SQL-Focused

You are building a database for ACME website to show users' skills. Each user can have multiple skills (Python, Java, SQL, NodeJS, etc…). You need to design the database schema and write a SQL query to find users that have at least 5 of the top 10 skills.

```
SELECT u.user_id, u.name
FROM User_data u
INNER JOIN UserSkills us ON u.user_id = us.user_id
INNER JOIN (
    SELECT skill_id
    FROM Skills
    ORDER BY skill_name DESC
    LIMIT 10
) top_skills ON us.skill_id = top_skills.skill_id
GROUP BY u.user_id, u.name
HAVING COUNT(*) >= 5;
```

Questions:

● Can you explain the performance of your query?

- The performance of the query depends on the size of the UserSkills and Skills tables, as well as the available indexes.
- It retrieves the top 10 skills based on skill_name and then joins the User and UserSkills tables to find users with at least 5 of those skills
- The performance can be impacted if the tables have a large number of records or if the necessary indexes are not in place.

● How can you improve the query performance?

- Index the skill_name column in the Skills table for efficient sorting.
- Regularly analyze and update statistics to help the query optimizer make informed decisions.
- Consider implementing query caching or result caching if the query is frequently executed with the same parameters.

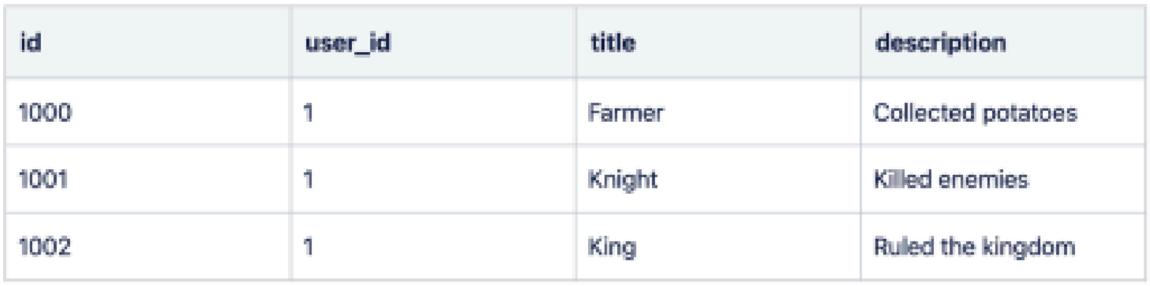
● Can you suggest different indexes or any other ways to optimize the query execution time?

- Index the skill_name and name column in tables.
- Apart from indexing and query optimization, consider denormalizing the data to avoid joins and improve query performance.
- Store a denormalized list of skills directly in the User table, allowing for faster retrieval without the need for joins.

An example task that we can give to candidates to assess their ability to write queries using window functions
Task: You have a table that contains information about user work experience.
The table has the following columns:

- id (integer)
- user_id (integer)
- title (text)
- description (text)
- date_added (date)
- date_edited (date)

You need to write a SQL query to get only 2nd work experience, for example user King Arthur has experience:



id	user_id	title	description
1000	1	Farmer	Collected potatoes
1001	1	Knight	Killed enemies
1002	1	King	Ruled the kingdom

The result of the query should be a record with id == 1001

```
SELECT *
FROM (
SELECT *, ROW_NUMBER() OVER (PARTITION BY user_id ORDER BY date_added) AS rn
FROM WorkExperience
) AS subquery
WHERE rn = 2 AND user_id = <user_id>;
```

Questions:

- Can you explain the performance of your query?

  - The performance depends on the size of the WorkExperience table and the availability of indexes.
  - The query uses the ROW_NUMBER() window function to assign a row number to each work experience based on the date_added column. It then filters the result to fetch the row with rn = 2.
  - The performance can be impacted if the table has a large number of records or if the necessary indexes are not in place.

- How can you improve the query performance?

  - Ensure proper indexing on the user_id and date_added columns in the WorkExperience table to optimize sorting and partitioning.
  - Regularly analyze and update statistics to help the query optimizer make informed decisions.
  - Consider implementing query caching or result caching if the query is frequently executed with the same parameters.

