databricks Criminal Expungement Lab

(https://databricks.com)

Working with Virginia Criminal Expungement Data

Last updated: June 10, 2021

Source:

https://virginiacourtdata.org/ (https://virginiacourtdata.org/)

INSTRUCTIONS

In this project, you will upload a criminal expungement dataset, save it as a Delta table, and perform various tasks. Some of these tasks reinforce your Spark SQL skills. Be sure to show all code and results.

TOTAL POINTS: 10

1. (1 PT) Upload the dataset from Collab and load it into a Delta lake table

```
# File location and type
file_location = "/FileStore/tables/circuit_criminal_2000_anon_00.csv"
file_type = "csv"

# CSV options
infer_schema = "false"
first_row_is_header = "true"
delimiter = ","

# The applied options are for CSV files. For other file types, these will be ignored.
df = spark.read.format(file_type) \
  .option("inferSchema", infer_schema) \
  .option("header", first_row_is_header) \
  .option("sep", delimiter) \
  .load(file_location)

display(df)
df.write.format('delta').mode('overwrite').save("/tmp/delta-table")
```

Table

	HearingDate	HearingResult	HearingJury	HearingPlea	HearingType
1	2000-12-19	Dismissed	null	null	Under Advisement
2	2000-09-19	Dismissed	null	null	Trial
3	2000-09-07	Sent	null	null	Trial
4	2000-09-07	Nolle Prosequi	null	null	Trial
5	2000-09-07	Sent	null	null	Trial
6	2000-09-07	Sent	null	null	Trial

4,510 rows | Truncated data

```
file_location = "/tmp/delta-table"
# CSV options
infer_schema = "false"
first_row_is_header = "true"
delimiter = ","
df = spark.read.format('delta') \
  .option("inferSchema", infer_schema) \
  .option("header", first_row_is_header) \
  .option("sep", delimiter) \
  .load(file_location)

# Create a view or table

temp_table_name = "expungement"

df.createOrReplaceTempView(temp_table_name)
```

2. (1 PT) Select the first 5 records from the delta lake table

```
sqlDF = spark.sql("SELECT * FROM expungement")
sqlDF.show(5, False)
```

```
+-----------+--------------+-----------+-----------+---------------+-----------+----+----------+-----------------------
-----+-----------------+----+-----------------------------+--------------------+-------------------------------+-------
----+-----------+-----+-----------+----------+--------------------+---------------+--------------+-------------+--------
---------+-----------------+----------------+---------------------+---------+------------+-----------------+-------------
----------------+----------+-----+-------------+-----------+-------------+-------------+---------------+-----------------
+----------------------+--------------------+-----------------------+------------------+---------------------+----------
-----+-----------------+--------+--------------+------------+---------------+
|HearingDate|HearingResult |HearingJury|HearingPlea|HearingType     |HearingRoom|fips|Filed     |Commencedby
|Locality          |Sex |Race                          |Address              |Charge                         |CodeSection|
ChargeType |Class|OffenseDate|ArrestDate|DispositionCode     |DispositionDate|ConcludedBy   |AmendedCharge|AmendedCodeSec
tion|AmendedChargeType|JailPenitentiary|ConcurrentConsecutive|LifeDeath|SentenceTime|SentenceSuspended|OperatorLicenseSus
pensionTime|FineAmount|Costs|FinesCostPaid|ProgramType|ProbationType|ProbationTime|ProbationStarts|CourtDMVSurrender|Driv
erImprovementClinic|DrivingRestrictions|RestrictionEffectiveDate|RestrictionEndDate|VAAlcoholSafetyAction|RestitutionPaid
|RestitutionAmount|Military|TrafficFatality|AppealedDate|person_id      |
+-----------+--------------+-----------+-----------+---------------+-----------+----+----------+-----------------------
-----+-----------------+----+-----------------------------+--------------------+-------------------------------+-------
----+-----------+-----+-----------+----------+--------------------+---------------+--------------+-------------+--------
---------+-----------------+----------------+---------------------+---------+------------+-----------------+-------------
----------------+----------+-----+-------------+-----------+-------------+-------------+---------------+-----------------
+----------------------+--------------------+-----------------------+------------------+---------------------+----------
-----+-----------------+--------+--------------+------------+---------------+
```

