL
UNION
        SELECT 'start_lng' AS variable, COUNT(*)
        FROM rides
        WHERE start_lng IS NULL
UNION
        SELECT 'end_lat' AS variable, COUNT(*)
        FROM rides
        WHERE end_lat IS NULL
UNION
        SELECT 'end_lng' AS variable, COUNT(*)
        FROM rides
        WHERE end_lng IS NULL
UNION
        SELECT 'member_casual' AS variable, COUNT(*)
        FROM rides
        WHERE member_casual IS NULL;
```

WITH Statement:

```
WITH
var_w_null AS (SELECT * FROM table_columns_missing),
total_records AS (SELECT COUNT(*) AS total FROM rides)
SELECT var_w_null.variable,
       Var_w_null.count,
       ROUND((var_w_null.count/total_records.total::numeric * 100), 2) AS percentage
FROM var_w_null, total_records
ORDER BY percentage DESC;
```

Table 4: Percentage of records with starting or ending station NULL

```
WITH
table1 AS (SELECT COUNT(*) AS cant_rows
            FROM rides
            WHERE start_station_name IS NULL OR end_station_name IS NULL),
table2 AS (SELECT COUNT(*) AS total
            FROM rides)
SELECT 'No start or end station' AS null_variable,
       table1.cant_rows,
       table2.total,
       ROUND((table1.cant_rows / table2.total::numeric) * 100, 2) AS percentage
FROM table1, table2;
```

Analyze

Table 5 and Graph 3: Distribution total rides (Annually)

```
SELECT DATE_PART('year', started_at) AS year,
       COUNT(CASE WHEN member_casual = 'member' THEN 1 END) AS member_count,
      ROUND(100.0  *  COUNT(CASE  WHEN  member_casual  =  'member'  THEN  1  END)  /  COUNT(*),  2)  AS
member_percentage,
       COUNT(CASE WHEN member_casual = 'casual' THEN 1 END) AS casual_count,
      ROUND(100.0  *  COUNT(CASE  WHEN  member_casual  =  'casual'  THEN  1  END)  /  COUNT(*),  2)  AS
casual_percentage,
       (COUNT(CASE WHEN member_casual = 'member' THEN 1 END) + COUNT(CASE WHEN member_casual = 'casual'
THEN 1 END)) AS total
FROM rides
GROUP BY year;
```

Table 6, Graph 4 and Graph 5: Distribution total rides (Seasonal)

```
SELECT season,
       COUNT(*) as total_rides,
       COUNT(CASE WHEN member_casual = 'member' THEN 1 END) AS member_count,
      ROUND(100.0  *  COUNT(CASE  WHEN  member_casual  =  'member'  THEN  1  END)  /  COUNT(*),  2)  AS
member_percentage,
       COUNT(CASE WHEN member_casual = 'casual' THEN 1 END) AS casual_count,
      ROUND(100.0  *  COUNT(CASE  WHEN  member_casual  =  'casual'  THEN  1  END)  /  COUNT(*),  2)  AS
casual_percentage
FROM rides
GROUP BY season
ORDER BY total_rides DESC;
```

Graph 6: Distribution total rides (Monthly)

```
SELECT to_char(started_at, 'Month') AS month,
       COUNT(*) AS count,
       (COUNT(*) * 100.0 / (SELECT COUNT(*) FROM rides))::numeric(5,2) AS percentage
FROM rides
GROUP BY EXTRACT(MONTH FROM started_at), month
ORDER BY EXTRACT(MONTH FROM started_at);
```

Graph 7: Distribution total rides by user type (Monthly)

```
SELECT to_char(started_at, 'Month') AS month,
       COUNT(*) as total_rides,
       COUNT(CASE WHEN member_casual = 'member' THEN 1 END) AS member_count,
      ROUND(100.0  *  COUNT(CASE  WHEN  member_casual  =  'member'  THEN  1  END)  /  COUNT(*),  2)  AS
member_percentage,
       COUNT(CASE WHEN member_casual = 'casual' THEN 1 END) AS casual_count,
      ROUND(100.0  *  COUNT(CASE  WHEN  member_casual  =  'casual'  THEN  1  END)  /  COUNT(*),  2)  AS
casual_percentage
FROM rides
GROUP BY EXTRACT(MONTH FROM started_at), month
ORDER BY EXTRACT(MONTH FROM started_at);
```

Graph 8: Mode Day of the Week Annually

