**Cyclistic Data Cleaning and Exploration with SQL Queries**

/\*

Cyclistic Bike Data Cleaning and Exploration in SQL Queries

Skills Used: Temporary Table, CTE, Aggregate Functions, Window Function, Timestamp Function, Conditional Function, Subquery

Platform: GCP BigQuery

\*/

--------------------------------------------------------------------------------------------

-- DATA CLEANING IN SQL QUERIES

-- Preview of the four tables

SELECT

    \*

FROM

    `causal-bison-323215.Cyclistic.DivvyTrips2019Q1` LIMIT 20;

SELECT

    \*

FROM

    `causal-bison-323215.Cyclistic.DivvyTrips2019Q2` LIMIT 20;

SELECT

    \*

FROM

    `causal-bison-323215.Cyclistic.DivvyTrips2019Q3` LIMIT 20;

SELECT

    \*

FROM

    `causal-bison-323215.Cyclistic.DivvyTrips2019Q4` LIMIT 20;

-- Change DivvyTrips2019Q2 column names to same as Q1, Q3 and Q4 using temp table

BEGIN

CREATE OR REPLACE TEMPORARY TABLE DivvyTrips2019Q2Temp

AS

SELECT

    \_01\_\_\_Rental\_Details\_Rental\_ID AS trip\_id,

    \_01\_\_\_Rental\_Details\_Local\_Start\_Time AS start\_time,

    \_01\_\_\_Rental\_Details\_Local\_End\_Time AS end\_time,

    \_01\_\_\_Rental\_Details\_Bike\_ID AS bikeid,

    \_01\_\_\_Rental\_Details\_Duration\_In\_Seconds\_Uncapped AS tripduration,

    \_03\_\_\_Rental\_Start\_Station\_ID AS from\_station\_id,

    \_03\_\_\_Rental\_Start\_Station\_Name AS from\_station\_name,

    \_02\_\_\_Rental\_End\_Station\_ID AS to\_station\_id,

    \_02\_\_\_Rental\_End\_Station\_Name AS to\_station\_name,

    User\_Type AS usertype,

    Member\_Gender AS gender,

    \_05\_\_\_Member\_Details\_Member\_Birthday\_Year AS birthyear

FROM

    `causal-bison-323215.Cyclistic.DivvyTrips2019Q2`;

End;

-- Temp table DivvyTrips2019Q2Temp stored as causal-bison-323215.\_script2cb74db8e872be8c7189bb08b16c7d5c3fba71e6.DivvyTrips2019Q2Temp

SELECT

    \*

FROM

    `causal-bison-323215.\_script2cb74db8e872be8c7189bb08b16c7d5c3fba71e6.DivvyTrips2019Q2Temp` LIMIT 10

-- Merge DivvyTrips2019Q, DivvyTrips2019Q2Temp, DivvyTrips2019Q3 and DivvyTrips2019Q4 into one table

BEGIN

CREATE OR REPLACE TEMP TABLE DivvyTrips2019

AS

SELECT

    \*

FROM

    `causal-bison-323215.Cyclistic.DivvyTrips2019Q1`

UNION ALL

SELECT

    \*

FROM

    `causal-bison-323215.\_script2cb74db8e872be8c7189bb08b16c7d5c3fba71e6.DivvyTrips2019Q2Temp`

UNION ALL

SELECT

    \*

FROM

    `causal-bison-323215.Cyclistic.DivvyTrips2019Q3`

UNION ALL

SELECT

    \*

FROM

    `causal-bison-323215.Cyclistic.DivvyTrips2019Q4`;

END;

-- Temp table DivvyTrips2019 stored`causal-bison-323215.\_script203775267b33d1a1aef3c416b8f33c8849058ea0.DivvyTrips2019`

SELECT

    \*

FROM

    `causal-bison-323215.\_script203775267b33d1a1aef3c416b8f33c8849058ea0.DivvyTrips2019`

-- Alternatively, one could create a CTE to rename the columns in DivvyTrips2019Q2, union the CTE with the other three tables and save the result into a new table, DivvyTrips2019V2

