**Points to note:**

-- 1. Check that all rows of csv files have been imported to MySQL workbench

Prior to importing these files successfully, I faced some importation issues (all rows of datasets were not being imported) with the import wizard’s chosen datatype INT for column, ‘Id’ of some tables. I changed the datatype INT to BIGINT for columns ‘Id’ this allowed for larger values for this column to be imported.

--2. Check to see how many unique participants there are per table

There are limited numbers of participants in each table. 33 in dailyactivity, hourlyintensities, hourlysteps; 24 in sleepday and 8 in weightloginfo. This data isn’t sufficient to make confident decisions based on a whole population.

-- 3. Check to see if data types for each column are appropriate and modifying where necessary

To stay consistent, I changed most data types for each column of each table, ensuring that any data which is present within two or more tables had the exact same data type. I could not change all datatypes at once so had to execute separate statements for each. I would otherwise have to create a new table as suggested by other users. I used %r to change the times from AM/PM to 24-hour format. I dropped columns with Date and Time after copying data into two separate columns within the tables.

-- 6. Check for duplicate rows

I created a primary key which automatically incremented from 1 for all rows of sleepday. This allowed for a unique identifier as there are many duplicate fields (as one Id would have multiple observations) to easily identify rows of duplicates to delete them from the table. 3 row duplicates were found. After removing duplicates, the PID column was dropped.

-- 7. Check for NULL or missing values

weightloginfo table was the only table with missing values in 65/67 rows within the column ‘Fat’. I will consider this when carrying out analysis.