```
SELECT to_char(started_at, 'Day') AS day_of_the_week,
       COUNT(*)
FROM rides
GROUP BY EXTRACT(DOW FROM started_at), day_of_the_week
ORDER BY EXTRACT(DOW FROM started_at);
```

Graph 9: Mode Day of the Week Seasonal

```
SELECT
    to_char(started_at, 'Day') AS DOW,
    COUNT(CASE WHEN season = 'Winter' THEN 1 END) AS winter_count,
    COUNT(CASE WHEN season = 'Spring' THEN 1 END) AS spring_count,
    COUNT(CASE WHEN season = 'Summer' THEN 1 END) AS summer_count,
    COUNT(CASE WHEN season = 'Fall' THEN 1 END) AS fall_count
FROM rides
GROUP BY EXTRACT(DOW FROM started_at), DOW
ORDER BY EXTRACT(DOW FROM started_at);
```

Table 7: Mode Day of the Week by Season

```
-- Winter
SELECT
    to_char(started_at, 'Day') AS DOW,
    COUNT(*) AS winter_count
FROM rides
WHERE season = 'Winter'
GROUP BY EXTRACT(DOW FROM started_at), DOW
ORDER BY COUNT(*) DESC;

-- Spring
SELECT
    to_char(started_at, 'Day') AS DOW,
    COUNT(*) AS spring_count
```

```
FROM rides
WHERE season = 'Spring'
GROUP BY EXTRACT(DOW FROM started_at), DOW
ORDER BY COUNT(*) DESC;

-- Summer
SELECT
    to_char(started_at, 'Day') AS DOW,
    COUNT(*) AS summer_count
FROM rides
WHERE season = 'Summer'
GROUP BY EXTRACT(DOW FROM started_at), DOW
ORDER BY COUNT(*) DESC;

-- Fall
SELECT
    to_char(started_at, 'Day') AS DOW,
    COUNT(*) AS fall_count
FROM rides
WHERE season = 'Fall'
GROUP BY EXTRACT(DOW FROM started_at), DOW
ORDER BY COUNT(*) DESC;
```

Graph 10: Distribution by users through the days of the week. (Annually)

```
SELECT to_char(started_at, 'Day') AS DOW,
       COUNT(CASE WHEN member_casual = 'member' THEN 1 END) AS member_rides,
       COUNT(CASE WHEN member_casual = 'casual' THEN 1 END) AS casual_rides
FROM rides
GROUP BY EXTRACT(DOW FROM started_at), DOW
ORDER BY EXTRACT(DOW FROM started_at);
```

Graphs 11-14: Distribution by users through the days of the week. (By season)

```
-- Winter
SELECT
    to_char(started_at, 'Day') AS DOW,
    COUNT(CASE WHEN season = 'Winter' AND member_casual = 'member' THEN 1 END) AS member_winter_count,
    COUNT(CASE WHEN season = 'Winter' AND member_casual = 'casual' THEN 1 END) AS casual_winter_count
FROM rides
GROUP BY EXTRACT(DOW FROM started_at), DOW
ORDER BY EXTRACT(DOW FROM started_at);

-- Spring
SELECT
    to_char(started_at, 'Day') AS DOW,
    COUNT(CASE WHEN season = 'Spring' AND member_casual = 'member' THEN 1 END) AS member_spring_count,
    COUNT(CASE WHEN season = 'Spring' AND member_casual = 'casual' THEN 1 END) AS casual_spring_count
FROM rides
GROUP BY EXTRACT(DOW FROM started_at), DOW
ORDER BY EXTRACT(DOW FROM started_at);

-- Summer
SELECT
    to_char(started_at, 'Day') AS DOW,
    COUNT(CASE WHEN season = 'Summer' AND member_casual = 'member' THEN 1 END) AS member_summer_count,
    COUNT(CASE WHEN season = 'Summer' AND member_casual = 'casual' THEN 1 END) AS casual_summer_count
FROM rides
```

```
GROUP BY EXTRACT(DOW FROM started_at), DOW
ORDER BY EXTRACT(DOW FROM started_at);

