**Working with SQL**

**Uploading Datasets to BigQuery:**

While attempting to upload certain .csv files to BigQuery using Schema Auto-detect, an error message was displayed indicating that some of the column formats in the csv files were not recognized by BigQuery. As a result, I proceeded to manually complete the Schema section by entering the following query in the “Edit as text” option:

[

    {

        "name": "Id",

        "type": "INTEGER",

        "mode": "NULLABLE",

        "description": "User unique number"

    },

    {

        "name": "Date",

        "type": "TIMESTAMP",

        "mode": "NULLABLE",

        "description": "Date in format MM/dd/YYYY hh:mm:ss a"

    },

    {

        "name": "WeightKg",

        "type": "FLOAT",

        "mode": "NULLABLE",

        "description": "Weight in Kg"

    },

    {

        "name": "WeightPounds",

        "type": "FLOAT",

        "mode": "NULLABLE",

        "description": "Weight in lbs"

    },

    {

        "name": "Fat",

        "type": "STRING",

        "mode": "NULLABLE",

        "description": "Fat percentage"

    },

    {

        "name": "BMI",

        "type": "FLOAT",

        "mode": "NULLABLE",

        "description": "Body Mass Index"

    },

    {

        "name": "IsManualReport",

        "type": "BOOLEAN",

        "mode": "NULLABLE",

        "description": "Manually logged"

    },

    {

        "name": "LogId",

        "type": "FLOAT",

        "mode": "NULLABLE",

        "description": "Log Number"

    }

]

However, the “Date” column could not be parsed as TIMESTAMP by BigQuery as it only accepts the following formats: 'YYYY-MM-DD HH:MM[:SS[.SSSSSS]]' or 'YYYY/MM/DD HH:MM[:SS[.SSSSSS]]'. Consequently, I attempted to add a format line to 'Date' in the following manner:

    {

        "name": "Date",

        "type": "TIMESTAMP",

        "mode": "NULLABLE",

        "description": "Date in format MM/dd/YYYY hh:mm:ss a"

  "format": "MM/dd/YYYY hh:mm:ss a”

    },

This approach was also unsuccessful as it is not supported by BigQuery. The final option that came to mind was to manually create the table by writing a query in the Big Console. Therefore, the query I composed is as follows:  
  
CREATE TABLE `Fitbit\_Fitness\_Data.deleteTable`

(

  Id INTEGER,

  Date TIMESTAMP,

  WeightKg FLOAT64,

  WeightPounds FLOAT64,

  Fat STRING,

  BMI FLOAT64,

  IsManualReport BOOLEAN,

  LogId FLOAT64

);

INSERT INTO `Fitbit\_Fitness\_Data.deleteTable`

SELECT

  Id,

  PARSE\_TIMESTAMP('%m/%d/%Y %I:%M:%S %p', Date) AS Date,

  WeightKg,

  WeightPounds,

  Fat,

  BMI,

  IsManualReport,

  LogId

FROM

  CSV (

    `https://drive.google.com/file/d/1TYAR6UI3oxWFjA\_Mf3gFCLC9sxFsFNaT/view?usp=sharing`,

    OPTIONS('skip\_first\_row'='true')

  )

This method appeared to be successful. However, it is important to consider the following key factors:

* INSERT INTO is not available for Sandbox accounts. Therefore, if you wish to create a table using this method, you must upgrade your BigQuery account.
* In order to load data into BigQuery, you must provide a URI that directs to a file in a supported storage service, such as Google Cloud Storage or Google Drive. If the file is not stored in one of these options, it will not work.

**Cleaning data using SQL:**

* Cleaning the data is crucial before analyzing it. Hence, since multiple datasets are utilized in this project, each dataset must be cleaned.
* The following steps can be taken to clean the data:
  1. Verifying that there are no duplicates:
     1. By using the DISTINCT function, we can ensure that there are no duplicates when compared to the total observations (without using DISTINCT).
     2. Table: dailyActivity:
        1. SELECT
        2. DISTINCT Id, ActivityDate
        3. FROM `juan-capstone-project-1.Fitbit\_Fitness\_Data.dailyActivity`
           1. This query made sure that both the id and ActivityDate together were unique from other entries.
  2. Ensuring consistent Id length:
     1. The LENGTH function can determine the length of observations. However, as the Id is in INT64 format, LENGTH alone is insufficient. Therefore, CAST should also be used to convert the Id to STRING:
        1. SELECT
        2. DISTINCT(LENGTH(CAST(Id as STRING))) AS num\_in\_Id
        3. FROM `juan-capstone-project-1.Fitbit\_Fitness\_Data.dailyActivity`
  3. Verifying that all attributes had the correct format:
     1. This was achieved by reviewing the SCHEMA of each dataset. If any format was incorrect, the CAST function would have been utilized.
     2. All datasets were found to have the correct format.
  4. Identifying empty cells in the datasets:
     1. This was accomplished by filtering each column with the IS NULL command. Fortunately, there were no empty cells in any of the datasets.
        1. SELECT \*
        2. FROM `juan-capstone-project-1.Fitbit\_Fitness\_Data.dailyActivity`
        3. WHERE
        4. Id IS NULL OR
        5. ActivityDate IS NULL OR …………
        6. TotalSteps IS NULL