#SQL queries for the process phase

**#Phase 3: Process**

-- cleaning and filtering the data

-- Remove columns from each table that are irrelevant

use bellabeat\_data;

select \* from dailyactivity\_merged; /\*this is the table name \*/

alter table dailyactivity\_merged

drop column LoggedActivitiesDistance,

drop column VeryActiveDistance,

drop column ModeratelyActiveDistance,

drop column LightActiveDistance,

drop column SedentaryActiveDistance;

/\*check the datatype and names of all columns in each table \*/

#Table 1 - dailyactivity\_merged

#I encountered error code 1175, so I added the safe updates statement

SET SQL\_SAFE\_UPDATES = 0;

describe dailyactivity\_merged;

/\*ActivityDate column was 'text' datatype. So changed it to 'date' and renamed it to 'date' \*/

update dailyactivity\_merged

set ActivityDate = str\_to\_date(ActivityDate, '%m/%d/%Y');

alter table dailyactivity\_merged

change column ActivityDate date date;

#Table 2 - sleepday\_merged

SET SQL\_SAFE\_UPDATES = 0;

describe sleepday\_merged;

/\* SleepDay column was 'text' datatype, so changed it to 'datetime' and renamed it to 'date\_time' \*/

update sleepday\_merged

set SleepDay = str\_to\_date(SleepDay, '%m/%d/%Y %h:%i:%s %p');

alter table sleepday\_merged

change column SleepDay date\_time datetime;

#Table 3 - hourlycalories\_merged

describe hourlycalories\_merged;

SET SQL\_SAFE\_UPDATES = 0;

/\* ActivityHour column was 'text' datatype, so I changed it to 'datetime' and renamed it to 'date\_time' \*/

update hourlycalories\_merged

set ActivityHour = str\_to\_date(ActivityHour, '%m/%d/%Y %h:%i:%s %p');

alter table hourlycalories\_merged

change column ActivityHour date\_time datetime;

#Table 4 - hourlysteps\_merged

describe hourlysteps\_merged

SET SQL\_SAFE\_UPDATES = 0;

/\* ActivityHour column was 'text' datatype so I changed it to 'datetime' and renamed it to 'date\_time' \*/

update hourlysteps\_merged

set ActivityHour = str\_to\_date(ActivityHour, '%Y-%m-%d %H:%i:%s');

alter table hourlysteps\_merged

change column ActivityHour date\_time datetime;

#Table 5 - hourlyintensities\_merged

describe hourlyintensities\_merged;

/\* ActivityHour column was 'text' datatype, so I changed it to 'datetime' and renamed it to 'date\_time' \*/

update hourlyintensities\_merged

set ActivityHour = str\_to\_date(ActivityHour, '%m/%d/%Y %h:%i:%s %p');

alter table hourlyintensities\_merged

change column ActivityHour date\_time datetime;

#1 Inspecting 'dailyactivity\_merged'

-- select \* from dailyactivity\_merged;

-- select count(distinct id) as total\_users from dailyactivity\_merged; -- 33 users

-- select count(distinct date) as total\_days from dailyactivity\_merged; -- 31 days

#2 Inspecting 'sleepday\_merged'

select \* from sleepday\_merged;

select count(distinct id) as total\_users from sleepday\_merged; -- 24 users

select count(distinct date\_time) as total\_days from sleepday\_merged; -- 31 days

#3 Inspecting 'hourlycalories\_merged'

-- select \* from hourlycalories\_merged;

-- select count(\*) from hourlycalories\_merged; -- 22099 total records

-- select count(distinct id) as total\_users from hourlycalories\_merged; -- 33 users

select count(distinct (date(date\_time))) as total\_days,

count(distinct (time(date\_time))) as total\_hours

from hourlycalories\_merged; -- 31 days (containing 24 hours)

#4 Inspecting 'hourlyintensities\_merged'

-- select \* from hourlyintensities\_merged;

-- select count(\*) from hourlyintensities\_merged; -- 22099 total records

-- select count(distinct id) as total\_users from hourlyintensities\_merged; -- 33 users

select count(distinct (date(date\_time))) as total\_days,

count(distinct (time(date\_time))) as total\_hours

from hourlyintensities\_merged; -- 31 days (containing 24 hours)

#5 Inspecting 'hourlysteps\_merged'

-- select \* from hourlysteps\_merged;

-- select count(\*) from hourlysteps\_merged; -- 22099 total records

-- select count(distinct id) as total\_users from hourlysteps\_merged; -- 33 users

select count(distinct (date(datetime))) as total\_days,

count(distinct (time(datetime))) as total\_hours from hourlysteps\_merged; -- 31 days (containing 24 hours)