-- Fall
SELECT
    to_char(started_at, 'Day') AS DOW,
    COUNT(CASE WHEN season = 'Fall' AND member_casual = 'member' THEN 1 END) AS member_fall_count,
    COUNT(CASE WHEN season = 'Fall' AND member_casual = 'casual' THEN 1 END) AS casual_fall_count
FROM rides
GROUP BY EXTRACT(DOW FROM started_at), DOW
ORDER BY EXTRACT(DOW FROM started_at);
```

Graph 15: Most common ride length by users - Annually

```
SELECT duration_ride,
       COUNT(CASE WHEN member_casual = 'member' THEN 1 END) AS member_count,
       COUNT(CASE WHEN member_casual = 'casual' THEN 1 END) AS casual_count
FROM rides
GROUP BY duration_ride
ORDER BY duration_ride;
```

Graphs 16-19: Most common ride length by users - Seasonal

```
-- Winter by users
SELECT duration_ride,
      COUNT(CASE WHEN season = 'Winter' AND member_casual = 'member' THEN 1 END) AS member_winter_count,
      COUNT(CASE WHEN season = 'Winter' AND member_casual = 'casual' THEN 1 END) AS casual_winter_count
FROM rides
GROUP BY duration_ride
ORDER BY duration_ride;

-- Spring by users
SELECT duration_ride,
      COUNT(CASE WHEN season = 'Spring' AND member_casual = 'member' THEN 1 END) AS member_spring_count,
      COUNT(CASE WHEN season = 'Spring' AND member_casual = 'casual' THEN 1 END) AS casual_spring_count
FROM rides
GROUP BY duration_ride
ORDER BY duration_ride;

-- Summer by users
SELECT duration_ride,
      COUNT(CASE WHEN season = 'Summer' AND member_casual = 'member' THEN 1 END) AS member_summer_count,
      COUNT(CASE WHEN season = 'Summer' AND member_casual = 'casual' THEN 1 END) AS casual_summer_count
FROM rides
GROUP BY duration_ride
ORDER BY duration_ride;

-- Fall by users
SELECT duration_ride,
      COUNT(CASE WHEN season = 'Fall' AND member_casual = 'member' THEN 1 END) AS member_fall_count,
      COUNT(CASE WHEN season = 'Fall' AND member_casual = 'casual' THEN 1 END) AS casual_fall_count
FROM rides
GROUP BY duration_ride
ORDER BY duration_ride;
```

Graph 20: Most common ride length by users - Monthly

```sql
SELECT to_char(started_at, 'Month') as month,
       mode() WITHIN GROUP (ORDER BY CASE WHEN member_casual = 'member' THEN duration_ride END DESC) AS
most_common_duration_member,
       mode() WITHIN GROUP (ORDER BY CASE WHEN member_casual = 'casual' THEN duration_ride END DESC) AS
most_common_duration_casual
FROM rides
GROUP BY EXTRACT(MONTH FROM started_at), month
ORDER BY EXTRACT(MONTH FROM started_at);
```

Graphs 21-24: Most common ride length by users - Day of the Week

```sql
-- Winter
SELECT to_char(started_at, 'Day') as DOW,
       mode() WITHIN GROUP (ORDER BY CASE WHEN member_casual = 'member' THEN duration_ride END DESC) AS
winter_most_common_duration_member,
       mode() WITHIN GROUP (ORDER BY CASE WHEN member_casual = 'casual' THEN duration_ride END DESC) AS
winter_most_common_duration_casual
FROM rides
WHERE season = 'Winter'
GROUP BY EXTRACT(DOW FROM started_at), DOW
ORDER BY EXTRACT(DOW FROM started_at);

-- Spring
SELECT to_char(started_at, 'Day') as DOW,
       mode() WITHIN GROUP (ORDER BY CASE WHEN member_casual = 'member' THEN duration_ride END DESC) AS
spring_most_common_duration_member,
       mode() WITHIN GROUP (ORDER BY CASE WHEN member_casual = 'casual' THEN duration_ride END DESC) AS
spring_most_common_duration_casual
FROM rides
WHERE season = 'Spring'
GROUP BY EXTRACT(DOW FROM started_at), DOW
ORDER BY EXTRACT(DOW FROM started_at);

