```
--number of rides per user_type
SELECT user_type, COUNT(*) as count_user_type
FROM baywheels_2019
GROUP BY 1;

--create trip duration column + category
SELECT bike_id, user_type, end_time - start_time AS trip_duration,
CASE
WHEN (end_time - start_time) > '00:00:00' AND (end_time - start_time) < '00:30:00' THEN
'Short'
WHEN (end_time - start_time) BETWEEN '00:30:00' AND '01:00:00' THEN 'Medium'
WHEN (end_time - start_time) > '01:00:00' AND (end_time - start_time) < '02:00:00' THEN
'Long'
WHEN (end_time - start_time) >= '02:00:00' THEN 'Super long'
ELSE 'n/a'
END AS trip_duration_category
FROM baywheels_2019
ORDER BY CASE WHEN (end_time - start_time) > '00:00:00' AND (end_time - start_time) <
'00:30:00' THEN 1
        WHEN (end_time - start_time) BETWEEN '00:30:00' AND '01:00:00' THEN 2
        WHEN (end_time - start_time) > '01:00:00' AND (end_time - start_time) < '02:00:00'
THEN 3
        WHEN (end_time - start_time) >= '02:00:00' THEN 4
        ELSE 5 END, 3

--start&end stations with names and count > 2000
SELECT b9.start_station_id, bs.name, b9.end_station_id, bs2.name, COUNT(*) AS count_rides
FROM baywheels_2019 b9
JOIN baywheels_stations bs
        ON b9.start_station_id = bs.id
JOIN baywheels_stations bs2
        ON b9.end_station_id = bs2.id
GROUP BY 1,2,3,4
HAVING COUNT(*) > 2000
ORDER BY 5 DESC, 1;

--number of rides by month 2019/18
--SOLUTION: Perform maintenance and tuning during winter months.
SELECT
CASE
WHEN DATE_PART('month', start_time) = 1 THEN 'January'
WHEN DATE_PART('month', start_time) = 2 THEN 'February'
WHEN DATE_PART('month', start_time) = 3 THEN 'March'
WHEN DATE_PART('month', start_time) = 4 THEN 'April'
```

```
WHEN DATE_PART('month', start_time) = 5 THEN 'May'
WHEN DATE_PART('month', start_time) = 6 THEN 'June'
WHEN DATE_PART('month', start_time) = 7 THEN 'July'
WHEN DATE_PART('month', start_time) = 8 THEN 'August'
WHEN DATE_PART('month', start_time) = 9 THEN 'September'
WHEN DATE_PART('month', start_time) = 10 THEN 'October'
WHEN DATE_PART('month', start_time) = 11 THEN 'November'
WHEN DATE_PART('month', start_time) = 12 THEN 'December'
ELSE '' END AS month, COUNT(*) as number_of_rides
FROM
        (SELECT *
        FROM baywheels_2018
        UNION
        SELECT *
        FROM baywheels_2019) as full_years
GROUP BY 1
ORDER BY 2 DESC;


-- slowest month/day s
SELECT DATE_PART('month', start_time), DATE_PART('day', start_time),  COUNT(*)
FROM
        (SELECT *
        FROM baywheels_2018
        UNION
        SELECT *
        FROM baywheels_2019) as full_years
GROUP BY 1,2
ORDER BY 3;

--number of rides starting at this station
SELECT b9.start_station_id, COUNT(*) AS ride_count
FROM baywheels_2019 b9
JOIN baywheels_stations bs
        ON b9.start_station_id = bs.id
GROUP BY 1
ORDER BY 2 DESC
LIMIT 15;

--number of rides ending at this station
SELECT b9.end_station_id, COUNT(*) AS ride_count
FROM baywheels_2019 b9
JOIN baywheels_stations bs
        ON b9.start_station_id = bs.id
```

```
GROUP BY 1
ORDER BY 2 DESC
LIMIT 100;
--Q. IS THERE A WAY TO UNITE THESE TWO ABOVE QUERIES AND GET A DIFFERENCE OF
COUNTS BY STATION_ID


--time spent in activity for each bike_idSELECT bike_id, end_time - start_time as trip_duration
FROM
        (SELECT *
         FROM baywheels_2017
         UNION
         SELECT *
         FROM baywheels_2018
         UNION
         SELECT *
         FROM baywheels_2019) as full_years
ORDER BY 2 DESC
LIMIT 10;

--Start/End_station_id, ride_count for each, docks, DIFFERENCE IN COUNT for start/end
with start_station AS (SELECT b9.start_station_id, COUNT(*) AS ride_count, bs.docks
FROM baywheels_2019 b9
JOIN baywheels_stations bs
        ON b9.start_station_id = bs.id
GROUP BY 1, 3
ORDER BY 2 DESC),

end_station AS (
SELECT b9.end_station_id, COUNT(*) AS ride_count, bs.docks
FROM baywheels_2019 b9
JOIN baywheels_stations bs
        ON b9.end_station_id = bs.id
GROUP BY 1, 3
ORDER BY 2 DESC)

SELECT *, (e.ride_count - s.ride_count)/365 as difference
FROM start_station s
JOIN end_station e
on s.start_station_id = e.end_station_id
ORDER BY 7 DESC;

--REVISED ABOVE -Start/End_station_id, ride_count for each, docks, DIFFERENCE IN COUNT for
start/end.  (Thank you Claudia!)
