--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;