-- Summer
SELECT to_char(started_at, 'Day') as DOW,
       mode() WITHIN GROUP (ORDER BY CASE WHEN member_casual = 'member' THEN duration_ride END DESC) AS
summer_most_common_duration_member,
       mode() WITHIN GROUP (ORDER BY CASE WHEN member_casual = 'casual' THEN duration_ride END DESC) AS
summer_most_common_duration_casual
FROM rides
WHERE season = 'Summer'
GROUP BY EXTRACT(DOW FROM started_at), DOW
ORDER BY EXTRACT(DOW FROM started_at);

-- Fall
SELECT to_char(started_at, 'Day') as DOW,
       mode() WITHIN GROUP (ORDER BY CASE WHEN member_casual = 'member' THEN duration_ride END DESC) AS
fall_most_common_duration_member,
       mode() WITHIN GROUP (ORDER BY CASE WHEN member_casual = 'casual' THEN duration_ride END DESC) AS
fall_most_common_duration_casual
FROM rides
WHERE season = 'Fall'
GROUP BY EXTRACT(DOW FROM started_at), DOW
ORDER BY EXTRACT(DOW FROM started_at);
```

Table 8: Average ride duration by users. Annually.

```
SELECT member_casual,
       ROUND(AVG(duration_ride), 1) AS average
FROM rides
GROUP BY member_casual
ORDER BY average DESC;
```

Graph 25: Average ride duration by users. Seasonal.

```
SELECT season,
      ROUND(AVG(CASE  WHEN  member_casual  =  'member'  THEN  duration_ride  ELSE  NULL  END),  1)  AS
avg_member_duration,
      ROUND(AVG(CASE  WHEN  member_casual  =  'casual'  THEN  duration_ride  ELSE  NULL  END),  1)  AS
avg_casual_duration
FROM rides
GROUP BY season
ORDER BY avg_casual_duration DESC;
```

Graph 26: Average ride duration by users. Monthly.

```
SELECT to_char(started_at, 'Month') AS month,
      ROUND(AVG(CASE  WHEN  member_casual  =  'member'  THEN  duration_ride  ELSE  NULL  END),  1)  AS
avg_member_duration,
      ROUND(AVG(CASE  WHEN  member_casual  =  'casual'  THEN  duration_ride  ELSE  NULL  END),  1)  AS
avg_casual_duration
FROM rides
GROUP BY EXTRACT(MONTH FROM started_at), month
ORDER BY EXTRACT(MONTH FROM started_at);
```

Graphs 27-30: Average ride duration by users. Seasonal by Day of the Week.

```
-- Winter
SELECT to_char(started_at, 'Day') AS DOW,
      ROUND(AVG(CASE WHEN season = 'Winter' AND member_casual = 'member' THEN duration_ride ELSE NULL
END), 1) AS member_avg_winter,
       ROUND(AVG(CASE WHEN season = 'Winter' AND member_casual = 'casual' THEN duration_ride ELSE NULL
END), 1) AS casual_avg_winter
FROM rides
GROUP BY EXTRACT(DOW FROM started_at), DOW
ORDER BY EXTRACT(DOW FROM started_at);

-- Spring
SELECT to_char(started_at, 'Day') AS DOW,
       ROUND(AVG(CASE WHEN season = 'Spring' AND member_casual = 'member' THEN duration_ride ELSE NULL
END), 1) AS member_avg_spring,
       ROUND(AVG(CASE WHEN season = 'Spring' AND member_casual = 'casual' THEN duration_ride ELSE NULL
END), 1) AS casual_avg_spring
FROM rides
GROUP BY EXTRACT(DOW FROM started_at), DOW
ORDER BY EXTRACT(DOW FROM started_at);

-- Summer
SELECT to_char(started_at, 'Day') AS DOW,
       ROUND(AVG(CASE WHEN season = 'Summer' AND member_casual = 'member' THEN duration_ride ELSE NULL
END), 1) AS member_avg_summer,
       ROUND(AVG(CASE WHEN season = 'Summer' AND member_casual = 'casual' THEN duration_ride ELSE NULL
END), 1) AS casual_avg_summer
```

```
FROM rides
GROUP BY EXTRACT(DOW FROM started_at), DOW
ORDER BY EXTRACT(DOW FROM started_at);