```

```
with start_station AS (SELECT b9.start_station_id, COUNT(*) AS ride_count, bs.docks
FROM baywheels_2019 b9
JOIN baywheels_stations bs
        ON b9.start_station_id = bs.id
GROUP BY 1, 3
ORDER BY 2 DESC),

end_station AS (
SELECT b9.end_station_id, COUNT(*) AS ride_count, bs.docks
FROM baywheels_2019 b9
JOIN baywheels_stations bs
        ON b9.end_station_id = bs.id
GROUP BY 1, 3
ORDER BY 2 DESC)

SELECT s.start_station_id as station_id, s.ride_count start_ride_count,
        e.ride_count end_ride_count,
        (e.ride_count - s.ride_count)/365 as eod_end_minus_start_ride,
        e.docks, e.docks - (e.ride_count - s.ride_count)/365 as eod_docks_minus_bike_count
FROM start_station s
JOIN end_station e
ON s.start_station_id = e.end_station_id
ORDER BY 4 DESC;
```

```
--Start/End_station_id, ride_count for each, docks, DIFFERENCE IN COUNT for start/end
WITH start_station AS (SELECT b9.start_station_id, COUNT(*) AS ride_count, bs.docks
FROM baywheels_2019 b9
JOIN baywheels_stations bs
        ON b9.start_station_id = bs.id
GROUP BY 1, 3
ORDER BY 2 DESC),

end_station AS (
SELECT b9.end_station_id, COUNT(*) AS ride_count, bs.docks
FROM baywheels_2019 b9
JOIN baywheels_stations bs
        ON b9.end_station_id = bs.id
GROUP BY 1, 3
ORDER BY 2 DESC)

SELECT s.start_station_id as station_id, s.ride_count start_ride_count,
        e.ride_count end_ride_count,
        (e.ride_count - s.ride_count)/365 as eod_end_minus_start_ride,
```

```
        e.docks, e.docks - (e.ride_count - s.ride_count)/365 as eod_docks_minus_bike_count
FROM start_station s
JOIN end_station e
ON s.start_station_id = e.end_station_id
ORDER BY 4 DESC;

SELECT DATE_PART('year', year_day) as year, ROUND(AVG(daily_count), 1) as avg_daily_rides
FROM (
        SELECT DATE_TRUNC('day', start_time) as year_day, COUNT(*) as daily_count
        FROM (
                SELECT *
                FROM baywheels_2017
                UNION
                SELECT *
                FROM baywheels_2018
                UNION
                SELECT *
                FROM baywheels_2019) as all_years
        GROUP BY 1
        ORDER BY 1) as monthly_rides
GROUP BY 1;

SELECT to_char(DATE_TRUNC('day', MIN(start_time)), 'yyyy-mm-dd') as first_day,
   to_char(DATE_TRUNC('day', MAX(start_time)), 'yyyy-mm-dd') as last_day
FROM
   (SELECT *
   FROM baywheels_2017
   UNION
   SELECT *
   FROM baywheels_2018
   UNION
   SELECT *
   FROM baywheels_2019) as all_years

--------------------------------------------------------------------------------
WITH all_years AS
       (SELECT *
                  FROM baywheels_2017
       UNION
       SELECT *
       FROM baywheels_2018
       UNION
       SELECT *
       FROM baywheels_2019),
```

```
 monthly_rides AS
        (SELECT DATE_TRUNC('month', start_time) as year_month, COUNT(*) as monthly_count
    FROM all_years a
    GROUP BY 1
    ORDER BY 1)


SELECT DATE_PART('year', year_month) as year, ROUND(AVG(monthly_count), 1) as
avg_monthly_rides
FROM monthly_rides m
GROUP BY 1
ORDER BY 2 DESC;
```