/\*MAIN FILE\*/

-- MERGED ALL DATA

-- For combining monthly into a full year

CREATE OR REPLACE TABLE `apt-passage-408907.BikeShare\_TripData.YearlyData`

AS

SELECT \* FROM `apt-passage-408907.BikeShare\_TripData.2210\_Oct`

UNION ALL

SELECT \* FROM `apt-passage-408907.BikeShare\_TripData.2211\_Nov`

UNION ALL

SELECT \* FROM `apt-passage-408907.BikeShare\_TripData.2212\_Dec`

UNION ALL

SELECT \* FROM `apt-passage-408907.BikeShare\_TripData.2301\_Jan`

UNION ALL

SELECT \* FROM `apt-passage-408907.BikeShare\_TripData.2302\_Feb`

UNION ALL

SELECT \* FROM `apt-passage-408907.BikeShare\_TripData.2303\_Mar`

UNION ALL

SELECT \* FROM `apt-passage-408907.BikeShare\_TripData.2304\_Apr`

UNION ALL

SELECT \* FROM `apt-passage-408907.BikeShare\_TripData.2305\_May`

UNION ALL

SELECT \* FROM `apt-passage-408907.BikeShare\_TripData.2306\_Jun`

UNION ALL

SELECT \* FROM `apt-passage-408907.BikeShare\_TripData.2307\_Jul`

UNION ALL

SELECT \* FROM `apt-passage-408907.BikeShare\_TripData.2308\_Aug`

UNION ALL

SELECT \* FROM `apt-passage-408907.BikeShare\_TripData.2309\_Sep`;

-- NEW VARIABLES

-- Add ‘rental\_length’ or rental duration

-- For testing

SELECT started\_at, ended\_at, ROUND(TIMESTAMP\_DIFF(ended\_at, started\_at, MINUTE)) AS rental\_length

FROM `apt-passage-408907.BikeShare\_TripData.YearlyData`;

-- Add the new variable

ALTER TABLE `apt-passage-408907.BikeShare\_TripData.YearlyData`

ADD COLUMN rental\_length FLOAT64;

-- Update the new columns with values

UPDATE `apt-passage-408907.BikeShare\_TripData.YearlyData`

SET rental\_length = ROUND(TIMESTAMP\_DIFF(ended\_at, started\_at, MINUTE))

WHERE ride\_id IS NOT NULL;

-- Add ‘distance\_in\_kilometers’

-- Creating the function

CREATE OR REPLACE FUNCTION `apt-passage-408907.BikeShare\_TripData.distance\_in\_km`

(

start\_lat FLOAT64,

end\_lat FLOAT64,

start\_lng FLOAT64,

end\_lng FLOAT64

)

RETURNS FLOAT64

LANGUAGE js AS """

// Constants

const R = 6367.45; // Kilometers

// Convert latitude and longitude from degrees to radians

const dLat = (end\_lat - start\_lat) \* Math.PI / 180;

const dLon = (end\_lng - start\_lng) \* Math.PI / 180;

// Haversine formula

const a = Math.sin(dLat / 2) \* Math.sin(dLat / 2) +

Math.cos(start\_lat \* Math.PI / 180) \* Math.cos(end\_lat \* Math.PI / 180) \*

Math.sin(dLon / 2) \* Math.sin(dLon / 2);

const c = 2 \* Math.asin(Math.min(Math.sqrt(a), 1));

// Calculate distance

const distance = R \* c;

return distance;

""";

-- Add the new variable

ALTER TABLE `apt-passage-408907.BikeShare\_TripData.YearlyData`

ADD COLUMN distance\_in\_km FLOAT64;

-- Update the new columns with values

UPDATE `apt-passage-408907.BikeShare\_TripData.YearlyData`

SET distance\_in\_km = `apt-passage-408907.BikeShare\_TripData.distance\_in\_km`(start\_lat, end\_lat, start\_lng, end\_lng)

WHERE ride\_id IS NOT NULL;

-- Add a year-quarter column

-- Add the new variable

ALTER TABLE `apt-passage-408907.BikeShare\_TripData.YearlyData`

ADD COLUMN quarter\_new STRING;

-- Update the new columns with values

UPDATE `apt-passage-408907.BikeShare\_TripData.YearlyData`

SET quarter\_new = FORMAT\_TIMESTAMP('%Y Q%Q', started\_at)

WHERE ride\_id IS NOT NULL;

-- Add a year-month column

-- Add the new variable

ALTER TABLE `apt-passage-408907.BikeShare\_TripData.YearlyData`

ADD COLUMN start\_year\_month STRING;

