Tutorial to create beeline tables

Step 9: Creating Table - top_crime_types

- **Purpose**: To create a table capturing the top 15 crime types based on their frequency.
- **Guide**:
    - Execute the query to drop the existing "top_crime_types" table if it exists: DROP TABLE IF EXISTS top_crime_types;
    - Replace placeholders if necessary.
    - Run the query provide to create the new table: CREATE TABLE IF NOT EXISTS top_crime_types ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' STORED AS TEXTFILE LOCATION 'Crimes/top_crime_types' AS SELECT Primary_Type, COUNT(*) AS crime_count FROM crime GROUP BY Primary_Type ORDER BY crime_count DESC LIMIT 15;

Step 10: Retrieving Data - top_crime_types

- **Purpose**: To retrieve data from the "top_crime_types" table.
- **Guide**:
    - Execute the query **SELECT * FROM top_crime_types LIMIT 15;** to view the top 15 crime types.

Step 11: File Management - top_crime_types

- **Purpose**: To check the existence of the table file in HDFS.
- **Guide**:
    - Run **hdfs dfs -ls /user/username/top_crime_types** to list files in the directory.

Step 12: Downloading Data - top_crime_types

- **Purpose**: To download the table file from HDFS to the local machine.
- **Guide**:
    - Run **hdfs dfs -get /user/username/top_crime_types/000000_0 top_crime_types.csv** to download the file.

Step 13: Transfer Data - top_crime_types

- **Purpose**: To transfer the downloaded file to another location.
- **Guide**:
    - Open a new terminal window.
    - Use **scp** to transfer the file to another location: scp username@IP_address:/home/username/top_crime_types.csv .

Step 14: Creating Table - Arrest Analysis Table

- **Purpose**: To create a table analyzing the percentage of reported crimes resulting in arrests.
- **Guide**:
    - Execute the query to drop the existing "arrest_analysis" table if it exists: DROP TABLE IF EXISTS arrest_analysis;
    - Run the query provide to create the new table: CREATE TABLE arrest_analysis ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' STORED AS

TEXTFILE LOCATION 'Crimes/arrest_analysis' AS SELECT crime.primary_type, SUM(CASE WHEN crime.arrest = true THEN 1 ELSE 0 END) AS arrests, COUNT(*) AS total_crimes, (SUM(CASE WHEN crime.arrest = true THEN 1 ELSE 0 END) / COUNT(*)) * 100 AS arrest_percentage FROM crime GROUP BY crime.primary_type;

Step 15: Retrieving Data - Arrest Analysis Table

- **Purpose**: To retrieve data from the "arrest_analysis" table.
- **Guide**:
  - Execute the query **SELECT * FROM arrest_analysis LIMIT 15;** to view the data.

Step 16: File Management - Arrest Analysis Table

- **Purpose**: To check the existence of the table file in HDFS.
- **Guide**:
  - Run **hdfs dfs -ls /user/username/arrest_analysis** to list files in the directory.

Step 17: Deleting Files - Arrest Analysis Table

- **Purpose**: To delete unnecessary files from HDFS.
- **Guide**:
  - Run **rm 00000*_0** to delete files.

Step 18: Downloading Data - Arrest Analysis Table

- **Purpose**: To download the table file from HDFS to the local machine.
- **Guide**:
  - Run **hdfs dfs -get /user/username/arrest_analysis/00000*_0 arrest_analysis.csv**.

Step 19: Transfer Data - Arrest Analysis Table

- **Purpose**: To transfer the downloaded file to another location.
- **Guide**:
  - Open a new terminal window.
  - Use **scp** to transfer the file to another location: scp username@IP_address:/home/username/arrest_analysis.csv .

Step 20: Creating Table - Temporal Analysis Table

- **Purpose**: To create a table analyzing the temporal distribution of crimes.
- **Guide**:
  - Execute the query to drop the existing "temporal_analysis" table if it exists: DROP TABLE IF EXISTS temporal_analysis;
  - Run the provide query to create the new table: CREATE TABLE temporal_analysis ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' STORED AS TEXTFILE LOCATION 'Crimes/temporal_analysis' AS SELECT YEAR(FROM_UNIXTIME(UNIX_TIMESTAMP(crime.date_time, 'MM/dd/yyyy hh:mm:ss a'))) AS crime_year,

MONTH(FROM_UNIXTIME(UNIX_TIMESTAMP(crime.date_time, 'MM/dd/yyyy hh:mm:ss a'))) AS crime_month, COUNT(*) AS crime_count FROM crime GROUP BY YEAR(FROM_UNIXTIME(UNIX_TIMESTAMP(crime.date_time, 'MM/dd/yyyy hh:mm:ss a'))), MONTH(FROM_UNIXTIME(UNIX_TIMESTAMP(crime.date_time, 'MM/dd/yyyy hh:mm:ss a'))) ORDER BY crime_year, crime_month;

Step 21: Retrieving Data - Temporal Analysis Table

- **Purpose**: To retrieve data from the "temporal_analysis" table.
- **Guide**:
  - Execute the query **SELECT * FROM temporal_analysis LIMIT 15;** to view the data.

