--COMBINING ALL 12 MONTHS DATA TOGETHER WITH UNION COMMAND

CREATE TABLE combined\_data (

ride\_id NVARCHAR(MAX),

rideable\_type NVARCHAR(MAX),

started\_at DATETIME2(7),

ended\_at DATETIME2(7),

start\_station\_name NVARCHAR(MAX),

start\_station\_id NVARCHAR(MAX),

end\_station\_name NVARCHAR(MAX),

end\_station\_id NVARCHAR(MAX),

start\_lat FLOAT,

start\_lng FLOAT,

end\_lat FLOAT,

end\_lng FLOAT,

member\_casual NVARCHAR(MAX),

ride\_length TIME(7),

day\_of\_week TINYINT

);

INSERT INTO combined\_data

SELECT \*

FROM [01\_january]

UNION ALL

SELECT \*

FROM [02\_february]

UNION ALL

SELECT \*

FROM [03\_march]

UNION ALL

SELECT \*

FROM [04\_april]

UNION ALL

SELECT \*

FROM [06\_june]

UNION ALL

SELECT \*

FROM [07\_july]

UNION ALL

SELECT \*

FROM [08\_august]

UNION ALL

SELECT \*

FROM [09\_september]

UNION ALL

SELECT \*

FROM [10\_october]

UNION ALL

SELECT \*

FROM [11\_november]

UNION ALL

SELECT \*

FROM [12\_december]

;

--REMOVING NULL VALUES

SELECT \*

INTO cyclistic\_nullsremoved\_tripdata

FROM [cyclistic\_combined\_tripdata]

WHERE ride\_id IS NOT NULL

AND rideable\_type IS NOT NULL

AND started\_at IS NOT NULL

AND ended\_at IS NOT NULL

AND start\_station\_name IS NOT NULL

AND start\_station\_id IS NOT NULL

AND end\_station\_name IS NOT NULL

AND end\_station\_id IS NOT NULL

AND start\_lat IS NOT NULL

AND start\_lng IS NOT NULL

AND end\_lat IS NOT NULL

AND end\_lng IS NOT NULL

AND member\_casual IS NOT NULL

AND ride\_length IS NOT NULL

AND day\_of\_week IS NOT NULL;

-- CREATING A NEW COLUMN IN EXISTING TABLE NAMED "DAY\_OF\_WEEK\_NAME"

ALTER TABLE [cyclistic\_nullsremoved\_tripdata]

ADD day\_of\_week\_name VARCHAR(20) -- Adjust the data type **and** length as needed

SELECT \*

FROM [cyclistic\_nullsremoved\_tripdata];

UPDATE [cyclistic\_nullsremoved\_tripdata]

SET day\_of\_week\_name =

CASE

WHEN day\_of\_week = 1 THEN 'Sunday'

WHEN day\_of\_week = 2 THEN 'Monday'

WHEN day\_of\_week = 3 THEN 'Tuesday'

WHEN day\_of\_week = 4 THEN 'Wednesday'

WHEN day\_of\_week = 5 THEN 'Thursday'

WHEN day\_of\_week = 6 THEN 'Friday'

WHEN day\_of\_week = 7 THEN 'Saturday'

END;

-- CREATING NEW TABLE WITH TRIM COMMAND APPLIED TO REMOVE EXTRA SPACE IN DATASET AND TO GET CLEAN DATA

CREATE TABLE cyclistic\_trimmed\_tripdata(

ride\_id NVARCHAR(MAX),

rideable\_type NVARCHAR(MAX),

started\_at DATETIME2(7),

ended\_at DATETIME2(7),

start\_station\_name NVARCHAR(MAX),

start\_station\_id NVARCHAR(MAX),

end\_station\_name NVARCHAR(MAX),

end\_station\_id NVARCHAR(MAX),

start\_lat FLOAT,

start\_lng FLOAT,

end\_lat FLOAT,

end\_lng FLOAT,

member\_casual NVARCHAR(MAX),

ride\_length TIME(7),

day\_of\_week TINYINT,

day\_of\_week\_name VARCHAR(20)

);

