dipesh.tripathi_SQL_p_tk (SQL)

Import Notebook

Imorting Json Header

```
%python
from pyspark.sql import SparkSession
spark = SparkSession.builder.appName("MultilineJSON").getOrCreate()
json_file_path = "dbfs:/FileStore/shared_uploads/tripathidipesh13@gmail.com/header-1.json"
df = spark.read.option("multiLine", "true").json(json_file_path)
df.show()
df.createOrReplaceTempView("temp_view")
spark.sql("CREATE OR REPLACE TABLE header_info AS SELECT * FROM temp_view")
```

```
|70004|     40058|      spouse|
|70005|     40088|      friend|
|70006|     40170|       child|
|70007|     40194|      parent|
|70008|     40079|      spouse|
|70009|     40466|     sibling|
|70010|     40061|       child|
|70011|     40413|      spouse|
|70012|     40237|      spouse|
|70013|     40273|      friend|
|70014|     40484|      parent|
|70015|     40317|      friend|
|70016|     40371|       child|
|70017|     40384|       child|
|70018|     40273|      parent|
|70019|     40217|       child|
|70020|     40391|     sibling|
+-----+----------+------------+
only showing top 20 rows

Out[6]: DataFrame[num_affected_rows: bigint, num_inserted_rows: bigint]
```

Importing Address

```
%python
import pandas as pd
df1 = spark.read.format("csv").option("header", "true").load("dbfs:/FileStore/shared_uploads/tripathidipesh13@gmail.com/Address.csv")
# Convert to Pandas DataFrame
pandas_df = df1.toPandas()
spark_df = spark.createDataFrame(pandas_df)
spark_df.createOrReplaceTempView("temp_view2")
spark.sql("CREATE OR REPLACE TABLE address_info AS SELECT * FROM temp_view2")
```

```
Out[2]: DataFrame[num_affected_rows: bigint, num_inserted_rows: bigint]
```

Importing Details

```
%python
import pandas as pd
df2 = spark.read.format("csv").option("header", "true").load("dbfs:/FileStore/shared_uploads/tripathidipesh13@gmail.com/Detail-2.csv")
# Convert to Pandas DataFrame
pandas_df = df2.toPandas()
spark_df = spark.createDataFrame(pandas_df)
spark_df.createOrReplaceTempView("temp_view3")
spark.sql("CREATE OR REPLACE TABLE detail_info AS SELECT * FROM temp_view3")
```

```
Out[3]: DataFrame[num_affected_rows: bigint, num_inserted_rows: bigint]
```

Importing ContactInfo

```
%python
df4 = spark.read.format("csv") \
    .option("header", "true") \
    .option("sep", "\t") \
    .load("dbfs:/FileStore/shared_uploads/tripathidipesh13@gmail.com/contactinfo-1.txt")

pandas_df4 = df4.toPandas()
spark_df4 = spark.createDataFrame(pandas_df4)
spark_df4.createOrReplaceTempView("temp_view4")

# Create or replace a table using Spark SQL
spark.sql("CREATE OR REPLACE TABLE contact_info AS SELECT * FROM temp_view4")
```

```
Out[4]: DataFrame[num_affected_rows: bigint, num_inserted_rows: bigint]
```