3. (1 PT) Show the number ("count") of each ChargeType in the table, and sort descending by count

```
sqlDF = spark.sql('''SELECT Chargetype, count(Chargetype) AS charge_counts
                  FROM expungement
                  GROUP BY Chargetype
                  ORDER BY charge_counts
                  DESC''')
sqlDF.show()
```

```
+--------------------+-------------+
|          Chargetype|charge_counts|
+--------------------+-------------+
|              Felony|        92561|
|         Misdemeanor|        43921|
|Other (Animal Vio...|            8|
|          Infraction|            5|
```

```
|              Civil|           1|
|               null|           0|
+-------------------+------------+
```

4. (2 PTS) Create a small dataframe with these specs:

- 2 rows of data
- a subset of the columns from the Delta table
- a column that is NOT in the Delta table

Next, try writing the records to the Delta table. Since the records don't follow the schema of the Delta table, it would be problematic if this data was written.
This is one of the issues with a *data lake*: it can become a dumping ground for bad data, producing a *data swamp*.

Fortunately, a Delta table prevents the write. **Make sure to show the error message that prints.**

```
sqlDF = spark.sql('''SELECT HearingResult, HearingType, Chargetype,
                  CAST(fips AS float)/2 AS fips_div
                  FROM expungement
                  ORDER BY fips_div
                  DESC
                  LIMIT 2''')
sqlDF.show()
```

```
+-------------+-----------+----------+--------+
|HearingResult|HearingType|Chargetype|fips_div|
+-------------+-----------+----------+--------+
|         Sent|       Plea|    Felony|   420.0|
|         Sent|       Plea|    Felony|   420.0|
+-------------+-----------+----------+--------+
```

```
sqlDF.write.format("delta").mode("append").save("/tmp/delta-table")
```

```
  AnalysisException: A schema mismatch detected when writing to the Delta table (Table ID: 67a74c53-7792-4824-b814-7f4b8b
d9ebf0).
To enable schema migration using DataFrameWriter or DataStreamWriter, please set:
'.option("mergeSchema", "true")'.
For other operations, set the session configuration
spark.databricks.delta.schema.autoMerge.enabled to "true". See the documentation
specific to the operation for details.

Table schema:
root
-- HearingDate: string (nullable = true)
-- HearingResult: string (nullable = true)
-- HearingJury: string (nullable = true)
-- HearingPlea: string (nullable = true)
-- HearingType: string (nullable = true)
-- HearingRoom: string (nullable = true)
-- fips: string (nullable = true)
-- Filed: string (nullable = true)
-- Commencedby: string (nullable = true)
-- Locality: string (nullable = true)
-- Sex: string (nullable = true)
-- Race: string (nullable = true)
-- Address: string (nullable = true)
-- Charge: string (nullable = true)
-- CodeSection: string (nullable = true)
-- ChargeType: string (nullable = true)
-- Class: string (nullable = true)
-- OffenseDate: string (nullable = true)
-- ArrestDate: string (nullable = true)
```

```
-- DispositionCode: string (nullable = true)
-- DispositionDate: string (nullable = true)
-- ConcludedBy: string (nullable = true)
-- AmendedCharge: string (nullable = true)
-- AmendedCodeSection: string (nullable = true)
-- AmendedChargeType: string (nullable = true)
-- JailPenitentiary: string (nullable = true)
-- ConcurrentConsecutive: string (nullable = true)
-- LifeDeath: string (nullable = true)
-- SentenceTime: string (nullable = true)
-- SentenceSuspended: string (nullable = true)
-- OperatorLicenseSuspensionTime: string (nullable = true)
-- FineAmount: string (nullable = true)
-- Costs: string (nullable = true)
-- FinesCostPaid: string (nullable = true)
-- ProgramType: string (nullable = true)
-- ProbationType: string (nullable = true)
-- ProbationTime: string (nullable = true)
-- ProbationStarts: string (nullable = true)
-- CourtDMVSurrender: string (nullable = true)
-- DriverImprovementClinic: string (nullable = true)
-- DrivingRestrictions: string (nullable = true)
-- RestrictionEffectiveDate: string (nullable = true)
-- RestrictionEndDate: string (nullable = true)
-- VAAlcoholSafetyAction: string (nullable = true)
-- RestitutionPaid: string (nullable = true)
-- RestitutionAmount: string (nullable = true)
-- Military: string (nullable = true)
-- TrafficFatality: string (nullable = true)
-- AppealedDate: string (nullable = true)
-- person_id: string (nullable = true)


Data schema:
root
-- HearingResult: string (nullable = true)
-- HearingType: string (nullable = true)
-- Chargetype: string (nullable = true)
-- fips_div: double (nullable = true)
```