WITH NewDivvyTrips2019Q2 AS (SELECT

    \_01\_\_\_Rental\_Details\_Rental\_ID AS trip\_id,

    \_01\_\_\_Rental\_Details\_Local\_Start\_Time AS start\_time,

    \_01\_\_\_Rental\_Details\_Local\_End\_Time AS end\_time,

    \_01\_\_\_Rental\_Details\_Bike\_ID AS bikeid,

    \_01\_\_\_Rental\_Details\_Duration\_In\_Seconds\_Uncapped AS tripduration,

    \_03\_\_\_Rental\_Start\_Station\_ID AS from\_station\_id,

    \_03\_\_\_Rental\_Start\_Station\_Name AS from\_station\_name,

    \_02\_\_\_Rental\_End\_Station\_ID AS to\_station\_id,

    \_02\_\_\_Rental\_End\_Station\_Name AS to\_station\_name,

    User\_Type AS usertype,

    Member\_Gender AS gender,

    \_05\_\_\_Member\_Details\_Member\_Birthday\_Year AS birthyear

FROM

    `causal-bison-323215.Cyclistic.DivvyTrips2019Q2`

)

SELECT

    \*

FROM

    `causal-bison-323215.Cyclistic.DivvyTrips2019Q1`

UNION ALL

SELECT

    \*

FROM

    NewDivvyTrips2019Q2

UNION ALL

SELECT

    \*

FROM

    `causal-bison-323215.Cyclistic.DivvyTrips2019Q3`

UNION ALL

SELECT

    \*

FROM

    `causal-bison-323215.Cyclistic.DivvyTrips2019Q4`;

-- Check if the primary key column, trip-id, has duplicate value

SELECT

    COUNT(DISTINCT trip\_id) AS unique\_count\_tripId,

    COUNT(\*) AS total\_number\_rows

FROM

    `causal-bison-323215.\_script203775267b33d1a1aef3c416b8f33c8849058ea0.DivvyTrips2019`

-- Check if columns contained nulls

SELECT

    COUNTIF(trip\_id IS NULL) AS trip\_id,

    COUNTIF(start\_time IS NULL) AS start\_time,

    COUNTIF(end\_time IS NULL) AS end\_time,

    COUNTIF(tripduration IS NULL) AS tripduration,

    COUNTIF(bikeid IS NULL) AS bikeid,

    COUNTIF(from\_station\_id IS NULL) AS from\_station\_id,

    COUNTIF(from\_station\_name IS NULL) AS from\_station\_name,

    COUNTIF(to\_station\_id IS NULL) AS to\_station\_id,

    COUNTIF(to\_station\_name IS NULL) AS to\_station\_name,

    COUNTIF(usertype IS NULL) AS usertype,

    COUNTIF(gender IS NULL) AS gender,

    COUNTIF(birthyear IS NULL) AS birthyear

FROM

    `causal-bison-323215.\_script203775267b33d1a1aef3c416b8f33c8849058ea0.DivvyTrips2019`;

-- Examine the birthyear column

SELECT

    MIN(birthyear) AS min\_birthyear,

    MAX(birthyear) AS max\_birthyear

FROM

    `causal-bison-323215.\_script203775267b33d1a1aef3c416b8f33c8849058ea0.DivvyTrips2019`;

-- Update the gender column; replace nulls with 'Unknown'

UPDATE `causal-bison-323215.\_script203775267b33d1a1aef3c416b8f33c8849058ea0.DivvyTrips2019`

SET gender = 'Unknown'

WHERE gender IS NULL;

SELECT

    \*

FROM

    `causal-bison-323215.\_script203775267b33d1a1aef3c416b8f33c8849058ea0.DivvyTrips2019`

WHERE

    gender IS NULL;

-- Create and populate colum ride\_length

ALTER TABLE `causal-bison-323215.\_script203775267b33d1a1aef3c416b8f33c8849058ea0.DivvyTrips2019`

ADD COLUMN ride\_length INTEGER;

UPDATE `causal-bison-323215.\_script203775267b33d1a1aef3c416b8f33c8849058ea0.DivvyTrips2019`

SET ride\_length = TIMESTAMP\_DIFF(end\_time, start\_time, SECOND)

