```sql
-- CREATE TABLE BIKESHARE2023

CREATE TABLE cyclistc.dbo.bikeshare2023 (
    ride_id nvarchar(50),
    rideable_type nvarchar(50),
    start_station_name nvarchar(max),
    end_station_name nvarchar(max),
    start_lat float,
    start_lng float,
    end_lat float,
    end_lng float,
    member_type nvarchar(50),
    started_day date,
    started_time time(0),
    ended_day date,
    ended_time time(0),
    day_name nvarchar(50),
    trip_duration
)

-- BULK INSERT DATA INTO THE TABLE FROM THE CSV FILE
BULK INSERT cyclistc.dbo.bikeshare2023
FROM 'D:\test\bikeshare2023.csv'
WITH (
  FORMAT = 'CSV',
  FIRSTROW = 2,
  FIELDTERMINATOR = ',',
  ROWTERMINATOR = '\n',
  TABLOCK
);


-- CHECK IF THE DATA HAS BEEN INSERTED CORRECTLY

select * from cyclistc..bikeshare2023

-- FIND OUT MEAN LENGTH OF BIKE RIDE

SELECT AVG(trip_duration) AS mean_ride_length
FROM cyclistc..bikeshare2023

-- CALCULATED THE MAXIMUM RIDE LENGTH

SELECT MAX(trip_duration) AS max_ride_length
FROM cyclistc..bikeshare2023

-- CALCULATE THE MODE OF THE DAY OF THE WEEK
```

```sql
SELECT TOP 1 day_name
FROM cyclistc..bikeshare2023
GROUP BY day_name
ORDER BY COUNT(*) DESC

-- CALCULATE AVERAGE TRIP DURATION BY USER TYPE

SELECT
  member_type,
  AVG(trip_duration) AS avg_ride_length
FROM cyclistc..bikeshare2023
GROUP BY member_type;

-- Calculate the average ride_length for users by day_of_week
SELECT
  DATENAME(WEEKDAY, started_day) AS day_of_week,
  AVG(DATEDIFF(MINUTE, started_time, ended_time)) AS avg_ride_length
FROM cyclistc..bikeshare2023
GROUP BY DATENAME(WEEKDAY, started_day)

-- Calculate the average ride length for subscribers and casual riders as per day
  of the week

  SELECT
  day_name,
  AVG(CASE WHEN member_type = 'Subscriber'
  THEN DATEDIFF(MINUTE, started_time, ended_time)
  END) AS subscriber_avg_ride_length,
  AVG(CASE WHEN member_type = 'Casual'
  THEN DATEDIFF(MINUTE, started_time, ended_time)
  END) AS casual_avg_ride_length
FROM
  cyclistc.dbo.bikeshare2023
GROUP BY
  day_name
ORDER BY
  CASE day_name
    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;

-- Calculate the number of rides for users by day_of_week by adding Count of
  trip_id to Values
```

```sql
SELECT
  DATENAME(WEEKDAY, started_day) AS day_of_week,
  SUM(CASE WHEN member_type = 'Subscriber' THEN 1 ELSE 0 END) AS
    num_subscriber_rides,
  SUM(CASE WHEN member_type = 'Casual' THEN 1 ELSE 0 END) AS num_casual_rides
FROM
  cyclistc.dbo.bikeshare2023
GROUP BY
  DATENAME(WEEKDAY, started_day)

-- RIDE DURATIONS BY SEASONS

SELECT
  CASE
    WHEN MONTH(CAST(started_day AS date)) BETWEEN 2 AND 4 THEN 'Spring'
    WHEN MONTH(CAST(started_day AS date)) BETWEEN 5 AND 8 THEN 'Summer'
    WHEN MONTH(CAST(started_day AS date)) BETWEEN 9 AND 11 THEN 'Fall'
    ELSE 'Winter'
  END AS season,
  member_type,
  AVG(DATEDIFF(minute, started_time, ended_time)) AS avg_ride_duration
FROM
  cyclistc.dbo.bikeshare2023
GROUP BY
  CASE
    WHEN MONTH(CAST(started_day AS date)) BETWEEN 2 AND 4 THEN 'Spring'
    WHEN MONTH(CAST(started_day AS date)) BETWEEN 5 AND 8 THEN 'Summer'
    WHEN MONTH(CAST(started_day AS date)) BETWEEN 9 AND 11 THEN 'Fall'
    ELSE 'Winter'
