# Checking for unique id for each dataset

SELECT COUNT (DISTINCT id)

FROM [dbo].[dailyActivity\_merged]

# 33

SELECT COUNT (DISTINCT id)

FROM [dbo].[dailyCalories\_merged]

# 33

SELECT COUNT (DISTINCT id)

FROM [dbo].[dailyIntensities\_merged]

# 33

SELECT COUNT (DISTINCT id)

FROM [dbo].[dailySteps\_merged]

# 33

SELECT COUNT (DISTINCT id)

FROM [dbo].[sleepDay\_merged]

# 24

SELECT COUNT (DISTINCT id)

FROM [dbo].[weightLogInfo\_merged]

# 8

#To check calories, match up with Activity table

SELECT COUNT (DISTINCT Calories)

FROM [dbo].[dailyCalories\_merged]

#734

SELECT COUNT (DISTINCT Calories)

FROM [dbo].[dailyActivity\_merged]

#734

#To check steps, match up with Activity table

SELECT COUNT (DISTINCT TotalSteps)

FROM [dbo].[dailyActivity\_merged]

# 842

SELECT COUNT (DISTINCT StepTotal)

FROM [dbo].[dailySteps\_merged]

# 842

#To check sedentarytime, match up with Activity table

SELECT COUNT (DISTINCT SedentaryMinutes)

FROM [dbo].[dailyActivity\_merged]

# 549

SELECT COUNT (DISTINCT SedentaryMinutes)

FROM [dbo].[dailyIntensities\_merged]

#549

.

#Activity date period

SELECT MIN(ActivityDate) AS StartDate,

MAX(ActivityDate) AS EndDate

FROM [dbo].[dailyActivity\_merged]

#StartDate - 2016-04-12

#EndDate - 2016-05-12

SELECT MIN(SleepDay) AS StartDate, MAX(SleepDay) AS EndDate

FROM [dbo].[sleepDay\_merged]

#StartDate - 2016-04-12

#EndDate - 2016-05-12

SELECT MIN(Date) AS StartDate, MAX(Date) AS EndDate

FROM [dbo].[weightLogInfo\_merged]

#StartDate - 2016-04-12

#EndDate - 2016-05-12

#To Check ID length for dailyActivity, sleepDay, weightLog

SELECT DISTINCT( (LEN(Id)))

FROM [dbo].[dailyActivity\_merged]

SELECT DISTINCT ( (LEN(Id)))

FROM [dbo].[sleepDay\_merged]

SELECT DISTINCT ( (LEN(Id)))

FROM [dbo].[weightLogInfo\_merged]

# Find Duplicates

SELECT

Id,

ActivityDate,

Count(\*)

FROM [dbo].[dailyActivity\_merged]

GROUP BY

Id, ActivityDate

Having count(\*)>1

# No duplicates found

SELECT

Id,

SleepDay,

Count(\*)

FROM [dbo].[sleepDay\_merged]

GROUP BY

Id, SleepDay

Having count(\*)>1

# No duplicates found

SELECT

Id,

Date,

Count(\*)

FROM [dbo].[weightLogInfo\_merged]

GROUP BY

Id, Date

Having count(\*)>1

# No duplicates found

# Checking Null data

SELECT \*

FROM [dbo].[dailyActivity\_merged]

WHERE Id IS NULL

# No Null Values found

SELECT \*

FROM [dbo].[dailyActivity\_merged]

WHERE Id IS NULL

# No Null Values found

SELECT \*

FROM [dbo].[sleepDay\_merged]

WHERE Id IS NULL

# No Null Values found

SELECT \*

FROM [dbo].[weightLogInfo\_merged]

WHERE Id IS NULL

# No Null Values found

SELECT \*

FROM [dbo].[weightLogInfo\_merged]

WHERE Fat IS NULL

# 65 found

# check the user active minutes by day of the week(including sedimentary time)

SELECT DATENAME(w, ActivityDate) AS week\_Day, AVG(VeryActiveMinutes) AS AvgVeryActiveMinutes, AVG(FairlyActiveMinutes) AS AvgFairlyActiveMinutes, AVG(LightlyActiveMinutes) AS AvgLightlyActiveMinutes, AVG(SedentaryMinutes) AS AvgSedentaryMinutes

FROM [dbo].[dailyActivity\_merged]

WHERE VeryActiveMinutes IS NOT NULL

AND FairlyActiveMinutes IS NOT NULL

AND LightlyActiveMinutes IS NOT NULL

AND SedentaryMinutes IS NOT NULL

GROUP BY DATENAME(w, ActivityDate)

**# Checking user lifestyle from steps per day**

SELECT

CASE

WHEN TotalSteps < 5000 THEN 'Sedentary'

WHEN TotalSteps BETWEEN 5000 AND 7499 THEN 'Low Active'

WHEN TotalSteps BETWEEN 7500 AND 9999 THEN 'Somewhat Active'

WHEN TotalSteps >= 10000 AND TotalSteps <= 12499 THEN 'Active'

WHEN TotalSteps > 12500 THEN 'Highly Active'

END AS ActivityLevel,

AVG(TotalSteps) AS AverageSteps

FROM [dbo].[dailyActivity\_merged]

GROUP BY

CASE

WHEN TotalSteps < 5000 THEN 'Sedentary'

WHEN TotalSteps BETWEEN 5000 AND 7499 THEN 'Low Active'

WHEN TotalSteps BETWEEN 7500 AND 9999 THEN 'Somewhat Active'

WHEN TotalSteps >= 10000 AND TotalSteps <= 12499 THEN 'Active'

WHEN TotalSteps > 12500 THEN 'Highly Active'

END

**# AVG distance, AVG step, AVG calorie as per days of the week**

SELECT DATENAME(w, ActivityDate) AS week\_Day, AVG(TotalSteps) AS AvgTotalSteps, ROUND (AVG(TotalDistance),2) AS AvgTotalDistance, AVG(Calories) AS AvgCalories

FROM [dbo].[dailyActivity\_merged]

WHERE TotalSteps IS NOT NULL

AND TotalDistance IS NOT NULL

AND Calories IS NOT NULL

GROUP BY DATENAME(w, ActivityDate)

**#** **Checking correlation between Avg steps and Avg Calories**

SELECT AVG(TotalSteps) as AvgSteps, AVG(Calories) AS AvgCalories, Id

FROM [dbo].[dailyActivity\_merged]

GROUP BY Id