**Hive Use Case Example**

**Problem Statement**

There are about 35,000 crime incidents that happened in the city of San Francisco in the last 3 months.

Our task is to store this relational data in an RDBMS. Use Sqoop to import it into Hadoop.

Can we answer the following queries on this data:

* Relative frequencies of different types of crime incidents
* Crime occurrence frequency as a function of day of the week

<https://data.sfgov.org/Public-Safety/SFPD-Incidents-Previous-Three-Months/tmnf-yvry>

**Step-1:**

Download the data into .csv format

**Step-2:**

Copy the file into hdfs, as shown as below command,



|  |  |
| --- | --- |
| 1  2 | hadoop fs -copyFromLocal /home/user1/Desktop/incidents.csv /user/hadoop |

**Step-3:**

if you observe the data,



|  |  |
| --- | --- |
| 1  2 | 150142102,OTHER OFFENSES,"DRIVERS LICENSE, SUSPENDED OR REVOKED",Sunday,02/15/2015,23:35,TENDERLOIN,"ARREST, BOOKED",TAYLOR ST / GEARY ST,-122.411518820359,37.7869408998805,"(37.7869408998805, -122.411518820359)",15014210265016 |

**Here,**



|  |  |
| --- | --- |
| 1  2 | "DRIVERS LICENSE, SUSPENDED OR REVOKED" --> 'description' |

column is surrounded by quotes and in between we have commas. so while exporting that data into RDBMS we need to remove that quotes and we should not consider comma inside ‘description’ as delimiter. This can be achieved with the help of CSVSerde in hive. Below is the high level notes on it.

**To Setup the CSVSerde, do the following:**

1. Download the CSV SerDe jar to from <https://github.com/ogrodnek/csv-serde>.
2. In Hive CLI, submit “add jar <*path-to-serde-jar*>".



|  |  |
| --- | --- |
| 1  2  3  4 | add jar path/to/csv-serde.jar;  create table my\_table(a string, b string, ...)  row format serde 'com.bizo.hive.serde.csv.CSVSerde' stored as textfile ; |

1. The list jars command should display the CSV SerDe if it has been added successfully.



|  |  |
| --- | --- |
| 1  2  3 | hive> list jars;  csv-serde-1.1.2-0.11.0-all.jar |

1. To add the CSV SerDe to an existing table:



|  |  |
| --- | --- |
| 1  2 | ALTER TABLE <table\_name> SET SERDE 'com.bizo.hive.serde.csv.CSVSerde' |

1. To add the CSV SerDe to a new table:



|  |  |
| --- | --- |
| 1  2 | CREATE TABLE <table\_name> (col\_name\_1 type1, col\_name\_2 type2, ...) row format serde 'com.bizo.hive.serde.csv.CSVSerde'; |

1. The default separator, quote, and escape characters from the opencsv library are:



|  |  |
| --- | --- |
| 1  2  3  4  5 | DEFAULT\_ESCAPE\_CHARACTER \  DEFAULT\_QUOTE\_CHARACTER  "  DEFAULT\_SEPARATOR        , |

1. We can use SERDEPROPERTIES to override default escape, quite and separator characters.  For example, the following overrides the defaults, using the tab character instead of comma for the separator, using a single quote instead of a double quote, and two backslashes for the escape character:



|  |  |
| --- | --- |
| 1  2 | . . ROW FORMAT SERDE 'com.bizo.hive.serde.csv.CSVSerde' WITH SERDEPROPERTIES ( "separatorChar" = "\t", "quoteChar" = "'", "escapeChar" = "\\" ) |

To achieve this we are first creating hive table and loading data into it,



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | CREATE TABLE  IncidentJson (IncidntNum int,                              Category string,                              Descript string,                              DayOfWeek string,                              dDate string,                              Ttime string,                              PdDistrict string,                              Resolution string,                              Address string,                              x string,                              y string,                              LLocation string,                              PdId string)  ROW FORMAT SERDE 'com.bizo.hive.serde.csv.CSVSerde' WITH SERDEPROPERTIES("separatorChar" = "\,","quoteChar" = "\"");    LOAD DATA INPATH '/user/hadoop/abc.csv' INTO TABLE IncidentJson; |

**Step-4:**

Now we are again copying data from hive table to hdfs to remove commas inside the description specified above.



|  |  |
| --- | --- |
| 1  2  3 | INSERT OVERWRITE DIRECTORY '/user/hadoop/' SELECT printf("%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s", IncidntNum, Category, Descript, DayOfWeek, dDate, Ttime, PdDistrict, Resolution, Address, x, y, LLocation, PdId)  FROM IncidentJson; |

**Step-5:**

Now exporting the data into RDBMS using Sqoop,



|  |  |
| --- | --- |
| 1  2 | sqoop export --connect jdbc:mysql//localhost/sqoop\_export --table crime\_incidents --username root -P --export-dir /user/hadoop/ |

**Step-6:**

**Final output:**

* Relative frequencies of different types of crime incidents



|  |  |
| --- | --- |
| 1  2 | SELECT Category, count(Category) from Incident group by Category; |

* Crime occurrence frequency as a function of day of the week



|  |  |
| --- | --- |
| 1  2 | SELECT Category,count(Category), DayOfWeek from crime\_incidents group by DayOfWeek; |