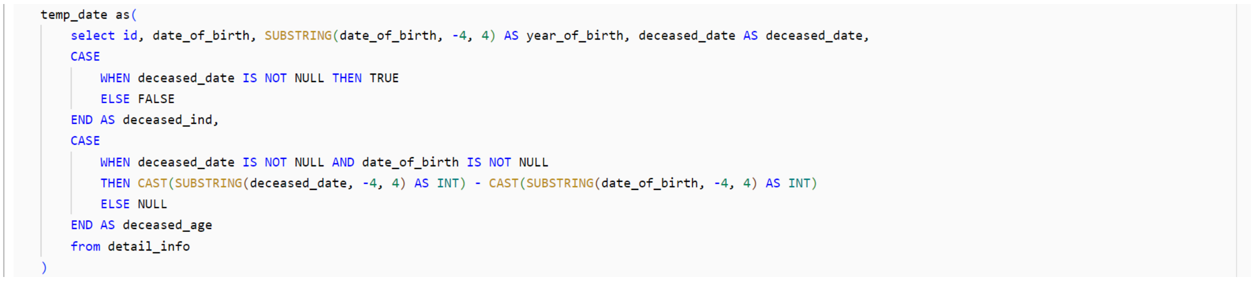
Creating Final Table

```
CREATE OR REPLACE TABLE final_result3(
  source_id STRING,
  subscriber_id STRING,
  first_name STRING,
  middle_name STRING,
  last_name STRING,
  prefix STRING,
  suffix STRING,
  name STRING,
  record_source STRING,
  recorded_ts TIMESTAMP,
  is_verified BOOLEAN,
  Address ARRAY<STRUCT<address_type: STRING, address_line_1: STRING, address_line_2: STRING, city: STRING, state: STRING, ZipCode: STRING, PostalCode: String,
  country: string>>,
  phones ARRAY<STRUCT<phone: STRING, usage_type: STRING>>,
  email STRING,
  privacy_preference BOOLEAN,
  national_id STRING,
  gender STRING,
  marital_status STRING,
  date_of_birth String,
  year_of_birth STRING,
  deceased_ind BOOLEAN,
  deceased_age INT,
  deceased_date String,
  languages STRUCT<spoken_language_1: STRING, spoken_language_2: STRING>,
  employment STRUCT<first_name: STRING, job_role: STRING, employee_status: STRING, job_hiredate: string>,
  additional_source_value MAP<STRING, STRING>
)
```