-- Fall
SELECT to_char(started_at, 'Day') AS DOW,
     ROUND(AVG(CASE WHEN season = 'Fall' AND member_casual = 'member' THEN duration_ride ELSE NULL END),
1) AS member_avg_fall,
     ROUND(AVG(CASE WHEN season = 'Fall' AND member_casual = 'casual' THEN duration_ride ELSE NULL END),
1) AS casual_avg_fall
FROM rides
GROUP BY EXTRACT(DOW FROM started_at), DOW
ORDER BY EXTRACT(DOW FROM started_at);
```

Graph 31: Bike types used by users. Annually

```
SELECT EXTRACT(YEAR FROM started_at) AS year,
       COUNT(CASE  WHEN  member_casual  =  'member'  AND  rideable_type  =  'classic_bike'  THEN  1  END)  AS
member_classic_rides,
       COUNT(CASE  WHEN  member_casual  =  'casual'  AND  rideable_type  =  'classic_bike'  THEN  1  END)  AS
casual_classic_rides,
       COUNT(CASE  WHEN  member_casual  =  'member'  AND  rideable_type  =  'electric_bike'  THEN  1  END)  AS
member_electric_rides,
       COUNT(CASE  WHEN  member_casual  =  'casual'  AND  rideable_type  =  'electric_bike'  THEN  1  END)  AS
casual_electric_rides,
       COUNT(CASE  WHEN  member_casual  =  'member'  AND  rideable_type  =  'docked_bike'  THEN  1  END)  AS
member_docked_rides,
       COUNT(CASE  WHEN  member_casual  =  'casual'  AND  rideable_type  =  'docked_bike'  THEN  1  END)  AS
casual_docked_rides
FROM rides
GROUP BY year;
```

Graph 32: Bike types used by users. Seasonal

```
SELECT season,
       COUNT(CASE  WHEN  member_casual  =  'member'  AND  rideable_type  =  'classic_bike'  THEN  1  END)  AS
member_classic_rides,
       COUNT(CASE  WHEN  member_casual  =  'casual'  AND  rideable_type  =  'classic_bike'  THEN  1  END)  AS
casual_classic_rides,
       COUNT(CASE  WHEN  member_casual  =  'member'  AND  rideable_type  =  'electric_bike'  THEN  1  END)  AS
member_electric_rides,
       COUNT(CASE  WHEN  member_casual  =  'casual'  AND  rideable_type  =  'electric_bike'  THEN  1  END)  AS
casual_electric_rides,
       COUNT(CASE  WHEN  member_casual  =  'member'  AND  rideable_type  =  'docked_bike'  THEN  1  END)  AS
member_docked_rides,
       COUNT(CASE  WHEN  member_casual  =  'casual'  AND  rideable_type  =  'docked_bike'  THEN  1  END)  AS
casual_docked_rides
FROM rides
GROUP BY season;
```

Graph 33: Bike types used by users. Monthly

```
SELECT to_char(started_at, 'Month') AS month_name,
       COUNT(CASE  WHEN  member_casual  =  'member'  AND  rideable_type  =  'classic_bike'  THEN  1  END)  AS
member_classic_rides,
       COUNT(CASE  WHEN  member_casual  =  'casual'  AND  rideable_type  =  'classic_bike'  THEN  1  END)  AS
casual_classic_rides,
      COUNT(CASE  WHEN  member_casual  =  'member'  AND  rideable_type  =  'electric_bike'  THEN  1  END)  AS
member_electric_rides,
      COUNT(CASE  WHEN  member_casual  =  'casual'  AND  rideable_type  =  'electric_bike'  THEN  1  END)  AS
casual_electric_rides,
      COUNT(CASE  WHEN  member_casual  =  'member'  AND  rideable_type  =  'docked_bike'  THEN  1  END)  AS
member_docked_rides,
      COUNT(CASE  WHEN  member_casual  =  'casual'  AND  rideable_type  =  'docked_bike'  THEN  1  END)  AS
casual_docked_rides
FROM rides
GROUP BY EXTRACT(MONTH FROM started_at), month_name
ORDER BY EXTRACT(MONTH FROM started_at);
```

Graph 34: Total time by users. Annually

