me19b190-assignment-5-q2

March 12, 2023

[ ]:
```
!pip install pyspark py4j
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: pyspark in /usr/local/lib/python3.9/dist-packages (3.3.2)
Requirement already satisfied: py4j in /usr/local/lib/python3.9/dist-packages (0.10.9.5)
```

[271]:
```
from pyspark.sql import SparkSession
spark = SparkSession.builder.appName("scd2_demo").getOrCreate()
```

Step1

Get the current customer data and the data that needs to be updated

[272]:
```
#creating current customer data
current_data = """
 SELECT   INT(1) AS id,
          STRING('Harsha') AS name,
          STRING('20-08-1990') AS dob,
          STRING('01-01-1970') AS validity_start,
          STRING('12-12-9999') AS validity_end

 UNION
 SELECT   INT(2) AS id,
          STRING('Goldie') AS name,
          STRING('11-02-1990') AS dob,
          STRING('01-01-1970') AS validity_start,
          STRING('12-12-9999') AS validity_end

 UNION
 SELECT   INT(3) AS id,
          STRING('Divya') AS name,
          STRING('25-12-1990') AS dob,
          STRING('01-01-1970') AS validity_start,
          STRING('12-12-9999') AS validity_end
"""
```

```
df_current_data = spark.sql(current_data)
df_current_data.createOrReplaceTempView("current_data")
df_current_data = spark.sql("SELECT * FROM current_data")
df_current_data.show()
```

```
+---+------+----------+--------------+------------+
| id|  name|       dob|validity_start|validity_end|
+---+------+----------+--------------+------------+
|  1|Harsha|20-08-1990|    01-01-1970|  12-12-9999|
|  2|Goldie|11-02-1990|    01-01-1970|  12-12-9999|
|  3| Divya|25-12-1990|    01-01-1970|  12-12-9999|
+---+------+----------+--------------+------------+
```

[273]:
```
#getting the data that needs to be updated
source_data = """
 SELECT
          STRING('Harsha') AS name,
          STRING('05-09-1990') AS updated_dob"""

df_source = spark.sql(source_data)
df_source.createOrReplaceTempView("customer_data")
df_source = spark.sql("SELECT * FROM customer_data")
df_source.show()
```

```
+------+-----------+
|  name|updated_dob|
+------+-----------+
|Harsha| 05-09-1990|
+------+-----------+
```

Step2

Do a inner join operation to get the names of the customers whose existing data needs to be updated

[274]:
```
# get the to be modified name key
update_curr = """
 SELECT
          customer_data.name
 FROM     customer_data
          INNER JOIN current_data
          ON current_data.name = customer_data.name
"""

to_bemodified = spark.sql(update_curr)
to_bemodified.createOrReplaceTempView("tobe_modified")
```

```
to_bemodified.show()
```

```
+------+
|  name|
+------+
|Harsha|
+------+
```

Step3

Change the validity end date of the current records(records in tobe_modified table) to present day date i.e 12-03-2023,

[275]:
```
# make the end validity date of current records(records in tobe_modified table)␣
 ↪to 12-03-2023
change_date = """
 SELECT   current_data.id,
          current_data.name,
          current_data.dob,
          current_data.validity_start,
          STRING('12-03-2023') AS validity_end


 FROM     current_data
          INNER JOIN tobe_modified
          ON tobe_modified.name = current_data.name

"""
df_changed_validity_date = spark.sql(change_date)

df_changed_validity_date.createOrReplaceTempView("update_curr_records_value")
df_changed_validity_date.show()
```

```
+---+------+----------+--------------+------------+
| id|  name|       dob|validity_start|validity_end|
+---+------+----------+--------------+------------+
|  1|Harsha|20-08-1990|    01-01-1970|  12-03-2023|
+---+------+----------+--------------+------------+
```

Step4

Get the unaffected records from the original customer data

[276]:
```
#get the datasets that are not modified from the original customer data

retrive_unaffected = """
 SELECT   current_data.id,
```

```
          current_data.name,
          current_data.dob,
          current_data.validity_start,
          current_data.validity_end

 FROM     current_data
          LEFT OUTER JOIN tobe_modified
          ON tobe_modified.name = current_data.name
 WHERE    tobe_modified.name IS NULL
"""
df_retrive_unaffected = spark.sql(retrive_unaffected)
df_retrive_unaffected.createOrReplaceTempView("unaffected_recs")
df_retrive_unaffected.show()
```

```
+---+------+----------+--------------+------------+
| id|  name|       dob|validity_start|validity_end|
+---+------+----------+--------------+------------+
|  2|Goldie|11-02-1990|    01-01-1970|  12-12-9999|
|  3| Divya|25-12-1990|    01-01-1970|  12-12-9999|
+---+------+----------+--------------+------------+
```

Step5

Get the new records thats needs to be appended to the final table

[277]:
```
#get the new record that needs to be updated to the SQL table
new_data = """
 SELECT
          current_data.id,
          customer_data.name,
          customer_data.updated_dob,
          '12-03-2023' AS validity_start,
          '12-12-9999' AS validity_end

 FROM     customer_data
          LEFT OUTER JOIN current_data
          ON current_data.name = customer_data.name

"""
df_new_data = spark.sql(new_data)
df_new_data.createOrReplaceTempView("tobeappended_record")
df_new_data.show()
```

```
+---+------+-----------+--------------+------------+
| id|  name|updated_dob|validity_start|validity_end|
+---+------+-----------+--------------+------------+
|  1|Harsha| 05-09-1990|    12-03-2023|  12-12-9999|
+---+------+-----------+--------------+------------+
```

Step6

Combine all the tables and print the resulting SCD type 2 table

[279]:
```
#combine all tables now to get the final SCD type 2 table
command = """
                SELECT  *
                FROM update_curr_records_value
                UNION
                SELECT *
                FROM unaffected_recs
                UNION
                SELECT *
                FROM tobeappended_record
                """

final_table = spark.sql(command)
print("Initial customer data: ")
df_current_data.show()
print("Final SCD type 2 table: ")
final_table.show()
```

```
Initial customer data:
+---+------+----------+--------------+------------+
| id|  name|       dob|validity_start|validity_end|
+---+------+----------+--------------+------------+
|  1|Harsha|20-08-1990|    01-01-1970|  12-12-9999|
|  2|Goldie|11-02-1990|    01-01-1970|  12-12-9999|
|  3| Divya|25-12-1990|    01-01-1970|  12-12-9999|
+---+------+----------+--------------+------------+

Final SCD type 2 table:
+---+------+----------+--------------+------------+
| id|  name|       dob|validity_start|validity_end|
+---+------+----------+--------------+------------+
|  1|Harsha|20-08-1990|    01-01-1970|  12-03-2023|
|  3| Divya|25-12-1990|    01-01-1970|  12-12-9999|
|  2|Goldie|11-02-1990|    01-01-1970|  12-12-9999|
|  1|Harsha|05-09-1990|    12-03-2023|  12-12-9999|
+---+------+----------+--------------+------------+
```