/\* finding duplicates in dailyactivity\_merged table\*/

select id, date, TotalSteps, TotalDistance, TrackerDistance, Calories, count(\*)

from dailyactivity\_merged

group by id, date, TotalSteps, TotalDistance, TrackerDistance, Calories

having count(\*) > 1; -- 0 Duplicates as the result was empty

/\* finding null/missing values in dailyactivity\_merged table\*/

SELECT

SUM(CASE WHEN id is null or id = 0 then 1 else 0 end) as missing\_id,

SUM(CASE WHEN date IS NULL THEN 1 ELSE 0 END) AS Missing\_ActivityDate,

SUM(CASE WHEN TotalSteps = 0 THEN 1 ELSE 0 END) AS Zero\_TotalSteps,

SUM(CASE WHEN TotalDistance = 0 THEN 1 ELSE 0 END) AS Zero\_TotalDistance,

SUM(CASE WHEN TrackerDistance = 0 THEN 1 ELSE 0 END) AS Zero\_TrackerDistance,

SUM(CASE WHEN VeryActiveMinutes = 0 THEN 1 ELSE 0 END) AS Zero\_VeryActiveMinutes,

SUM(CASE WHEN FairlyActiveMinutes = 0 THEN 1 ELSE 0 END) AS Zero\_FairlyActiveMinutes,

SUM(CASE WHEN LightlyActiveMinutes = 0 THEN 1 ELSE 0 END) AS Zero\_LightlyActiveMinutes,

SUM(CASE WHEN SedentaryMinutes = 0 THEN 1 ELSE 0 END) AS Zero\_SedentaryMinutes,

SUM(CASE WHEN Calories = 0 THEN 1 ELSE 0 END) AS Zero\_Calories

from dailyactivity\_merged;

/\* ID and date column as expected didn't have any missing values, but total\_steps column had 77 values as 0 which can't be right because if a person has used the device, he must have walked atleast 1 step that day. So, I removed these entries that had 0 step count. \*/

delete from dailyactivity\_merged where TotalSteps = 0;

/\* if totalsteps can't be 0 then calories also can't be zero \*/

select count(\*) from dailyactivity\_merged

where Calories = 0; -- no entries with 0 calories

/\* Checking if all the activity minutes add upto 24 hours or 1440 minutes, if not then those values are invalid \*/

select \* from

(

select VeryActiveMinutes, FairlyActiveMinutes, LightlyActiveMinutes, SedentaryMinutes,

(VeryActiveMinutes + FairlyActiveMinutes + LightlyActiveMinutes + SedentaryMinutes) as total\_minutes from dailyactivity\_merged

)

dailyactivity\_merged where total\_minutes > 1440; -- No invalid values as the output was empty

/\* finding duplicates in 'sleepday\_merged' table \*/

select \*, count(\*) as duplicates from sleepday\_merged

group by id, date\_time, TotalSleepRecords, TotalMinutesAsleep, TotalTimeInBed

having count(\*) > 1; -- 3 Duplicates

/\* Removing duplicates \*/

/\* since ids are also duplicated I added a new column called 'row\_num' to give each row a unique identifier, which made it easy to filter out and remove duplicates \*/

alter table sleepday\_merged

add column row\_num int auto\_increment, add primary key (row\_num);

/\* deleting duplicates using 'row\_num' column \*/

delete from sleepday\_merged

where row\_num in (select \* from

(select max(row\_num) as rn

from sleepday\_merged

group by id, date\_time, TotalSleepRecords, TotalMinutesAsleep, TotalTimeInBed

having count(\*) > 1

) sleepday\_merged);

/\* finding invalid values (values that are greater than 1440 mintues or 24 hours) \*/

select \* from sleepday\_merged

where TotalMinutesAsleep > 1440 or TotalTimeInBed > 1440; -- No invalid values found

/\* Adding a new column 'date' from 'date\_time' \*/

alter table sleepday\_merged;

-- add column date date after date\_time;

update sleepday\_merged

set date = date(date\_time);

/\* finding duplicates in 'hourlycalories\_merged' table\*/

select \*, count(\*) as duplicates from hourlycalories\_merged

group by id, date\_time, Calories having count(\*) > 1; -- No duplicates were found