- Can you suggest any other approach to solve this problem?

```
SELECT *
FROM WorkExperience
WHERE user_id = <user_id>
ORDER BY date_added
LIMIT 1 OFFSET 1;
```

PySpark

Environment: Databricks or EMR cluster Requirements: PySpark, Delta Lake, AWS S3

QUESTIONS:
Analyzing Customer Reviews (40 minutes) You have been provided with a large dataset of customer reviews in Delta Lake format on AWS S3: s3://your-bucket/customer_reviews/

The dataset contains a mixture of structured and unstructured data. Each line in the dataset represents a customer review, and has the following format in a set of delta files: product_id, user_id, user_name, rating, review_date, review_title, review_text

Your task is to write a PySpark script that reads this dataset and performs the following analysis: However, due to the presence of messy data, some lines may contain extra or missing fields,

improperly formatted dates, or special characters.
1. Create a table (format choice is yours) with the following columns: uuid (String): Unique identifier for each record, generated based on product_id, user_id, and the Unix timestamp of the review_date product_id (String): Unique identifier for each product user_id (String): Unique identifier for each user user_name (String): Name of the user rating (Float): Rating given by the user, ranging from 1.0 to 5.0 review_date (Date): Date of the review submission review_title (String): Title of the review review_text (String): Text of the review

2. Read the dataset and validate the data by: Validating each column using regex patterns, as applicable Removing any special characters from the user_name, review_title, and review_text columns Filtering out rows with an incorrect number of fields. If missing more than 40% of data. Converting improperly formatted dates in the review_date column to a standard date format (e.g., "yyyy-MM-dd") Alerting if a row does not match the regex patterns or contains an improperly formatted date (only if you dont clean the data col) Remove columns if column is missing more than 90% of data.

3. Calculate the average rating for each product.

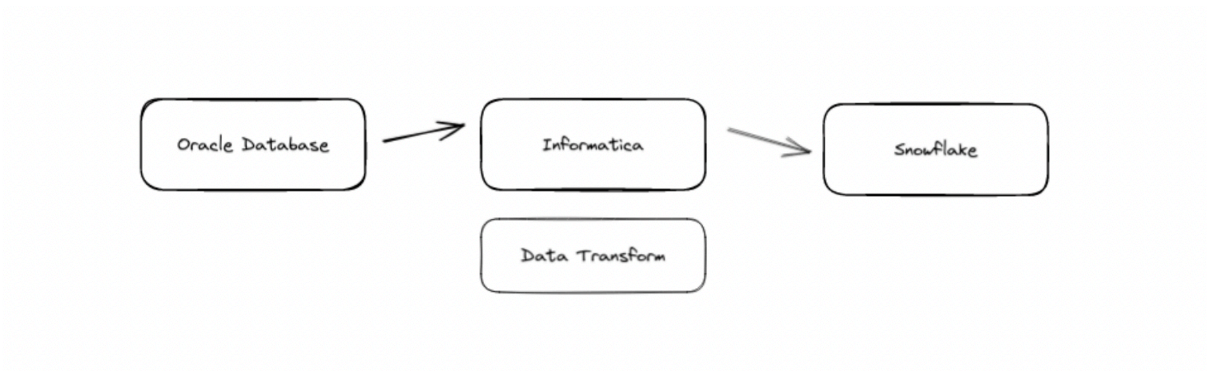
4. Find the top 10 products with the highest average ratings.

5. Identify the top three users with the most reviews submitted. Please use appropriate PySpark APIs and follow best practices for code readability and optimization. Ensure that your solution is optimized for processing large datasets (100 TBs) by leveraging big data best practices.

```
from pyspark.sql import SparkSession
from pyspark.sql.functions import col, udf, from_unixtime, current_date
from pyspark.sql.types import StringType, FloatType, DateType
import re
# Create SparkSession
spark = SparkSession.builder.appName("CustomerReviewsAnalysis").getOrCreate()

# Step 1: Create a table with the specified columns
spark.sql("""
    CREATE TABLE IF NOT EXISTS customer_reviews (
        uuid STRING,
        product_id STRING,
        user_id STRING,
        user_name STRING,
        rating FLOAT,
        review_date DATE,
        review_title STRING,
        review_text STRING
    )
    USING DELTA
    PARTITIONED BY (product_id)
    LOCATION 's3://your-bucket/customer_reviews/'
""")
# Step 2: Read the dataset and validate the data
df = spark.read.format("delta").load("s3://your-bucket/customer_reviews/")

# Define regex patterns for validation
uuid_pattern = re.compile(r"^\w{8}-\w{4}-\w{4}-\w{4}-\w{12}$")
date_pattern = re.compile(r"^\d{4}-\d{2}-\d{2}$")


# UDF for generating UUID based on product_id, user_id, and review_date
generate_uuid = udf(lambda product_id, user_id, review_date: str(hash(product_id +
user_id + str(review_date))))

# UDF for cleaning special characters from strings
clean_string = udf(lambda s: re.sub(r"[^\w\s]", "", s) if s is not None else None)

# UDF for converting improperly formatted dates to standard date format
convert_date = udf(lambda d: None if d is None else d if date_pattern.match(d) else
None, DateType())


# Apply data validation and cleaning
df = df.withColumn("uuid", generate_uuid(col("product_id"), col("user_id"),
col("review_date"))) \
    .withColumn("user_name", clean_string(col("user_name"))) \
    .withColumn("review_title", clean_string(col("review_title"))) \
    .withColumn("review_text", clean_string(col("review_text"))) \
    .withColumn("review_date", convert_date(col("review_date"))) \
    .filter(col("uuid").rlike(uuid_pattern)) \
    .filter(col("rating").between(1.0, 5.0)) \
    .filter(col("product_id").isNotNull() & col("user_id").isNotNull() &
col("user_name").isNotNull() &
            col("rating").isNotNull() & col("review_date").isNotNull() &
col("review_title").isNotNull() &
            col("review_text").isNotNull()) \
    .filter(col("user_name") != "") \
    .filter(col("review_title") != "") \
    .filter(col("review_text") != "") \
    .filter(col("review_date").isNotNull()) \
    .filter(col("review_date") <= current_date())
```

```
# Step 3: Calculate average rating for each product
average_rating = df.groupBy("product_id").agg({"rating": "avg"})

# Step 4: Find top 10 products with highest average ratings
top_products = average_rating.orderBy(col("avg(rating)").desc()).limit(10)

# Step 5: Identify top three users with most reviews submitted
top_users = df.groupBy("user_id").count().orderBy(col("count").desc()).limit(3)

# Display the results
top_products.show()
top_users.show()

# Stop the SparkSession
spark
```