Step 22: File Management - Temporal Analysis Table

- **Purpose**: To check the existence of the table file in HDFS.
- **Guide**:
  - Run **hdfs dfs -ls /user/username/temporal_analysis** to list files in the directory.

Step 23: Downloading Data - Temporal Analysis Table

- **Purpose**: To download the table file from HDFS to the local machine.
- **Guide**:
  - Run **hdfs dfs -get /user/username/temporal_analysis/000000_0 temporal_analysis.csv**.

Step 24: Transfer Data - Temporal Analysis Table

- **Purpose**: To transfer the downloaded file to another location.
- **Guide**:
  - Open a new terminal window
  - Use **scp** to transfer the file to another location: scp username@IP_address:/home/username/temporal_analysis.csv .

Step 25: Creating Table - Geospatial Analysis Table

- **Purpose**: To create a table extracting geospatial data from the dataset.
- **Guide**:
  - Execute the query to drop the existing "geospatial_analysis" table if it exists: DROP TABLE IF EXISTS geospatial_analysis;
  - Run the provide query to create the new table: CREATE TABLE geospatial_analysis ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' STORED AS TEXTFILE LOCATION 'Crimes/geospatial_analysis' AS SELECT crime.primary_type, crime.latitude, crime.longitude FROM crime WHERE crime.latitude IS NOT NULL AND crime.longitude IS NOT NULL;

Step 26: Retrieving Data - Geospatial Analysis Table

- **Purpose**: To retrieve data from the "geospatial_analysis" table.

- **Guide**:
    - Execute the query **SELECT * FROM geospatial_analysis LIMIT 15;** to view the data.

Step 27: File Management - Geospatial Analysis Table
- **Purpose**: To check the existence of the table file in HDFS.
- **Guide**:
    - Run **hdfs dfs -ls /user/username/geospatial_analysis** to list files in the directory.

Step 28: Deleting Files - Geospatial Analysis Table
- **Purpose**: To delete unnecessary files from HDFS.
- **Guide**:
    - Run **rm 00000*_0** to delete files.

Step 29: Downloading Data - Geospatial Analysis Table
- **Purpose: To download the table file from HDFS to the local machine.**
- **Guide:**
    - **Run hdfs dfs -get /user/ussername/geospatial_analysis/000000_0 geospatial_analysis.csv.**

Step 30: Transfer Data - Geospatial Analysis Table
- **Purpose: To transfer the downloaded file to another location.**
- **Guide:**
    - Open a new terminal window
    - Use **scp** to transfer the file to another location: scp username@IP_address:/home/username/geospatial_analysis.csv .

Step 31: Creating Table - crime_by_location_description Table
- **Purpose: To create a table analyzing crime occurrences based on location descriptions.**
- **Guide:**
    - Execute the query to drop the existing "crime_by_location_description" table if it exists: DROP TABLE IF EXISTS crime_by_location_description;
    - Run the provide query to create the new table: CREATE TABLE crime_by_location_description ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' STORED AS TEXTFILE LOCATION 'Crimes/crime_by_location_description' AS SELECT location_description, COUNT(*) AS crime_count FROM crime GROUP BY location_description ORDER BY crime_count DESC LIMIT 15;

Step 32: Retrieving Data - crime_by_location_description Table
- **Purpose:** To retrieve data from the "crime_by_location_description" table.
- **Guide:**

- Execute the query **SELECT * FROM** crime_by_location_description **LIMIT 15;** to view the data.

Step 33: File Management - crime_by_location_description Table

- **Purpose**: To check the existence of the table file in HDFS.
- **Guide**:
  - Run **hdfs dfs -ls /user/username/crime_by_location_description** to list files in the directory.

Step 34: Deleting Files - crime_by_location_description Table

- **Purpose:** To delete unnecessary files from HDFS.
- **Guide:**
  - Run **rm 00000*_0** to delete files**.**

Step 35: Downloading Data - crime_by_location_description Table

- **Purpose:** To download the table file from HDFS to the local machine.
- **Guide:**
  - Run **hdfs dfs -get /user/username/crime_by_location_description/000000_0 crime_by_location_description.csv.**

Step 36: Transfer Data - crime_by_location_description Table

- **Purpose:** To transfer the downloaded file to another location.
- **Guide:**
  - Open a new terminal window
  - Use **scp** to transfer the file to another location: scp username@IP_address:/home/username/000000_0 crime_by_location_description.txt