WHERE trip\_id IS NOT NULL;

SELECT

    MIN(ride\_length) AS min\_ride\_duration,

    MAX(ride\_length) AS max\_ride\_duration

FROM

    `causal-bison-323215.\_script203775267b33d1a1aef3c416b8f33c8849058ea0.DivvyTrips2019`;

SELECT

    COUNT(\*) AS rows\_with\_negative\_ride\_length,

FROM

    `causal-bison-323215.\_script203775267b33d1a1aef3c416b8f33c8849058ea0.DivvyTrips2019`

WHERE

    ride\_length <= 0;

-- Delete rows with negative ride\_length value

DELETE

FROM

    `causal-bison-323215.\_script203775267b33d1a1aef3c416b8f33c8849058ea0.DivvyTrips2019`

WHERE

    ride\_length <= 0;

SELECT

    COUNT(\*) AS rows\_with\_negative\_ride\_length,

FROM

    `causal-bison-323215.\_script203775267b33d1a1aef3c416b8f33c8849058ea0.DivvyTrips2019`

WHERE

    ride\_length <= 0;

SELECT

    MIN(ride\_length) AS min\_ride\_duration,

    MAX(ride\_length) AS max\_ride\_duration

FROM

    `causal-bison-323215.\_script203775267b33d1a1aef3c416b8f33c8849058ea0.DivvyTrips2019`;

-- Create and populate day\_of\_week column

ALTER TABLE `causal-bison-323215.\_script203775267b33d1a1aef3c416b8f33c8849058ea0.DivvyTrips2019`

ADD COLUMN day\_of\_week INTEGER;

UPDATE `causal-bison-323215.\_script203775267b33d1a1aef3c416b8f33c8849058ea0.DivvyTrips2019`

SET day\_of\_week = EXTRACT(DAYOFWEEK FROM start\_time)

WHERE trip\_id IS NOT NULL;

SELECT

    \*

FROM

    `causal-bison-323215.\_script203775267b33d1a1aef3c416b8f33c8849058ea0.DivvyTrips2019` LIMIT 10;

SELECT

    MIN(day\_of\_week) AS min\_day\_of\_week,

    MAX(day\_of\_week) AS max\_day\_of\_week

FROM

    `causal-bison-323215.\_script203775267b33d1a1aef3c416b8f33c8849058ea0.DivvyTrips2019`;

-- Create and populate day column

ALTER TABLE `causal-bison-323215.\_script203775267b33d1a1aef3c416b8f33c8849058ea0.DivvyTrips2019`

ADD COLUMN day STRING;

UPDATE `causal-bison-323215.\_script203775267b33d1a1aef3c416b8f33c8849058ea0.DivvyTrips2019`

SET day = CASE day\_of\_week

    WHEN 1 THEN 'Sunday'

    WHEN 2 THEN 'Monday'

    WHEN 3 THEN 'Tuesday'

    WHEN 4 THEN 'Wednesday'

    WHEN 5 THEN 'Thursday'

    WHEN 6 THEN 'Friday'

    WHEN 7 THEN 'Saturday'

    ELSE NULL

    END

WHERE trip\_id IS NOT NULL;

SELECT

    \*

FROM

    `causal-bison-323215.\_script203775267b33d1a1aef3c416b8f33c8849058ea0.DivvyTrips2019` LIMIT 10;

SELECT

    COUNTIF(day IS NULL) AS count

FROM

    `causal-bison-323215.\_script203775267b33d1a1aef3c416b8f33c8849058ea0.DivvyTrips2019`;

--------------------------------------------------------------------------------------------

-- EXPLORATORY DATA ANALYSIS WITH SQL QUERIES

-- Different types of usertype and their breakdown by count and gender

SELECT

    usertype AS user,

    gender AS gender,

    COUNT(\*) AS number

FROM

    `causal-bison-323215.\_script203775267b33d1a1aef3c416b8f33c8849058ea0.DivvyTrips2019`

GROUP BY usertype, gender

ORDER BY usertype, gender;

-- Average, lowest and highest ride duration, number of trips and busiest day of 2019