-- Update the new columns with values extracted from 'started\_at'

UPDATE `apt-passage-408907.BikeShare\_TripData.YearlyData`

SET start\_year\_month = FORMAT\_TIMESTAMP('%Y %m', started\_at)

WHERE ride\_id IS NOT NULL;

-- Add a weekpart column

-- Add the new variable

ALTER TABLE `apt-passage-408907.BikeShare\_TripData.YearlyData`

ADD COLUMN weekpart STRING;

-- Update the new columns with values

UPDATE `apt-passage-408907.BikeShare\_TripData.YearlyData`

SET

weekpart =

CASE

WHEN EXTRACT(DAYOFWEEK FROM started\_at) BETWEEN 2 AND 6 THEN 'weekday'

ELSE 'weekend'

END

WHERE ride\_id IS NOT NULL;

-- Add a new column for the day the trip was taken

-- Add the new variable

ALTER TABLE `apt-passage-408907.BikeShare\_TripData.YearlyData`

ADD COLUMN name\_of\_day STRING;

-- Update the new columns with values

UPDATE `apt-passage-408907.BikeShare\_TripData.YearlyData`

SET name\_of\_day = FORMAT\_DATE('%A', DATE(started\_at))

WHERE ride\_id IS NOT NULL;

-- Add a new column for the time/hour the trip was taken

-- Add the new variable

ALTER TABLE `apt-passage-408907.BikeShare\_TripData.YearlyData`

ADD COLUMN timeofday\_start STRING,

ADD COLUMN timeofday\_end STRING;

-- Update the new columns with values

UPDATE `apt-passage-408907.BikeShare\_TripData.YearlyData`

SET

timeofday\_start = FORMAT\_TIMESTAMP('%H:00', TIMESTAMP(started\_at)),

timeofday\_end = FORMAT\_TIMESTAMP('%H:00', TIMESTAMP(ended\_at))

WHERE ride\_id IS NOT NULL;

-- CLEANING

-- Checks for duplicates

SELECT ride\_id, COUNT(\*)

FROM `apt-passage-408907.BikeShare\_TripData.YearlyData`

GROUP BY ride\_id

HAVING COUNT(\*) > 1;

SELECT ride\_id,

FROM `apt-passage-408907.BikeShare\_TripData.YearlyData` AS yc

WHERE ride\_id = "6.32E+15";

-- Remove duplicates

-- Made a temp file to delete rows where row\_num > 1 (keeping only the first occurrence of each ride\_id)

CREATE OR REPLACE TABLE `apt-passage-408907.BikeShare\_TripData.temp`

AS

SELECT

\*,

ROW\_NUMBER() OVER (PARTITION BY ride\_id ORDER BY ride\_id) AS row\_num

FROM `apt-passage-408907.BikeShare\_TripData.YearlyData`;

-- Delete rows where row\_num > 1 (keeping only the first occurrence of each ride\_id)

DELETE FROM `apt-passage-408907.BikeShare\_TripData.temp`

WHERE row\_num > 1;

-- Check membership and bike type

-- Check for distinct values in a column

SELECT DISTINCT member\_casual

FROM `apt-passage-408907.BikeShare\_TripData.temp`;

SELECT DISTINCT rideable\_type

FROM `apt-passage-408907.BikeShare\_TripData.temp`;

-- Check for leading and trailing spaces in a column

SELECT member\_casual

FROM `apt-passage-408907.BikeShare\_TripData.temp`

WHERE LENGTH(member\_casual) != LENGTH(TRIM(member\_casual));

-- Check missing data on latitude and longitude

SELECT

COUNT(CASE WHEN start\_lat IS NULL THEN 1 END) AS null\_start\_lat\_count,

COUNT(CASE WHEN start\_lng IS NULL THEN 1 END) AS null\_start\_lng\_count,

COUNT(CASE WHEN end\_lat IS NULL THEN 1 END) AS null\_end\_lat\_count,

COUNT(CASE WHEN end\_lng IS NULL THEN 1 END) AS null\_end\_lng\_count

FROM

`apt-passage-408907.BikeShare\_TripData.temp`;

-- Delete missing location data

DELETE FROM `apt-passage-408907.BikeShare\_TripData.temp`

WHERE end\_lat IS NULL OR end\_lng IS NULL;

-- Check rental\_length

SELECT

COUNT(rental\_length) AS rental\_count,

AVG(rental\_length) AS avg\_rental,

MIN(rental\_length) AS min\_rental,

MAX(rental\_length) AS max\_rental

FROM `apt-passage-408907.BikeShare\_TripData.temp`;