OK

Insearting Data into Table

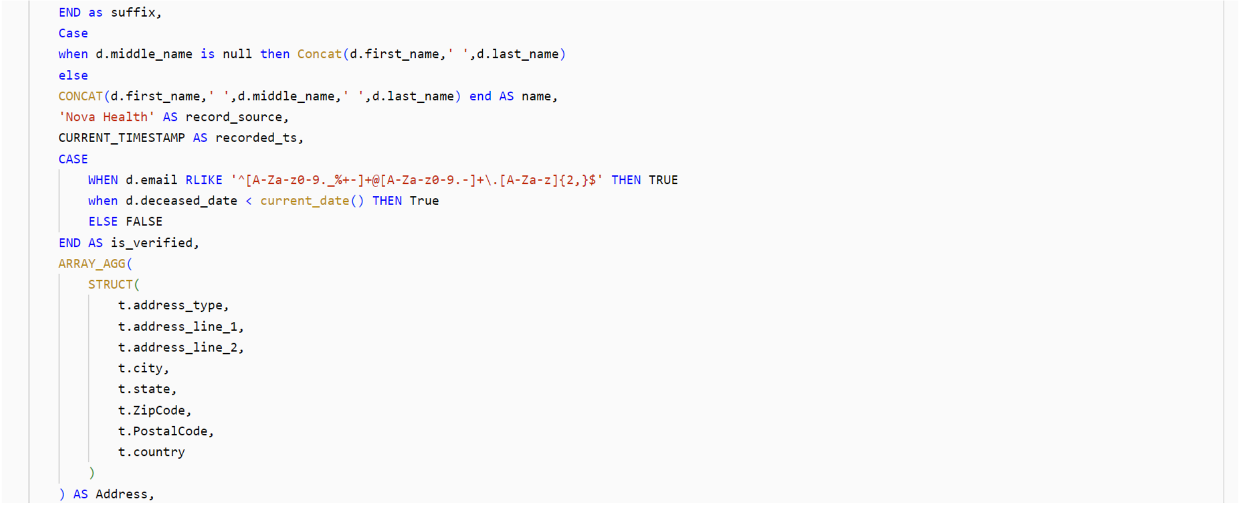
```
%sql
with temp_add as(
    SELECT id, address_type, address_line_1, address_line_2, city, state, CASE
    WHEN POSITION('-' IN zipcode) > 0 THEN SPLIT_PART(zipcode, '-', 1)
    WHEN LENGTH(zipcode) = 5 THEN zipcode
  END AS ZipCode,
  CASE
    WHEN POSITION('-' IN zipcode) > 0 THEN SPLIT_PART(zipcode, '-', 2)
    WHEN LENGTH(zipcode) = 4 THEN zipcode
    ELSE NULL
  END AS PostalCode,
  "USA" as country
FROM
  address_info
),
```



```
temp_date as(
    select id, date_of_birth, SUBSTRING(date_of_birth, -4, 4) AS year_of_birth, deceased_date AS deceased_date,
    CASE
        WHEN deceased_date IS NOT NULL THEN TRUE
        ELSE FALSE
    END AS deceased_ind,
    CASE
        WHEN deceased_date IS NOT NULL AND date_of_birth IS NOT NULL
        THEN CAST(SUBSTRING(deceased_date, -4, 4) AS INT) - CAST(SUBSTRING(date_of_birth, -4, 4) AS INT)
        ELSE NULL
    END AS deceased_age
    from detail_info
)
```

```
insert into final_result3
SELECT
    d.id AS source_id,
    h.insurer_id AS subscribe_id,
    d.first_name AS first_name,
    d.middle_name AS middle_name,
    d.last_name AS last_name,
    CASE
    WHEN (d.gender = "F" and (d.marital_status = "Married" or d.marital_status = "Widowed")) THEN "Mrs."
    WHEN d.gender = "F" and d.marital_status = "Single" THEN "Miss"
    WHEN d.gender = "M" THEN "Mr."
    END as prefix,

    case
    WHEN d.job_role like "%Nurse%" THEN "RN"
    when d.job_role like "%Doctor%" then "Dr."
    when d.job_role like "%Professor%" then "Prof."
    when d.job_role like "%VP%" then "VP"
    when d.job_role = "Clinical Specialist" then "CS"
    END as suffix,
```



```
    END as suffix,
    Case
    when d.middle_name is null then Concat(d.first_name,' ',d.last_name)
    else
    CONCAT(d.first_name,' ',d.middle_name,' ',d.last_name) end AS name,
    'Nova Health' AS record_source,
    CURRENT_TIMESTAMP AS recorded_ts,
    CASE
        WHEN d.email RLIKE '^[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,}$' THEN TRUE
        when d.deceased_date < current_date() THEN True
        ELSE FALSE
    END AS is_verified,
    ARRAY_AGG(
        STRUCT(
            t.address_type,
            t.address_line_1,
            t.address_line_2,
            t.city,
            t.state,
            t.ZipCode,
            t.PostalCode,
            t.country
        )
    ) AS Address,
    ARRAY_AGG(STRUCT(c.phone,c.usage_type)) AS phones,
    case
        when d.email RLIKE '^[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,}$' THEN d.email
        else
            "Invalid Email"
    end AS email,
    FALSE as privacy_preference,
    d.ssn AS national_id,
    d.gender AS gender,
    d.marital_status AS marital_status,
    td.date_of_birth AS date_of_birth,
    td.year_of_birth as year_of_birth,
    td.deceased_ind as deceased_ind,
    td.deceased_age as deceased_age,
    td.deceased_date AS deceased_date,
    struct(d.spoken_language_1, d.spoken_language_2) as Languages,
    Struct(
        d.first_name,
        d.job_role,
        CASE WHEN d.job_hiredate IS NULL THEN 'Inactive' ELSE 'Active' END as Employment_status,
        d.job_hiredate
    ) AS employment,
     MAP('relationship', h.relationship) AS additional_source_value
FROM header_info AS h
LEFT JOIN detail_info AS d ON h.id = d.id
LEFT JOIN contact_info AS c ON d.id = c.id
LEFT JOIN temp_add as t on d.id = t.id
left join temp_date as td on h.id = td.id
GROUP BY
    d.id, h.insurer_id, d.first_name, d.middle_name, d.last_name, d.ssn, d.gender, td.date_of_birth, td.year_of_birth, d.spoken_language_1, d.
    spoken_language_2, d.job_role, d.email, d.marital_status, d.deceased_date, td.deceased_date,td.deceased_ind, td.deceased_age, d.job_hiredate, d.company, d.
    religion, h.relationship;
```

Table

	num_affected_rows	num_inserted_rows
1	1500	1500

1 row

Final Table Result

```
select * from final_result3
```

Table

	source_id	subscriber_id	first_name	middle_name	last_name	prefix	suffix	name	record_source
1	70001	40184	Hettie	null	Keenlayside	Mrs.	CS	Hettie Keenlayside	Nova Health
2	70002	40092	Reade	null	Laverenz	Mr.	null	Reade Laverenz	Nova Health
3	70003	40233	Minnnie	null	Baack	Mrs.	null	Minnnie Baack	Nova Health
4	70004	40058	Tana	Agata	Aiken	null	VP	Tana Agata Aiken	Nova Health
5	70005	40088	Cyndia	null	Tolomelli	null	null	Cyndia Tolomelli	Nova Health