5. (1 PT) Explain the difference between INSERTING records and UPSERTING records.

An upsert is both an insert and update combined into one step. In an upsert, you can both insert new data and update existing data by, for example, transforming a column. We talked in class about how an upsert could be used to insert new data and update a column we transformed to measure temperature in Farenheit instead of Celsius. On the other hand, an insert solely inserts new records into an existing data lake without updating existing records or transforming the database (or its columns) in any way.

6. (2 PTS) You realized that all records where ChargeType: 'Infraction' should actually be ChargeType: 'Minor Infraction'
   Make this update to the Delta lake, and then rerun your query from Question 3.
   This should show that the Infractions migrated to Minor Infractions.

```
from delta.tables import *
from pyspark.sql.functions import *

# set the path
deltaTable = DeltaTable.forPath(spark, "/tmp/delta-table")
deltaTable.update(
  condition = expr("ChargeType=='Infraction'"),
  set = { "ChargeType": expr("'Minor Infraction'") })
df = deltaTable.toDF()
df.createOrReplaceTempView(temp_table_name)
sqlDF = spark.sql('''SELECT Chargetype, count(Chargetype) AS charge_counts
                  FROM expungement
                  GROUP BY Chargetype
                  ORDER BY charge_counts
                  DESC''')
sqlDF.show()
```

```
+--------------------+-------------+
|          Chargetype|charge_counts|
+--------------------+-------------+
|              Felony|        92561|
|         Misdemeanor|        43921|
|Other (Animal Vio...|            8|
|    Minor Infraction|            5|
|               Civil|            1|
|                null|            0|
+--------------------+-------------+
```

7. (2 PTS) You realize that actually 'Infraction' was the correct label for the migrated records from Question 6.
   Use the time travel functionality to load the original version of the delta table, and display the records with ChargeType: 'Infraction'
   Show only these columns: HearingDate, HearingResult, ChargeType

Wow, this feature bailed you out!

```
deltaTable = DeltaTable.forPath(spark, "/tmp/delta-table")
fullHistoryDF = deltaTable.history()    # get the full history of the table
fullHistoryDF.select('version','operation','operationMetrics').show(truncate=False)
```

```
+-------+---------+---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------+
|version|operation|operationMetrics
|
+-------+---------+---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------+
|2      |UPDATE   |{numRemovedFiles -> 3, numRemovedBytes -> 1370670, numCopiedRows -> 42783, numDeletionVectorsAdded -> 0, numDeletionVectorsRemoved -> 0, numAddedChangeFiles -> 0, executionTimeMs -> 16149, scanTimeMs -> 3990, numAddedFiles -> 3, numUpdatedRows -> 5, numAddedBytes -> 1370686, rewriteTimeMs -> 12132}|
|1      |WRITE    |{numFiles -> 8, numOutputRows -> 136499, numOutputBytes -> 4317006}
|
|0      |WRITE    |{numFiles -> 8, numOutputRows -> 136499, numOutputBytes -> 4317006}
|
+-------+---------+---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------+
```

```
+-----------+--------------------+----------+
|HearingDate|       HearingResult|ChargeType|
+-----------+--------------------+----------+
| 2000-02-03|           Dismissed|Infraction|
```

```
| 2000-06-15|               Sent|Infraction|
| 2000-10-16|Resolved Order Pe...|Infraction|
| 2000-01-12|           Resolved|Infraction|
| 2000-05-10|               Sent|Infraction|
+-----------+--------------------+----------+
```