/\* identifying missing/null values \*/

select sum(case when id is null or id = 0 then 1 else 0 end) as missing\_ids,

sum(case when date\_time is null or date\_time = 0 then 1 else 0 end) as missing\_dates,

sum(case when Calories is null or Calories = 0 then 1 else 0 end) as missing\_calories

from hourlycalories\_merged; -- No missing or null values were found

/\* finding duplicates in ‘hourlyintensities\_merged' table\*/

select \*, count(\*) as duplicates from hourlyintensities\_merged

group by id, date\_time, TotalIntensity, AverageIntensity

having count(\*) > 1; -- No duplicates were found

/\* identifying missing/null values \*/

/\*select \* from hourlyintensities\_merged

where id is null or id = 0

or date\_time is null or date\_time = 0 or

totalintensity is null or totalintensity = 0 or

averageintensity is null or averageintensity = 0; \*/

select sum(case when id is null or id = 0 then 1 else 0 end) as missing\_ids,

sum(case when date\_time is null or date\_time = 0 then 1 else 0 end) as missing\_dates,

sum(case when TotalIntensity is null or TotalIntensity = 0 then 1 else 0 end) as missing\_intensities,

sum(case when averageintensity is null or averageintensity = 0 then 1 else 0 end) as missing\_avg\_inensities

from hourlyintensities\_merged; -- '9097' missing values in both 'Totalintensity and AverageIntensity' columns

/\* finding duplicates in 'hourlysteps\_merged' table \*/

select \*, count(\*) as duplicates

from hourlysteps\_merged

group by id, datetime, StepTotal

having count(\*) > 1; -- No duplicates

/\* identifying missing/null values \*/

select sum(case when id is null or id = 0 then 1 else 0 end) as missing\_ids,

sum(case when datetime is null or datetime = 0 then 1 else 0 end) as missing\_dates,

sum(case when steptotal is null or steptotal = 0 then 1 else 0 end) as missing\_steptotal

from hourlysteps\_merged; -- '9297' missing values in 'Steptotal' column

/\*The next step was to transform the data and do further analysis \*/

/\* I combined data from tables containing data related to daily activity (dailyactivity\_merged, dailysleep\_merged) into a new table called 'daily\_activity\_sleep' and tables containing hourly data (hourlycalories\_merged, hourlyintensities\_merged, hourlysteps\_merged) into a new table called 'hourly\_activity' \*/

-- combining tables 'dailyactivity\_merged' and 'dailysleep\_merged'

create table daily\_activity\_sleep

select tbl1.\*, tbl2.TotalSleepRecords, TotalMinutesAsleep, TotalTimeInBed

from dailyactivity\_tbl tbl1

join dailysleep\_tbl tbl2 using(id, date);

/\* the above table had data for '24 users' \*/

/\*Next, I created one table of hourly\_activity by merging all the hourly data tables. For that first I had to rename a column name datetime in the hourly steps table as date\_time to match all the column names with the other hourly tables.\*/

-- combining 'hourlycalories\_merged' , 'hourlyintensities\_merged' and 'hourlysteps\_merged'

create table hourly\_activity

select tbl1.\*, tbl2.TotalIntensity, tbl2.AverageIntensity, tbl3.StepTotal

from hourlycalories\_merged tbl1

join hourlyintensities\_merged tbl2 using (id, date\_time)

join hourlysteps\_merged tbl3 using (id, date\_time);

/\* Adding a new column 'day' to 'daily\_activity\_sleep', 'dailyactivity\_merged', 'hourly\_activity' \*/

#1 adding day column to 'daily\_activity\_sleep'

SET SQL\_SAFE\_UPDATES = 0;

alter table daily\_activity\_sleep

add column day varchar(10) after date;

update daily\_activity\_sleep

set day = dayname(date);

#2 adding day column to 'dailyactivity\_merged'

SET SQL\_SAFE\_UPDATES = 0;

alter table dailyactivity\_merged

add column day varchar(10) after date;

update dailyactivity\_merged

set day = dayname(date);

#3 adding day column to 'hourly\_activity'

SET SQL\_SAFE\_UPDATES = 0;

alter table hourly\_activity

add column day varchar(10) after date\_time;

update hourly\_activity

set day = dayname(date\_time);

/\* Adding a new column 'total\_active\_minutes' in these 2 tables \*/

#1 adding 'total\_active\_minutes' column to 'daily\_activity\_sleep'

/\*SET SQL\_SAFE\_UPDATES = 0;

alter table daily\_activity\_sleep

add column total\_active\_minutes int after totalsteps; \*/

UPDATE daily\_activity\_sleep

SET total\_active\_minutes = veryactiveminutes + fairlyactiveminutes + lightlyactiveminutes;

#2 adding 'total\_active\_minutes' column to 'dailyactivity\_merged'