**SQL syntax and comments**

**-- 1. Check all rows of .CSV files have been imported to MySQL workbench**

**SELECT COUNT(\*) FROM dailyactivity; -- All 940 rows were imported successfully**

**SELECT COUNT(\*) FROM hourlyintensities; -- All 22099 rows were imported successfully**

**SELECT COUNT(\*) FROM hourlysteps; -- All 22099 rows were imported successfully**

**SELECT COUNT(\*) FROM sleepday; -- All 413 rows were imported successfully**

**SELECT COUNT(\*) FROM weightloginfo; -- All 67 rows were imported successfully**

**-- 2. Check how many unique participants (Id) per table**

**SELECT COUNT(distinct Id) AS totalparticipants**

**FROM dailyactivity; -- dailyactivity has 33 participants**

**SELECT COUNT(distinct Id) AS totalparticipants**

**FROM hourlyintensities; -- hourlyintensities has 33 participants**

**SELECT COUNT(distinct Id) AS totalparticipants**

**FROM hourlysteps; -- hourlysteps has 33 participants**

**SELECT COUNT(distinct Id) AS totalparticipants**

**FROM sleepday; -- sleepday has 24 participants**

**SELECT COUNT(distinct Id) AS totalparticipants**

**FROM weightloginfo; -- weightloginfo has 8 participants**

**-- 3. Check datatypes for each column are appropriate and modifying where necessary**

**DESCRIBE dailyactivity;**

**-- Default date format in MySQL is YYYY-MM-DD, I will change the format to conform to this.**

**-- This will change format of the date from mm/dd/yyyy to yyyy/mm/dd before I try and change datatype text to date.**

**UPDATE dailyactivity**

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

**-- Can now modify text to date and do a check to see if it has applied.**

**ALTER TABLE dailyactivity**

**MODIFY ActivityDate date;**

**DESCRIBE dailyactivity;**

**SELECT \* FROM dailyactivity;**

**-- Change the remaining columns of dailyactivity to their appropriate datatypes.**

**-- I changed the following from double to decimal(1.d.p)**

**ALTER TABLE dailyactivity MODIFY TotalDistance decimal(10,1);**

**ALTER TABLE dailyactivity MODIFY TrackerDistance decimal(10,1);**

**ALTER TABLE dailyactivity MODIFY LoggedActivitiesDistance decimal(10,1);**

**ALTER TABLE dailyactivity MODIFY VeryActiveDistance decimal(10,1);**

**ALTER TABLE dailyactivity MODIFY ModeratelyActiveDistance decimal(10,1);**

**ALTER TABLE dailyactivity MODIFY LightActiveDistance decimal(10,1);**

**ALTER TABLE dailyactivity MODIFY SedentaryActiveDistance decimal(10,1);**

**ALTER TABLE dailyactivity MODIFY VeryActiveDistance decimal(10,1);**

**-- Change datatypes for hourlyintensities**

**ALTER TABLE hourlyintensities MODIFY TotalIntensity decimal(10,1);**

**ALTER TABLE hourlyintensities MODIFY AverageIntensity decimal(10,1);**

**SELECT \* FROM hourlyintensities;**

**-- Change ActivityHour format to conform to sql format and time to be in 24 format using %r(AM/PM)**

**-- Convert Date datatype to datetime for ActivityHour**

**UPDATE hourlyintensities**

**SET ActivityHour=str\_to\_date(ActivityHour, '%m/%d/%Y %r');**

**ALTER TABLE hourlyintensities**

**MODIFY ActivityHour datetime;**

**-- Create columns ActiveDate and ActiveHour in hourlyintensities table and what order**

**ALTER TABLE hourlyintensities**

**ADD COLUMN ActiveDate date AFTER Id;**

**ALTER TABLE hourlyintensities**

**ADD COLUMN ActiveHour time AFTER ActiveDate;**

**-- Split date and time from ActivityHour column into ActiveDate and ActiveHour.**

**UPDATE hourlyintensities**

**SET ActiveDate=SUBSTRING\_INDEX(ActivityHour, ' ', 1), ActiveHour=SUBSTRING\_INDEX(ActivityHour, ' ', -1);**

**-- DROP COLUMN ActivityHour as we have new columns with this information**

**ALTER TABLE hourlyintensities**

**DROP COLUMN ActivityHour;**

**-- Change datatypes for hourlysteps by same process as hourlyintensities to split up ActivityHour column into ActiveDate and ActiveHour**

**ALTER TABLE hourlysteps MODIFY Id bigint;**

**UPDATE hourlysteps**

**SET ActivityHour=str\_to\_date(ActivityHour, '%m/%d/%Y %r');**

**ALTER TABLE hourlysteps**

**MODIFY ActivityHour datetime;**

**ALTER TABLE hourlysteps**

**ADD COLUMN ActiveDate date AFTER Id;**

**ALTER TABLE hourlysteps**

**ADD COLUMN ActiveHour time AFTER ActiveDate;**

**UPDATE hourlysteps**

**SET ActiveDate=SUBSTRING\_INDEX(ActivityHour, ' ', 1), ActiveHour=SUBSTRING\_INDEX(ActivityHour, ' ', -1);**

**ALTER TABLE hourlysteps**

**DROP COLUMN ActivityHour;**

**-- Change datatypes for sleepday by same process as hourlyintensities/hourlysteps to split SleepDay into SleepDate and SleepHour**

**UPDATE sleepday**

**SET SleepDay=str\_to\_date(SleepDay, '%m/%d/%Y %r');**

**ALTER TABLE sleepday**

**MODIFY SleepDay datetime;**

**ALTER TABLE sleepday**

**ADD COLUMN SleepDate date AFTER Id;**

**ALTER TABLE sleepday**

**ADD COLUMN SleepHour time AFTER SleepDate;**

**UPDATE sleepday**

**SET SleepDate=SUBSTRING\_INDEX(SleepDay, ' ', 1), SleepHour=SUBSTRING\_INDEX(SleepDay, ' ', -1);**

**ALTER TABLE sleepday**

**DROP COLUMN SleepDay;**

**-- Change datatypes for weightloginfo by same process as hourlyintensities/hourlysteps and sleepday to split Date into WLIDate and WLITime**

**ALTER TABLE weightloginfo MODIFY WeightKg decimal(20,1);**

**ALTER TABLE weightloginfo MODIFY WeightPounds decimal(20,1);**

**ALTER TABLE weightloginfo MODIFY BMI decimal(20,1);**

**UPDATE weightloginfo**

**SET Date=str\_to\_date(Date, '%m/%d/%Y %r');**

**ALTER TABLE weightloginfo**

**MODIFY Date datetime;**

**ALTER TABLE weightloginfo**

**ADD COLUMN WLIDate date AFTER Id;**

**ALTER TABLE weightloginfo**

**ADD COLUMN WLITime time AFTER WLIDate;**

**UPDATE weightloginfo**

**SET WLIDate=SUBSTRING\_INDEX(Date, ' ', 1), WLITime=SUBSTRING\_INDEX(Date, ' ', -1);**

**ALTER TABLE weightloginfo**

**DROP COLUMN Date;**

**-- Check that TotalTimeInBed values are bigger >= TotalMinutesAsleep**

**-- I will add another column when it comes to analysis**

**SELECT COUNT(\*) FROM sleepday**

**WHERE TotalTimeInBed >= TotalMinutesAsleep; -- All records agree with this statement**

**-- 4. Check duration of data for all tables**

**-- Data recorded is across 30 days (2016-05-12 and 2016-04-12) within all tables**

**SELECT min(ActivityDate), max(ActivityDate) FROM dailyactivity; -- Shows min date is 2016-04-12 and max date 2016-05-12**

