**Data Loading Script**

use bus\_transport;

**INTERNAL TABLES:**

Tables stored as ORC or PARQUET needs to be loaded with usage of other tables stored as text file – as we are loading data from .txt files

CREATE TABLE route\_txt (

    route\_id INT,

    route\_name STRING,

    metrics MAP<STRING, INT>

)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ','

COLLECTION ITEMS TERMINATED BY '|'

MAP KEYS TERMINATED BY ':'

STORED AS TEXTFILE;

LOAD DATA INPATH 'hdfs:///user/mmajewska/Route\_2020.txt' overwrite INTO TABLE route\_txt;

INSERT OVERWRITE table route SELECT \* FROM route\_txt;

drop table route\_txt;

CREATE TABLE junk\_txt (

    junk\_id INT,

    satisfaction\_level\_category STRING,

    occupation\_level\_category STRING

)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ','

STORED AS TEXTFILE;

LOAD DATA INPATH 'hdfs:///user/mmajewska/Junk.txt' overwrite INTO TABLE junk\_txt;

INSERT OVERWRITE table junk SELECT \* FROM junk\_txt;

drop table junk\_txt;

**EXTERNAL TABLE:**

Placing file Service\_Office\_2020.txt in directory hdfs:///user/mmajewska/database

**STATIC PARTITIONING:**

Here the same situation as for internal tables - we are creating tables for .txt files (table format is ORC)

CREATE TABLE bus\_txt (

   bus\_id INT,

    bus\_registration STRING,

    bus\_office\_id INT,

   additional\_equipment ARRAY<STRING>,

   bus\_type STRING

)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ','

COLLECTION ITEMS TERMINATED BY '|'

STORED AS TEXTFILE;

LOAD DATA LOCAL INPATH 'Bus\_low\_floor.txt' INTO TABLE bus\_txt;

INSERT OVERWRITE TABLE bus PARTITION (bus\_type='low floor')

SELECT bus\_id, bus\_registration, bus\_office\_id, additional\_equipment FROM bus\_txt;

TRUNCATE TABLE bus\_txt; -- to clean temporary table

LOAD DATA LOCAL INPATH 'Bus\_standard.txt' INTO TABLE bus\_txt;

INSERT INTO TABLE bus PARTITION (bus\_type='standard')

SELECT bus\_id, bus\_registration, bus\_office\_id, additional\_equipment FROM bus\_txt;

TRUNCATE TABLE bus\_txt;

LOAD DATA LOCAL INPATH 'Bus\_minibus.txt' INTO TABLE bus\_txt;

INSERT INTO TABLE bus PARTITION (bus\_type='minibus')

SELECT bus\_id, bus\_registration, bus\_office\_id, additional\_equipment FROM bus\_txt;

drop table bus\_txt;

**DYNAMIC PARTITIONING:**

set hive.exec.dynamic.partition=true;

set hive.exec.dynamic.partition.mode=nonstrict;

Creation of temporary tables – to load all data from file (including partitioning field), need to change storing format to PARQUET or ORC

CREATE TABLE date\_tmp (

    date\_id INT,

    date\_format DATE ,

    year INT,

    month STRING,

    month\_no INT,

   day\_type STRING

)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ','

STORED AS TEXTFILE;

LOAD DATA INPATH 'hdfs:///user/mmajewska/Date.txt' overwrite INTO TABLE date\_tmp;

INSERT OVERWRITE TABLE date\_dim partition(month\_no, day\_type)

SELECT date\_id, date\_format, year, month, month\_no, day\_type FROM date\_tmp;

drop table date\_tmp;

CREATE TABLE time\_tmp (

    time\_id INT,

    hour INT,

    minutes INT,

   time\_of\_day STRING

)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ','

STORED AS TEXTFILE;

LOAD DATA INPATH 'hdfs:///user/mmajewska/Time.txt' overwrite INTO TABLE time\_tmp;

INSERT OVERWRITE TABLE time\_dim partition(time\_of\_day)

SELECT time\_id, hour, minutes, time\_of\_day FROM time\_tmp;

drop table time\_tmp;

CREATE TABLE travel\_tmp (

    bus\_id INT,

    route\_id INT,

   departure\_time INT,

    arrival\_time INT,

   tickets\_validated INT,

   bus\_capacity INT,

   avg\_satisfaction\_level\_received INT,

   satisfaction\_surveys\_number INT,

   junk\_id INT,

  travel\_date INT

)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ','

STORED AS TEXTFILE;

LOAD DATA INPATH 'hdfs:///user/mmajewska/Travel\_2020.txt' overwrite INTO TABLE travel\_tmp;

INSERT OVERWRITE TABLE travel partition(travel\_date)

SELECT bus\_id, route\_id, departure\_time, arrival\_time , tickets\_validated, bus\_capacity, avg\_satisfaction\_level\_received, satisfaction\_surveys\_number, junk\_id, travel\_date FROM travel\_tmp;

drop table travel\_tmp;