/\*SET SQL\_SAFE\_UPDATES = 0;

alter table dailyactivity\_merged

add column total\_active\_minutes int after totalsteps; \*/

UPDATE dailyactivity\_merged

SET total\_active\_minutes = veryactiveminutes + fairlyactiveminutes + lightlyactiveminutes;

/\* Adding a new column 'total\_active\_minutes' in 'daily\_activity\_sleep'\*/

/\*SET SQL\_SAFE\_UPDATES = 0;

alter table daily\_activity\_sleep

add column minutes\_awake int ; \*/

UPDATE daily\_activity\_sleep

SET minutes\_awake = totaltimeinbed - totalminutesasleep;

/\* Removing distance related columns, as they don't much info about usage preferences

or health profile, because for that we have totalsteps and activity minutes data \*/

#1 removing distance columns from 'daily\_activity\_sleep'

SET SQL\_SAFE\_UPDATES = 0;

alter table daily\_activity\_sleep

drop column TotalDistance,

drop column TrackerDistance;

#2 removing distance columns from 'dailyactivity\_merged'

SET SQL\_SAFE\_UPDATES = 0;

alter table dailyactivity\_merged

drop column TotalDistance,

drop column TrackerDistance;

/\* FINAL COMPLETE TABLES \*/

select \* from daily\_activity\_sleep;

select \* from dailyactivity\_merged;

select \* from hourly\_activity;

-- ANALYSIS

-- Device usage Analysis

#1 Total users of each feature

select count(distinct id) as sleeptracker\_users,

count(distinct id) as activitytracker\_users

from daily\_activity\_sleep; -- 24 users

select count(distinct id) as activitytracker\_users

from dailyactivity\_merged; -- 33 users

#2 Device usage level

select id, days\_used,

case when days\_used between 1 and 10 then 'Seldom'

when days\_used between 10 and 25 then 'Very Often'

when days\_used > 25 then 'Regular'

else 'Wrong'

end as usage\_type

from (

select id, count(date) as days\_used

from dailyactivity\_merged

group by id

order by days\_used desc

) dailyactivity\_merged

group by id, days\_used;

3#User average active minutes during the week.

For this, I just used Tableau Public and used the dailyactivity\_merged table to draw this dual chart.

#4 sedentary vs active during the month

For this, I just used Tableau Public and used the dailyactivity\_merged table to draw this dual chart.

#5 usage rate by day of the week

select day, day\_num, round(avg(total\_users),2) as avg\_users

from (

select date, weekday(date)+1 as day\_num, day, count(\*) as total\_users

from dailyactivity\_merged

group by date, day

order by date, day\_num

)dailyactivity\_merged

group by day, day\_num

order by day\_num;

#6 Average steps taken by hour and days of the week

select (row\_number() over (partition by day order by hour))-1 as hour\_num,

day, day\_num, hour, round(avg(total\_steps),2) as avg\_steps

from (

select id, day, weekday(date\_time)+1 as day\_num, time(date\_time) as hour, sum(StepTotal) as total\_steps

from hourly\_activity

group by id, day, hour, day\_num

order by day\_num

)hourly\_activity

group by day, hour, day\_num

order by day\_num, hour\_num;

#7 Average time in bed by day

select day, avg(TotalMinutesAsleep), avg(TotalTimeInBed), avg(minutes\_awake)

from daily\_activity\_sleep

group by day;

#Some extra queries that I used for drawing graphs.

### **#Classifying Users with respect to their Average\_Daily\_Steps**

#Average by users

#Step 1: For that purpose, I first created a table to use it in the next steps.

Create Table AverageByUsers

SELECT id,

round(avg(TotalSteps), 2) AS Avg\_Daily\_Steps,

round(avg(TotalMinutesAsleep), 0) AS Avg\_Sleep,

round(avg(Calories), 0) AS Avg\_Calories

FROM

daily\_activity\_sleep

GROUP BY id;

#Step 2:  I created another table to use it in the next step by using the table which i created in step 1.

CREATE TABLE user\_classification\_by\_avg\_daily\_steps

SELECT id, AVG\_Daily\_Steps,

(CASE

WHEN AVG\_Daily\_Steps < 5000 THEN "Sedentary"

WHEN AVG\_Daily\_Steps >= 5000 AND AVG\_Daily\_Steps < 7499 THEN "Lighlity Active"

WHEN AVG\_Daily\_Steps >= 7500 AND AVG\_Daily\_Steps < 9999 THEN "Fairly Active"

WHEN AVG\_Daily\_Steps >= 10000 THEN "Very Active"

END) AS User\_Type

FROM AverageByUsers;