SELECT

    ROUND(AVG(ride\_length), 2) AS average\_ride\_duration\_sec,

    MIN(ride\_length) AS lowest\_ride\_duration\_sec,

    MAX(ride\_length) AS highest\_ride\_duration\_sec,

    COUNT(trip\_id) AS total\_trip,

    APPROX\_TOP\_COUNT(day, 1) AS busiest\_day

FROM

    `causal-bison-323215.\_script203775267b33d1a1aef3c416b8f33c8849058ea0.DivvyTrips2019`;

-- Average, lowest and highest ride duration, number of trips and busiest day for each type of user

SELECT

    usertype AS user,

    ROUND(AVG(ride\_length), 2) AS average\_ride\_duration\_sec,

    MIN(ride\_length) AS lowest\_ride\_duration\_sec,

    MAX(ride\_length) AS highest\_ride\_duration\_sec,

    COUNT(trip\_id) AS total\_trip,

    APPROX\_TOP\_COUNT(day, 1) AS busiest\_day

FROM

    `causal-bison-323215.\_script203775267b33d1a1aef3c416b8f33c8849058ea0.DivvyTrips2019`

GROUP BY usertype;

-- Proportion of rides above average ride duration for each type of user

SELECT

    usertype AS user,

    ROUND(AVG(ride\_length), 2) AS avg\_ride\_duration\_sec,

    COUNTIF((ride\_length - avg\_ride\_duration) > 0) AS ride\_above\_avg\_duration,

    COUNT(trip\_id) AS total\_trip,

    ROUND(COUNTIF((ride\_length - avg\_ride\_duration) > 0)/COUNT(trip\_id)\*100, 2) AS percent\_ride\_above\_avg\_duration,

FROM (

    SELECT

    usertype,

    ride\_length,

    trip\_id,

    AVG(ride\_length) OVER (

    PARTITION BY usertype

    ROWS BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING)

    AS avg\_ride\_duration,

    FROM

    `causal-bison-323215.\_script203775267b33d1a1aef3c416b8f33c8849058ea0.DivvyTrips2019`)

GROUP BY usertype;

-- Average ride duration for users by the day of the week

SELECT

    usertype AS user,

    day AS day,

    ROUND(AVG(ride\_length), 2) AS avg\_ride\_duration,

FROM

    `causal-bison-323215.\_script203775267b33d1a1aef3c416b8f33c8849058ea0.DivvyTrips2019`

GROUP BY usertype, day

ORDER BY usertype;

-- Number of rides for users by day of the week

SELECT

    usertype AS user,

    day AS day,

    COUNT(trip\_id) AS total\_trip

FROM

    `causal-bison-323215.\_script203775267b33d1a1aef3c416b8f33c8849058ea0.DivvyTrips2019`

GROUP BY usertype, day

ORDER BY usertype;

-- Average ride duration for users by the day of the week by quarter

SELECT

    usertype AS user,

    EXTRACT(QUARTER FROM start\_time) AS quarter,

    day AS day,

    ROUND(AVG(ride\_length), 2) AS avg\_ride\_duration

FROM

    `causal-bison-323215.\_script203775267b33d1a1aef3c416b8f33c8849058ea0.DivvyTrips2019`

GROUP BY usertype, quarter, day

ORDER BY usertype, quarter;

-- Number of rides for users by day of the week by quarter

SELECT

    usertype AS user,

    EXTRACT(QUARTER FROM start\_time) AS quarter,

    day AS day,

    COUNT(trip\_id) AS total\_trip

FROM

    `causal-bison-323215.\_script203775267b33d1a1aef3c416b8f33c8849058ea0.DivvyTrips2019`

GROUP BY usertype, quarter, day

ORDER BY usertype, quarter;

-- Save temporary table as BigQuery table for export in order to visualize with Tableau

SELECT

    trip\_id,

    start\_time,

    end\_time,

    bikeid,

    from\_station\_id,

    from\_station\_name,

    to\_station\_id,

    to\_station\_name,

    usertype,

    ride\_length,

    day

FROM

    `causal-bison-323215.\_script203775267b33d1a1aef3c416b8f33c8849058ea0.DivvyTrips2019`;