=> Convert date to required date format using python

python3 all_semi.py run

=> Create normal table

create table parking_violations_csv

> (

> summons_number bigint, plate_id string, registration_state string, plate_type string, issue_date string, violation_code int, vehicle_body_type string, vehicle_make string, issuing_agency string, street_code_1 string, street_code_2 string, street_code_3 string, vehicle_expiration_date int, violation_location int, violation_precinct int, issuer_precinct int, issuer_code string, issuer_command string, issuer_squad string, violation_time string, time_first_observed string, violation_county string, violation_in_front_of_or_opposite string, house_number string, street_name string, intersecting_street string, date_first_observed string, law_section string, sub_division string, violation_legal_code string, days_parking_in_effect string, from_hours_in_effect string, to_hours_in_effect string, vehicle_color string, unregistered_vehicle string, vehicle_year string, meter_number string, feet_from_curb string, violation_post_code string, violation_description string, no_standing_or_stopping_violation string, hydrant_violation string, double_parking_violation string

>)

> row format delimited

> fields terminated by "\u0059"

> stored as textfile tblproperties ("skip.header.line.count"="1");

OK

Time taken: 0.485 seconds

=> Load data

hive> load data local inpath "/home/cloudera/Downloads/parking_violations_new.csv" into table parking_violations_csv;

=> Create partitioned and bucketed table

create table parking_violations_part_buck

> (

> summons_number bigint, plate_id string, issue_date timestamp, street_code_1 string, street_code_2 string, street_code_3 string, violation_location int, issuer_code string, violation_time string, time_first_observed string, violation_legal_code string, from_hours_in_effect string, to_hours_in_effect string, violation_post_code string, violation_description string

>)

> partitioned by (violation_code int, violation_county string, vehicle_body_type string, violation_precinct int, issuer_precinct int, registration_state string, vehicle_make string)

> clustered by (summons_number)

> sorted by (summons_number ASC) into 200 buckets;

=> Insert into partitioned table

hive> insert overwrite table parking_violations_part_buck

partition (violation_code, violation_county, vehicle_body_type, violation_precinct, issuer_precinct, registration_state, vehicle_make)

select summons_number, plate_id, issue_date, street_code_1, street_code_2, street_code_3, violation_location, issuer_code, violation_time, time_first_observed, violation_legal_code, from_hours_in_effect, to_hours_in_effect, violation_post_code, violation_description, violation_code, violation_county, vehicle_body_type, violation_precinct, issuer_precinct, registration_state, vehicle_make

from parking_violations_csv limit 10000;

sample_output => Partition big_data.parking_violations_part_buck{violation_code=84, violation_county=NY, vehicle_body_type=DELV, violation_precinct=1, issuer_precinct=1, registration_state=PA, vehicle_make=MACK} stats: [numFiles=1, numRows=1, totalSize=115, rawDataSize=114]

==> select * from parking_violations_part_buck limit 2;

==> 8497064320 0XZ4943 2017-06-08 00:00:00 10,610 0 0 14
361,062 0239P 11 01-No Intercity Pmt Displ
1 NY BUS 14 14 NJ FORD

8528412155 P942771 2017-04-27 00:00:00 26,550 65,020 41,620 43 364,841
0309P 17 10-No Stopping 10 BX TRLR 43
43 IL FRUEH

Time taken: 13.984 seconds, Fetched: 2 row(s)

1.) Find the total number of tickets for the year.

=> select count(summons_number) from parking_violations_part_buck;

=> output = 10000

2.) Find out how many unique states the cars which got parking tickets came from

hive> select count(distinct Registration_State) as Reg_state_count from parking_violations_part_buck;

=> output = reg_state_count

50

3.) Some parking tickets don't have addresses on them, which is cause for concern. Find out how many such tickets there are(i.e. tickets where either "Street Code 1" or "Street Code 2" or "Street Code 3" is empty)

hive> select count(distinct summons_number) as without_address from parking_violations_part_buck where Street_code_1 = 0 or Street_code_2 = 0 or Street_code_3 = 0;

=> output: without_address

3416

1.) How often does each violation code occur? (frequency of violation codes - find the top 5)

Hive> select count(Violation_Code) as frequency_of_violation,Violation_Code from parking_violations_part_buck group by Violation_Code order by frequency_of_violation desc limit 5;

frequency_of_violation violation_code

1383 21

1297 36

948 38

840 14

603 20

2.) How often does each vehicle body type get a parking ticket? How about the vehicle make? (find the top 5 for both)

Hive> select Vehicle_Body_Type,count(summons_number)as frequency_of_getting_parking_ticket from parking_violations_part_buck group by Vehicle_Body_Type order by frequency_of_getting_parking_ticket desc limit 5;

vehicle_body_type frequency_of_getting_parking_ticket

SUBN 3462

4DSD 2870

VAN 1329

DELV 641

SDN 381