INSERT INTO cyclistic\_trimmed\_tripdata

SELECT

TRIM(ride\_id) AS ride\_id,

TRIM(rideable\_type) AS rideable\_type,

started\_at,

ended\_at,

TRIM(start\_station\_name) AS start\_station\_name,

TRIM(start\_station\_id) AS start\_station\_id,

TRIM(end\_station\_name) AS end\_station\_name,

TRIM(end\_station\_id) AS end\_station\_id,

start\_lat,

start\_lng,

end\_lat,

end\_lng,

TRIM(member\_casual) AS member\_casual,

ride\_length,

day\_of\_week,

TRIM(day\_of\_week\_name) AS day\_of\_week\_name

FROM cyclistic\_nullsremoved\_tripdata;

SELECT \*

FROM cyclistic\_trimmed\_tripdata;

--CASUAL MEMBER COUNT ON START STATION

SELECT start\_station\_name,

start\_station\_id,

COUNT (member\_casual) AS casual\_member\_count

FROM [cyclistic\_trimmed\_tripdata]

WHERE member\_casual = 'casual'

GROUP BY start\_station\_name, start\_station\_id

ORDER BY casual\_member\_count DESC;

--CASUAL MEMBER COUNT ON END STATION

SELECT end\_station\_name,

end\_station\_id,

COUNT (member\_casual) AS casual\_member\_count

FROM [cyclistic\_trimmed\_tripdata]

WHERE member\_casual = 'casual'

GROUP BY end\_station\_name, end\_station\_id

ORDER BY casual\_member\_count DESC;

--ANNUAL MEMBER COUNT ON START STATION

SELECT start\_station\_name,

start\_station\_id,

COUNT (member\_casual) AS annual\_member\_count

FROM [cyclistic\_trimmed\_tripdata]

WHERE member\_casual = 'member'

GROUP BY start\_station\_name, start\_station\_id

ORDER BY annual\_member\_count DESC;

--ANNUAL MEMBER COUNT ON END STATION

SELECT end\_station\_name,

end\_station\_id,

COUNT (member\_casual) AS annual\_member\_count

FROM [cyclistic\_trimmed\_tripdata]

WHERE member\_casual = 'member'

GROUP BY end\_station\_name, end\_station\_id

ORDER BY annaul\_member\_count DESC;

--TWO SEPARTE TABLES FOR ONLY CASUAL RIDERS AND ANNUAL RIDERS

--FOR CASUAL

SELECT \*

INTO cyclistic\_casual\_riders\_tripdata

FROM cyclistic\_trimmed\_tripdata

WHERE 0 = 1;

INSERT INTO cyclistic\_casual\_riders\_tripdata

SELECT \*

FROM cyclistic\_trimmed\_tripdata

WHERE member\_casual = 'casual';

--FOR MEMBER

SELECT \*

INTO cyclistic\_member\_riders\_tripdata

FROM cyclistic\_trimmed\_tripdata

WHERE 0 = 1;

INSERT INTO cyclistic\_member\_riders\_tripdata

SELECT \*

FROM cyclistic\_trimmed\_tripdata

WHERE member\_casual = 'member';

--CREATED 4 TABLES FOR LAT\LONG ROUNDED CASUAL AND MEMBERS FOR START STATIONS AND END STATIONS

CREATE TABLE startstation\_latlong\_rounded\_casual (

start\_station\_name NVARCHAR(MAX),

start\_lat\_round FLOAT,

start\_lng FLOAT,

start\_trips\_per\_station INT);

INSERT INTO startstation\_latlong\_rounded\_casual

SELECT start\_station\_name,

ROUND(AVG(start\_lat), 4) AS start\_lat\_round,

ROUND(AVG(start\_lng), 4) AS start\_lng\_round,

COUNT (\*) AS start\_trips\_per\_station

FROM cyclistic\_casual\_riders\_tripdata

GROUP BY start\_station\_name

ORDER BY start\_trips\_per\_station DESC;

SELECT \*

FROM startstation\_latlong\_rounded\_casual

ORDER BY start\_trips\_per\_station DESC;

--Same way created member table

CREATE TABLE endstation\_latlong\_rounded\_member (

end\_station\_name NVARCHAR(MAX),

end\_lat\_round FLOAT,

end\_lng\_round FLOAT,

end\_trips\_per\_station INT);