```
SELECT  ROUND(SUM(CASE  WHEN  member_casual  =  'member'  THEN  duration_ride  END)/60::numeric,  1)  AS
member_total_hours_used,
        ROUND(SUM(CASE  WHEN  member_casual  =  'casual'  THEN  duration_ride  END)/60::numeric,  1)  AS
casual_total_hours_used
FROM rides;
```

Graph 35: Total time by users. Seasonal

```
SELECT CASE WHEN season = 'Winter' THEN 1
            WHEN season = 'Spring' THEN 2
            WHEN season = 'Summer' THEN 3
            WHEN season = 'Fall' THEN 4
            END AS season_num,
           season,
           ROUND(SUM(CASE  WHEN  member_casual  =  'member'  THEN  duration_ride  END)/60::numeric,  1)  AS
member_total_hours_used,
           ROUND(SUM(CASE  WHEN  member_casual  =  'casual'  THEN  duration_ride  END)/60::numeric,  1)  AS
casual_total_hours_used
FROM rides
GROUP BY season_num, season;
```

Graph 36: Total time by users. Monthly

```
SELECT to_char(started_at, 'Month') AS month_name,
      ROUND(SUM(CASE  WHEN  member_casual  =  'member'  THEN  duration_ride  END)/60::numeric,  1)  AS
member_total_hours_used,
      ROUND(SUM(CASE  WHEN  member_casual  =  'casual'  THEN  duration_ride  END)/60::numeric,  1)  AS
casual_total_hours_used
FROM rides
GROUP BY EXTRACT(MONTH FROM started_at), month_name
ORDER BY EXTRACT(MONTH FROM started_at);
```

Graphs 37-40: Total time by users. By Day of the Week and Season.

```
-- Winter
SELECT to_char(started_at, 'Day') AS DOW,
       ROUND(SUM(CASE  WHEN  season  =  'Winter'  AND  member_casual  =  'member'  THEN  duration_ride
END)/60::numeric, 1) AS member_winter_total_hours_used,
       ROUND(SUM(CASE  WHEN  season  =  'Winter'  AND  member_casual  =  'casual'  THEN  duration_ride
END)/60::numeric, 1) AS casual_winter_total_hours_used
FROM rides
GROUP BY EXTRACT(DOW FROM started_at), DOW
ORDER BY EXTRACT(DOW FROM started_at);
```

```sql
-- Spring
SELECT to_char(started_at, 'Day') AS DOW,
       ROUND(SUM(CASE  WHEN  season  =  'Spring'  AND  member_casual  =  'member'  THEN  duration_ride
END)/60::numeric, 1) AS member_spring_total_hours_used,
       ROUND(SUM(CASE  WHEN  season  =  'Spring'  AND  member_casual  =  'casual'  THEN  duration_ride
END)/60::numeric, 1) AS casual_spring_total_hours_used
FROM rides
GROUP BY EXTRACT(DOW FROM started_at), DOW
ORDER BY EXTRACT(DOW FROM started_at);

-- Summer
SELECT to_char(started_at, 'Day') AS DOW,
       ROUND(SUM(CASE  WHEN  season  =  'Summer'  AND  member_casual  =  'member'  THEN  duration_ride
END)/60::numeric, 1) AS member_summer_total_hours_used,
       ROUND(SUM(CASE  WHEN  season  =  'Summer'  AND  member_casual  =  'casual'  THEN  duration_ride
END)/60::numeric, 1) AS casual_summer_total_hours_used
FROM rides
GROUP BY EXTRACT(DOW FROM started_at), DOW
ORDER BY EXTRACT(DOW FROM started_at);

-- Fall
SELECT to_char(started_at, 'Day') AS DOW,
         ROUND(SUM(CASE  WHEN  season  =  'Fall'  AND  member_casual  =  'member'  THEN  duration_ride
END)/60::numeric, 1) AS member_fall_total_hours_used,
         ROUND(SUM(CASE  WHEN  season  =  'Fall'  AND  member_casual  =  'casual'  THEN  duration_ride
END)/60::numeric, 1) AS casual_fall_total_hours_used
FROM rides
GROUP BY EXTRACT(DOW FROM started_at), DOW
ORDER BY EXTRACT(DOW FROM started_at);
```