  END,
  member_type
ORDER BY season, member_type


-- RIDE DURATIONS BY QUARTERS

SELECT
  CASE
    WHEN MONTH(CAST(started_day AS date)) BETWEEN 1 AND 3 THEN 'Q1'
    WHEN MONTH(CAST(started_day AS date)) BETWEEN 4 AND 6 THEN 'Q2'
    WHEN MONTH(CAST(started_day AS date)) BETWEEN 7 AND 9 THEN 'Q3'
    ELSE 'Q4'
  END AS quarter,
  member_type,
  AVG(DATEDIFF(minute, started_time, ended_time)) AS avg_ride_duration
FROM
  cyclistc.dbo.bikeshare2023
GROUP BY
  CASE
```

```
    WHEN MONTH(CAST(started_day AS date)) BETWEEN 1 AND 3 THEN 'Q1'
    WHEN MONTH(CAST(started_day AS date)) BETWEEN 4 AND 6 THEN 'Q2'
    WHEN MONTH(CAST(started_day AS date)) BETWEEN 7 AND 9 THEN 'Q3'
    ELSE 'Q4'
  END,
  member_type
ORDER BY quarter, member_type

-- TOP 10 CITIES BY RIDE DURATION FOR SUBSCRIBERS

SELECT TOP 10
  start_station_name AS city,
  AVG(trip_duration) AS avg_ride_duration
FROM
  cyclistc.dbo.bikeshare2023
WHERE member_type = 'Subscriber'
GROUP BY
  start_station_name
ORDER BY
  AVG(trip_duration) DESC;

-- TOP 10 CITIES BY RIDE DURATION FOR CASUAL MEMBERS

SELECT TOP 10
  start_station_name AS city,
  AVG(trip_duration) AS avg_ride_duration
FROM
  cyclistc.dbo.bikeshare2023
WHERE member_type = 'Casual'
GROUP BY
  start_station_name
ORDER BY
  AVG(trip_duration) DESC;

 -- PREFERRED BIKE TYPE FOR CASUAL MEMBERS

 SELECT
  rideable_type,
  AVG(trip_duration) AS avg_trip_duration
FROM
  cyclistc.dbo.bikeshare2023
WHERE
  member_type = 'Casual'
GROUP BY
  rideable_type
ORDER BY
  AVG(trip_duration) ASC;
```

```sql
-- PERCENTAGE OF SUSBSCRIBERS AND CASUAL MEMBERS

SELECT
  member_type,
  COUNT(*) AS num_members,
  CAST(COUNT(*) * 100.0 / (SELECT COUNT(*) FROM cyclistc.dbo.bikeshare2023) AS
    DECIMAL(5,2)) AS percentage
FROM
  cyclistc.dbo.bikeshare2023
GROUP BY
  member_type


-- TOTAL NUMBER OF RIDES FOR SUBSCRIBERS AND CASUALS FOR EACH DAY

SELECT
  DATENAME(WEEKDAY, started_day) AS day_of_week,
  COUNT(CASE WHEN member_type = 'Subscriber' THEN 1 END) AS num_subscriber_rides,
  COUNT(CASE WHEN member_type = 'Casual' THEN 1 END) AS num_casual_rides
FROM
  cyclistc.dbo.bikeshare2023
WHERE
  CAST(started_time AS DATETIME) BETWEEN CAST('09:00:00' AS DATETIME) AND CAST
    ('21:00:00' AS DATETIME)
GROUP BY
  DATENAME(WEEKDAY, started_day);


-- AVERAGE TRIP DURATION BY CUSTOMER TYPE FOR EACH DAY OF THE WEEK

  SELECT
  DATENAME(WEEKDAY, started_day) AS day_of_week,
  member_type,
  AVG(DATEDIFF(MINUTE, started_time, ended_time)) AS avg_trip_duration
FROM
  cyclistc.dbo.bikeshare2023
WHERE
  CAST(started_time AS DATETIME) BETWEEN CAST('09:00:00' AS DATETIME) AND CAST
    ('21:00:00' AS DATETIME)
GROUP BY
  DATENAME(WEEKDAY, started_day),
  member_type
ORDER BY
  CASE
    WHEN DATENAME(WEEKDAY, started_day) = 'Monday' THEN 1
    WHEN DATENAME(WEEKDAY, started_day) = 'Tuesday' THEN 2
    WHEN DATENAME(WEEKDAY, started_day) = 'Wednesday' THEN 3
    WHEN DATENAME(WEEKDAY, started_day) = 'Thursday' THEN 4
    WHEN DATENAME(WEEKDAY, started_day) = 'Friday' THEN 5
```

```
    WHEN DATENAME(WEEKDAY, started_day) = 'Saturday' THEN 6
    WHEN DATENAME(WEEKDAY, started_day) = 'Sunday' THEN 7
  END ASC,
  member_type ASC
```