-- Remove rental\_length values <= 0

DELETE FROM `apt-passage-408907.BikeShare\_TripData.temp`

WHERE rental\_length <= 0;

/\*STATIONS FILE\*/

-- MERGED ALL DATA

-- For combining stations data into a full year

CREATE TABLE `apt-passage-408907.BikeShare\_TripData.YearlyStationsData` AS

SELECT \* FROM `apt-passage-408907.BikeShare\_TripData.Station\_2210\_2211\_3`

UNION ALL

SELECT \* FROM `apt-passage-408907.BikeShare\_TripData.Station\_2212\_2303\_2`

UNION ALL

SELECT \* FROM `apt-passage-408907.BikeShare\_TripData.Station\_2304\_2305`

UNION ALL

SELECT \* FROM `apt-passage-408907.BikeShare\_TripData.Station\_2306`

UNION ALL

SELECT \* FROM `apt-passage-408907.BikeShare\_TripData.Station\_2307`

UNION ALL

SELECT \* FROM `apt-passage-408907.BikeShare\_TripData.Station\_2308`

UNION ALL

SELECT \* FROM `apt-passage-408907.BikeShare\_TripData.Station\_2309`;

/\*CLEANING\*/

-- Remove duplicates from ride ids, made a temp stations data

CREATE OR REPLACE TABLE `apt-passage-408907.BikeShare\_TripData.tempstations`

AS

SELECT

\*,

ROW\_NUMBER() OVER (PARTITION BY ride\_id ORDER BY ride\_id) AS row\_num

FROM `apt-passage-408907.BikeShare\_TripData.YearlyStationsData`;

-- Delete rows where row\_num > 1 (keeping only the first occurrence of each ride\_id)

DELETE FROM `apt-passage-408907.BikeShare\_TripData.tempstations`

WHERE row\_num > 1;

-- Check missing data on station names

SELECT

COUNT(CASE WHEN start\_station\_name IS NULL THEN 1 END) AS start\_station\_name,

COUNT(CASE WHEN start\_station\_id IS NULL THEN 1 END) AS start\_station\_name,

COUNT(CASE WHEN end\_station\_name IS NULL THEN 1 END) AS start\_station\_name,

COUNT(CASE WHEN end\_station\_id IS NULL THEN 1 END) AS start\_station\_name,

FROM

`apt-passage-408907.BikeShare\_TripData.tempstations`;

-- Drop unnecessary columns

ALTER TABLE `apt-passage-408907.BikeShare\_TripData.tempstations`

DROP COLUMN start\_station\_id;

ALTER TABLE `apt-passage-408907.BikeShare\_TripData.tempstations`

DROP COLUMN end\_station\_id;

-- Add station names in Main file

ALTER TABLE `apt-passage-408907.BikeShare\_TripData.temp`

ADD COLUMN start\_station\_name STRING,

ADD COLUMN end\_station\_name STRING;

-- Merge stations vars in the temp file by using LEFT JOIN

UPDATE `apt-passage-408907.BikeShare\_TripData.temp` AS y

SET

start\_station\_name = s.start\_station\_name,

end\_station\_name = s.end\_station\_name

FROM

`apt-passage-408907.BikeShare\_TripData.tempstations` AS s

WHERE

y.ride\_id = s.ride\_id;

/\*FINAL FILES\*/

-- Create a new table without duplicates

CREATE OR REPLACE TABLE `apt-passage-408907.BikeShare\_TripData.YearlyCleanedFinal` AS

SELECT \*

FROM `apt-passage-408907.BikeShare\_TripData.temp`;

CREATE OR REPLACE TABLE `apt-passage-408907.BikeShare\_TripData.YearlyStationsFinal` AS

SELECT \*

FROM `apt-passage-408907.BikeShare\_TripData.tempstations`;

/\*DATA RUNS\*/

-- Trips by Membership Type

SELECT

COUNT (\*) AS Total,

COUNTIF(member\_casual = 'member') AS member\_trips,

COUNTIF(member\_casual = 'casual') AS casual\_trips

FROM `apt-passage-408907.BikeShare\_TripData.YearlyCleanedFinal`;

-- Trips by Quarter

SELECT

quarter\_new,

COUNT (\*) AS Total,

COUNTIF(member\_casual = 'member') AS member\_trips,

COUNTIF(member\_casual = 'casual') AS casual\_trips

FROM `apt-passage-408907.BikeShare\_TripData.YearlyCleanedFinal`

GROUP BY

quarter\_new

ORDER BY

quarter\_new;