INSERT INTO endstation\_latlong\_rounded\_member

SELECT end\_station\_name,

ROUND(AVG(end\_lat), 4) AS end\_lat\_round,

ROUND(AVG(end\_lng), 4) AS end\_lng\_round,

COUNT (\*) AS end\_trips\_per\_station

FROM cyclistic\_member\_riders\_tripdata

GROUP BY end\_station\_name

ORDER BY end\_trips\_per\_station DESC;

SELECT \*

FROM endstation\_latlong\_rounded\_member

ORDER BY end\_trips\_per\_station DESC;

--Same way created casual table

--CREATED TWO TABLES FOR RIDE DATES CASUAL AND MEMBERS TO CHECK MOST RIDEABLE DATES

CREATE TABLE ride\_dates\_casual (

ride\_date DATE,

rides\_per\_date INT);

INSERT INTO ride\_dates\_casual

SELECT CAST(started\_at AS DATE) AS ride\_date,

COUNT(\*) AS ride\_per\_date

FROM cyclistic\_casual\_riders\_tripdata

GROUP BY CAST(started\_at AS DATE)

ORDER BY ride\_date;

CREATE TABLE ride\_dates\_member (

ride\_date DATE,

rides\_per\_date INT);

INSERT INTO ride\_dates\_member

SELECT CAST(started\_at AS DATE) AS ride\_date,

COUNT(\*) AS ride\_per\_date

FROM cyclistic\_member\_riders\_tripdata

GROUP BY CAST(started\_at AS DATE)

ORDER BY ride\_date;

--CREATED TWO TABLES TO COUNT NUMBER OF RIDES WEEKDAYS WITH CASUAL AND MEMBER RIDERS

CREATE TABLE weekday\_ride\_count\_casual (

member\_casual NVARCHAR(MAX),

day\_of\_week\_name NVARCHAR(20),

weekday\_ride\_count INT );

INSERT INTO weekday\_ride\_count\_casual

SELECT

member\_casual,

day\_of\_week\_name,

COUNT (\*) AS weekday\_ride\_count

FROM cyclistic\_trimmed\_tripdata

WHERE member\_casual = 'casual'

GROUP BY member\_casual, day\_of\_week\_name

ORDER BY weekday\_ride\_count DESC;

CREATE TABLE weekday\_ride\_count\_member (

member\_casual NVARCHAR(MAX),

day\_of\_week\_name NVARCHAR(20),

weekday\_ride\_count INT );

INSERT INTO weekday\_ride\_count\_member

SELECT

member\_casual,

day\_of\_week\_name,

COUNT (\*) AS weekday\_ride\_count

FROM cyclistic\_trimmed\_tripdata

WHERE member\_casual = 'member'

GROUP BY member\_casual, day\_of\_week\_name

ORDER BY weekday\_ride\_count DESC;

--CREATED A TABLE FOR RIDE LENGTH MINUTES MEMBER AND CASUAL

CREATE TABLE ride\_length\_member\_casual (

member\_casual NVARCHAR(MAX),

ride\_length\_minutes INT);

INSERT INTO ride\_length\_member\_casual

SELECT

member\_casual,

DATEDIFF (MINUTE, started\_at, ended\_at) AS ride\_length\_minutes

FROM cyclistic\_trimmed\_tripdata;

--CREATE TWO TABLES TO COUNT OF RIDE TYPES THROUGH CASUAL COUNT AND MEMBER COUNT

CREATE TABLE rideable\_type\_count\_casual (

member\_casual NVARCHAR(MAX),

rideable\_type NVARCHAR(MAX),

rideable\_type\_count INT );

INSERT INTO rideable\_type\_count\_casual

SELECT

member\_casual,

rideable\_type,

COUNT (\*) AS rideable\_type\_count

FROM cyclistic\_trimmed\_tripdata

WHERE member\_casual = 'casual'

GROUP BY member\_casual, rideable\_type;

CREATE TABLE rideable\_type\_count\_member (

member\_casual NVARCHAR(MAX),

rideable\_type NVARCHAR(MAX),

rideable\_type\_count INT );

INSERT INTO rideable\_type\_count\_member

SELECT

member\_casual,

rideable\_type,

COUNT (\*) AS rideable\_type\_count

FROM cyclistic\_trimmed\_tripdata

WHERE member\_casual = 'member'

GROUP BY member\_casual, rideable\_type;