**SELECT datediff('2016-05-12', '2016-04-12'); -- Shows there is 30 days of data between those two dates**

**-- 5. Check Id characters are appropriate character lengths**

**SELECT**

**Length(Id) AS length\_of\_id**

**FROM dailyactivity; -- This shows that Id character length should be 10 for all tables**

**SELECT Id**

**FROM dailyactivity**

**WHERE LENGTH(Id) <>10; -- All Id characters are of 10 characters**

**SELECT Id**

**FROM hourlyintensities**

**WHERE LENGTH(Id) <>10; -- All Id characters are of 10 characters**

**SELECT Id**

**FROM hourlysteps**

**WHERE LENGTH(Id) <>10; -- All Id characters are of 10 characters**

**SELECT Id**

**FROM sleepday**

**WHERE LENGTH(Id) <>10; -- All Id characters are of 10 characters**

**SELECT Id**

**FROM weightloginfo**

**WHERE LENGTH(Id) <>10; -- All Id characters are of 10 characters**

**-- For weightloginfo, there is a column with LogId which I will check for consistent character length**

**SELECT**

**Length(LogId) AS lengthoflogid**

**FROM weightloginfo; -- This shows that there should be 13 characters per LogId**

**SELECT LogId**

**FROM weightloginfo**

**WHERE LENGTH(LogId) <>13; -- All Id characters are of 13 characters**

**-- 6. Check for duplicate rows**

**-- in dailyactivity**

**SELECT**

**Id,**

**ActivityDate,**

**TotalSteps,**

**TotalDistance,**

**TrackerDistance,**

**LoggedActivitiesDistance,**

**VeryActiveDistance,**

**SedentaryMinutes,**

**ModeratelyActiveDistance,**

**LightActiveDistance,**

**SedentaryActiveDistance,**

**VeryActiveMinutes,**

**FairlyActiveMinutes,**

**LightlyActiveMinutes,**

**SedentaryMinutes,**

**Calories,**

**COUNT(\*)**

**FROM dailyactivity**

**GROUP BY**

**Id,**

**ActivityDate,**

**TotalSteps,**

**TotalDistance,**

**TrackerDistance,**

**LoggedActivitiesDistance,**

**VeryActiveDistance,**

**SedentaryMinutes,**

**ModeratelyActiveDistance,**

**LightActiveDistance,**

**SedentaryActiveDistance,**

**VeryActiveMinutes,**

**FairlyActiveMinutes,**

**LightlyActiveMinutes,**

**SedentaryMinutes,**

**Calories**

**HAVING COUNT(\*) >1; -- there are no duplicate rows within dailyactivity**

**-- Check for duplicate rows in hourlyintensities**

**SELECT Id, ActiveDate, ActiveHour, TotalIntensity, AverageIntensity, COUNT(\*)**

**FROM hourlyintensities**

**GROUP BY Id, ActiveDate, ActiveHour, TotalIntensity, AverageIntensity**

**HAVING COUNT(\*) >1; -- there are no duplicate rows within hourlyintensities**

**-- Check for duplicate rows in hourlysteps**

**SELECT Id, ActiveDate, ActiveHour, StepTotal, COUNT(\*)**

**FROM hourlysteps**

**GROUP BY Id, ActiveDate, ActiveHour, StepTotal**

**HAVING COUNT(\*) >1; -- there are no duplicate rows within hourlysteps**

**-- Check for duplicate rows in sleepday**

**SELECT Id, SleepDate, SleepHour, TotalSleepRecords, TotalMinutesAsleep, TotalTimeInBed, COUNT(\*)**

**FROM sleepday**

**GROUP BY Id, SleepDate, SleepHour, TotalSleepRecords, TotalMinutesAsleep, TotalTimeInBed**

**HAVING COUNT(\*) >1; -- there are 3 rows with duplicates**

**ALTER TABLE sleepday**

**ADD PID int AUTO\_INCREMENT PRIMARY KEY; -- Auto increment primary key gives each row a unique identifier**

**SELECT \* FROM sleepday**

**WHERE Id = '4388161847' AND TotalTimeInBed = 495; -- this confirms that the rows are duplicate**

**SELECT \* FROM sleepday**

**WHERE Id = '4702921684' AND TotalTimeInBed = 543; -- this confirms that the rows are duplicate**

**SELECT \* FROM sleepday**

**WHERE Id = '8378563200' AND TotalTimeInBed = 402; -- this confirms that the rows are duplicate**

**-- Using the unique identifier for each duplicate row, I removed the duplicate**

**DELETE FROM sleepday WHERE PID IN (19, 249, 380);**

**-- Drop the column used to remove duplicates**

**ALTER TABLE sleepday**

**DROP COLUMN PID;**

**SELECT COUNT(\*) FROM sleepday; -- Now there are 410 rows in sleepday rather than 413**

**-- Check for duplicate rows in weightloginfo**

**SELECT Id, WLIDate, WLITime, WeightKg, WeightPounds, Fat, BMI, ISManualReport, LogId, COUNT(\*)**

**FROM weightloginfo**

**GROUP BY Id, WLIDate, WLITime, WeightKg, WeightPounds, Fat, BMI, ISManualReport, LogId**

**HAVING COUNT(\*) >1; -- there are no duplicate rows within weightloginfo**

**-- 7. Check for NULL or missing values**

**-- weightloginfo was the only table that displayed missing values, there were no NULLS (65/67 records), when it comes to analysis, i will not being using the FAT column.**

**SELECT \* FROM weightloginfo**

**WHERE Id IS NULL OR Id = ''**

**OR WLIDate IS NULL**

**OR WLITime IS NULL OR WLITime = ''**

**OR WeightKg IS NULL OR WeightKg = ''**

**OR WeightPounds IS NULL OR WeightPounds = ''**

**OR Fat IS NULL OR Fat = ''**

**OR BMI IS NULL OR BMI = ''**

**OR IsManualReport IS NULL OR IsManualReport = ''**

**OR LogID IS NULL OR LogID = '';**