-- Trips by Month

SELECT

start\_year\_month,

COUNT (\*) AS Total,

COUNTIF(member\_casual = 'member') AS member\_trips,

COUNTIF(member\_casual = 'casual') AS casual\_trips

FROM `apt-passage-408907.BikeShare\_TripData.YearlyCleanedFinal`

GROUP BY

start\_year\_month

ORDER BY

start\_year\_month;

-- Bike Type by Membership

SELECT

membership2,

rideable\_type,

COUNT(\*) AS count\_rideable,

100 \* COUNT(\*) / SUM(COUNT(\*)) OVER (PARTITION BY membership2) AS percent\_of\_total

FROM `apt-passage-408907.BikeShare\_TripData.YearlyCleanedFinal`

GROUP BY membership2, rideable\_type

ORDER BY membership2 DESC;

-- Weekpart by Membership

SELECT

membership2,

weekpart,

COUNT(\*) AS count\_rideable,

100 \* COUNT(\*) / SUM(COUNT(\*)) OVER (PARTITION BY membership2) AS percent\_of\_total

FROM `apt-passage-408907.BikeShare\_TripData.YearlyCleanedFinal`

GROUP BY membership2, weekpart

ORDER BY membership2 DESC;

-- Weekpart and Time of Day by Membership

SELECT

membership2,

weekpart,

timeofday\_start,

COUNT(\*) AS count\_rideable,

FROM `apt-passage-408907.BikeShare\_TripData.YearlyCleanedFinal`

GROUP BY membership2, weekpart,timeofday\_start

ORDER BY membership2 DESC, weekpart,timeofday\_start;

-- Average Rental and Distance by Membership

SELECT

membership2,

AVG(rental\_length),

AVG(distance\_in\_km),

COUNT(\*) AS count\_rideable,

FROM `apt-passage-408907.BikeShare\_TripData.YearlyCleanedFinal`

GROUP BY membership2

ORDER BY membership2;

-- Day by Membership

SELECT

name\_of\_day,

membership2,

COUNT(\*) AS count\_rideable,

FROM `apt-passage-408907.BikeShare\_TripData.YearlyCleanedFinal`

GROUP BY membership2, name\_of\_day

ORDER BY

CASE name\_of\_day

WHEN 'Sunday' THEN 1

WHEN 'Monday' THEN 2

WHEN 'Tuesday' THEN 3

WHEN 'Wednesday' THEN 4

WHEN 'Thursday' THEN 5

WHEN 'Friday' THEN 6

ELSE 7

END;

-- Avg Rental Length by Weekpart by Membership

SELECT

membership2,

weekpart,

AVG(rental\_length) AS rental\_length

FROM `apt-passage-408907.BikeShare\_TripData.YearlyCleanedFinal`

GROUP BY membership2, weekpart

ORDER BY membership2 DESC, weekpart;

-- Total Count, Avg Rental Length by Bike Type by Membership

SELECT

membership2,

rideable\_type,

COUNT(\*) AS count\_trips,

AVG(rental\_length) AS rental\_length,

FROM `apt-passage-408907.BikeShare\_TripData.YearlyCleanedFinal`

GROUP BY membership2, rideable\_type

ORDER BY membership2 DESC, rideable\_type DESC;

-- Stations

SELECT

membership2,

start\_station\_name,

COUNT(\*) AS count\_rideable

FROM `apt-passage-408907.BikeShare\_TripData.YearlyCleanedFinal`

WHERE membership2 = "Member" AND start\_station\_name IS NOT NULL

GROUP BY membership2, start\_station\_name

ORDER BY count\_rideable DESC

LIMIT 5;

SELECT

membership2,

end\_station\_name,

COUNT(\*) AS count\_rideable

FROM `apt-passage-408907.BikeShare\_TripData.YearlyCleanedFinal`

WHERE membership2 = "Member" AND end\_station\_name IS NOT NULL

GROUP BY membership2, end\_station\_name

ORDER BY count\_rideable DESC

LIMIT 5;

SELECT

membership2,

start\_station\_name,

COUNT(\*) AS count\_rideable

FROM `apt-passage-408907.BikeShare\_TripData.YearlyCleanedFinal`

WHERE membership2 = "Casual User" AND start\_station\_name IS NOT NULL

GROUP BY membership2, start\_station\_name

ORDER BY count\_rideable DESC

LIMIT 5;

SELECT

membership2,

end\_station\_name,

COUNT(\*) AS count\_rideable

FROM `apt-passage-408907.BikeShare\_TripData.YearlyCleanedFinal`

WHERE membership2 = "Casual User" AND end\_station\_name IS NOT NULL

GROUP BY membership2, end\_station\_name

ORDER BY count\_rideable DESC

LIMIT 5;