## AIM:

Ex No: 7 Export Data from Hadoop using Sqoop and Import Data to Hive using Sqoop

To simulate the process of exporting data from Hadoop Distributed File System (HDFS) and importing it into a Hive table using Sqoop, implemented using Python with SQLite and Pandas.

## Algorithm :

1. Start the program.
2. Generate sample weather data (year-wise temperatures) and store it in a CSV file, simulating an HDFS file.
3. Establish a SQLite connection to simulate a Hive database.
4. Read the CSV file and import its data into the SQLite table, simulating the Sqoop import process.
5. Create an index on the year column to optimize query performance (like Hive index).
6. Query the table to calculate yearly minimum and maximum temperatures.
7. Display the summarized report and sample table data.
8. End the program.

## Python Implementation

import pandas as pd import sqlite3 import random

from contextlib import contextmanager

# Step 1: Generate sample weather data (simulating HDFS CSV file) def generate\_sample\_data(num\_records=1000):

years = list(range(1900, 2021)) data = {

'record\_id': range(1, num\_records + 1),

'year': [random.choice(years) for \_ in range(num\_records)], 'temperature\_c': [random.uniform(-50, 50) for \_ in range(num\_records)]

}

df = pd.DataFrame(data)

csv\_path = 'weather\_data.csv' # Simulating HDFS file df.to\_csv(csv\_path, index=False)

print(f"Sample data generated and saved to {csv\_path} (simulating HDFS file).") return csv\_path

# Step 2: SQLite connection (simulating Hive) @contextmanager

def sqlite\_connection(db\_name): conn = sqlite3.connect(db\_name) try:

yield conn finally:

conn.close()

# Step 3: Simulate Sqoop export/import

def sqoop\_like\_import(csv\_path, db\_name, table\_name): df = pd.read\_csv(csv\_path)

print(f"Sqoop-like export: Read {len(df)} records from {csv\_path} (HDFS).") with sqlite\_connection(db\_name) as conn:

df.to\_sql(table\_name, conn, if\_exists='replace', index=False)

print(f"Sqoop-like import: Loaded data into {db\_name}.{table\_name} (Hive table).") conn.execute(f'CREATE INDEX idx\_year ON {table\_name}(year)')

print(f"Index 'idx\_year' created on {table\_name}.year.")

# Step 4: Generate weather report

def generate\_weather\_report(db\_name, table\_name): with sqlite\_connection(db\_name) as conn:

query = f''' SELECT year,

MIN(temperature\_c) AS min\_temp\_c, MAX(temperature\_c) AS max\_temp\_c

FROM {table\_name} GROUP BY year ORDER BY year

'''

report\_df = pd.read\_sql\_query(query, conn) report\_df['min\_temp\_c'] = report\_df['min\_temp\_c'].round(1) report\_df['max\_temp\_c'] = report\_df['max\_temp\_c'].round(1)

return report\_df

# Step 5: Run program

if name == " main ":

print("=== Simulating Sqoop Export/Import to Hive ===") csv\_path = generate\_sample\_data(1000)

db\_name = 'weather\_hive.db' table\_name = 'weather\_data'

sqoop\_like\_import(csv\_path, db\_name, table\_name)

print("\nGenerating Weather Temperature Statistics Report...") report = generate\_weather\_report(db\_name, table\_name)

print("\n=== Weather Report ===") print("Year\tMin Temp (°C)\tMax Temp (°C)") print("-" \* 35)

for \_, row in report.iterrows(): print(f"{int(row['year'])}\t{row['min\_temp\_c']}\t\t{row['max\_temp\_c']}")

print(f"\nSample data from {table\_name} (first 5 rows):") with sqlite\_connection(db\_name) as conn:

sample\_data = pd.read\_sql\_query(f'SELECT \* FROM {table\_name} LIMIT 5', conn) print(sample\_data)

# Expected Output:

=== Simulating Sqoop Export/Import to Hive ===

Sample data generated and saved to weather\_data.csv (simulating HDFS file). Sqoop-like export: Read 1000 records from weather\_data.csv (HDFS).

Sqoop-like import: Loaded data into weather\_hive.db.weather\_data (Hive table). Index 'idx\_year' created on weather\_data.year.

Generating Weather Temperature Statistics Report...

=== Weather Report ===

Year Min Temp (°C) Max Temp (°C)

|  |  |  |
| --- | --- | --- |
| 1900 | -48.7 | 49.2 |
| 1901 | -44.3 | 47.9 |
| 1902 | -46.1 | 48.5 |
| ... |  |  |
| 2020 | -49.6 | 49.9 |

Sample data from weather\_data (first 5 rows):

record\_id year temperature\_c

|  |  |  |  |
| --- | --- | --- | --- |
| 0 | 1 | 1915 | -10.345678 |
| 1 | 2 | 1992 | 25.456789 |
| 2 | 3 | 2005 | 15.123456 |
| 3 | 4 | 1967 | -22.987654 |
| 4 | 5 | 2018 | 40.678912 |

# Result:

The simulation successfully demonstrated how Sqoop can export data from Hadoop (HDFS) and import it into Hive, using Python and SQLite as a lightweight prototype. It generated a summarized report of yearly minimum and maximum temperatures from the imported dataset.