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

AIM:

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.