Data Analysis Project
P2: Data Integration
Analysis of speed and traffic congestion in Basel (Group 10)

Severin Memmishofer and Yash Trivedi

University of Basel
Databases (CS244) course
Autumn Semester 2022

1 Database Setup Guide

Step 1.1: Clean the datasets

Make sure that there exists a directory named 'DataSources' in the same directory, where all the python files are located.

Make sure that the datasets exist in the 'DataSources' directory in a .csv format, and that they are named correctly; e.g. '100006.csv' and '100097.csv'. (As they are downloaded from the source; 'data.bs.ch').

Now you are ready to run the 'datacleaning.py' code; which calls the other two python files and performs the respective data cleaning steps of the data sources.

If everything worked correctly, there should now exist a folder in the same directory, named 'cleanedFiles'.

Step 1.2: Create tables in DataGrip

Make sure that MySQL and DataGrip (or any of your desired database tools) are configured and installed correctly. Start a mySQL server inside DataGrip (or connect to it, if it's already running) Open and run 'createTables.sql' script in DataGrip.

Step 1.3: Import data into mySQL via the terminal

First we need to set up the mySQL commandline interface. *Note: This step may work differently, depending on your system configuration and the OS used.* For us it worked like this (for MacOS):

```
cd /usr/local/mysql-8.2.0-macos13-arm64/bin
```

Modify the following file: `open -e ~/.zshrc` and insert the following line:

```
export PATH="/usr/local/mysql-8.2.0-macos13-arm64/bin:$PATH"
```

Now you should be able to directly execute SQL commands like this:

```
mysql --version
```

If this command works, and gives you the SQL version, you should be ready to go.

Log into mySQL using the following command and enter your password:

```
mysql --local-infile=1 -u root -p
```

Note: we used the root user/account; change this depending on the account used

Enable local infile data loading via the following command, if it isn't already:

```
set global local_infile=true;
```

to check, if it worked correctly:

```
show global variables like 'local_infile';
```

This should now show the 'local_infile' variable as ON.

Now, we can finally load the Data into the database, using the sql script 'loadDataInfile.sql'. Run this in the mySQL command line interface:

```
source 'PathToSQLFile'
```

Note: adjust the file path depending on your local system and storage location; enter the absolute path to the location of the 'loadDataInfile.sql'file

Now, after this is done correctly, you should be able to take a look at the tables inside DataGrip.

Step 1.4: Create the Integrated Database on mySQL

Run the 'integrated.sql' script inside DataGrip. This should load the data into the integrated SQL table, and performs some additional steps, like creating coherent artificial primary keys.

2 Data Access

The SQL Dump can be accessed via the following access link for SWITCHfilesender:

`https://filesender.switch.ch/filesender2/?s=download&token=d88d091d-c4ea-4b69-906d-75bd5ed60063`

3 Revision

As a small revision, compared to the integrated ER diagram in the previous milestone, we added attribute 'TIME' to the 'Velocity'-table, as we discussed with the tutors. This way, we don't lose any granularity, and if we might need the exact time of a Velocity Measurement, we already have it in the integrated Database. This can be seen in the following CREATE TABLE SQL statement:

```
CREATE TABLE Velocity (
    Measurement_ID INT PRIMARY KEY,
    Time TIME,
    Speed INT,
    Zone INT,
    Vehicle_length DECIMAL(3, 1),
    FOREIGN KEY (Measurement_ID) REFERENCES Measurement(Measurement_ID)
);
```