Data Migration



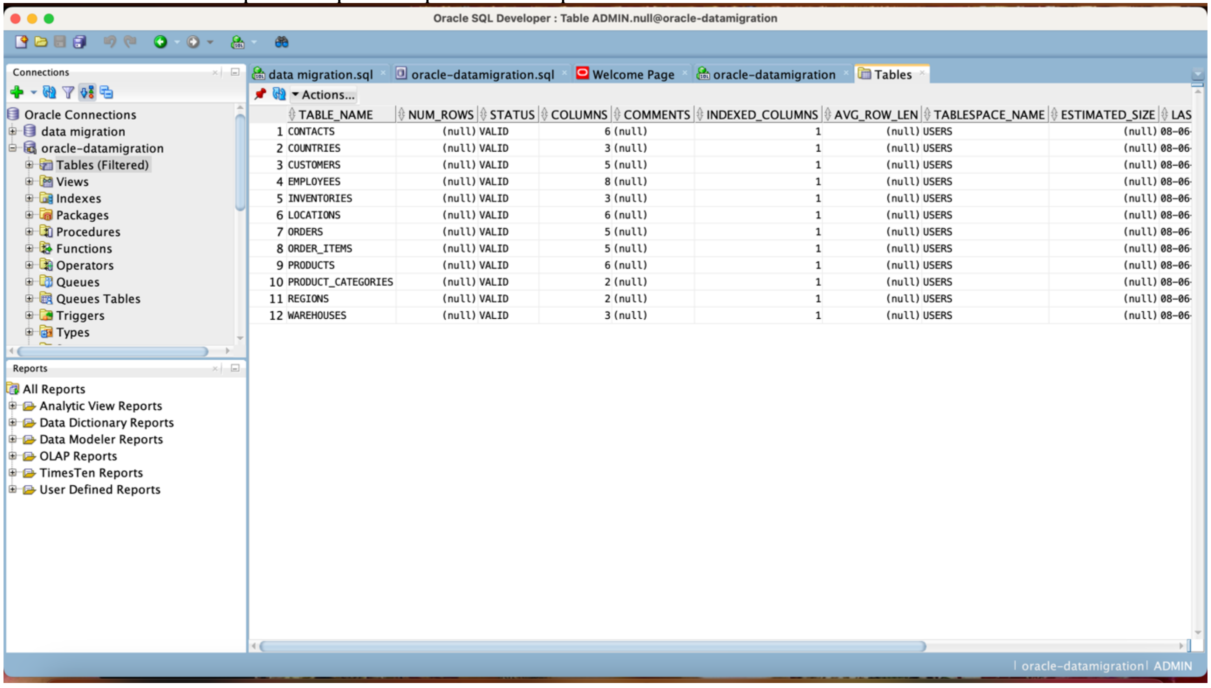
You need to perform the migration of an Oracle database to Snowflake. Your goal is to efficiently transfer the data from Oracle to Snowflake and ensure that the data is stored correctly in the new environment.

Considerations

- Import the database from the link below into your Oracle database
- Analysis of the Oracle database
- Schema and table mapping
- Extraction of data from Oracle
- Creation of tables in Snowflake
- Data loading into Snowflake using Informatica
- Migration of database logic and application code
- Testing and validation
- Documentation

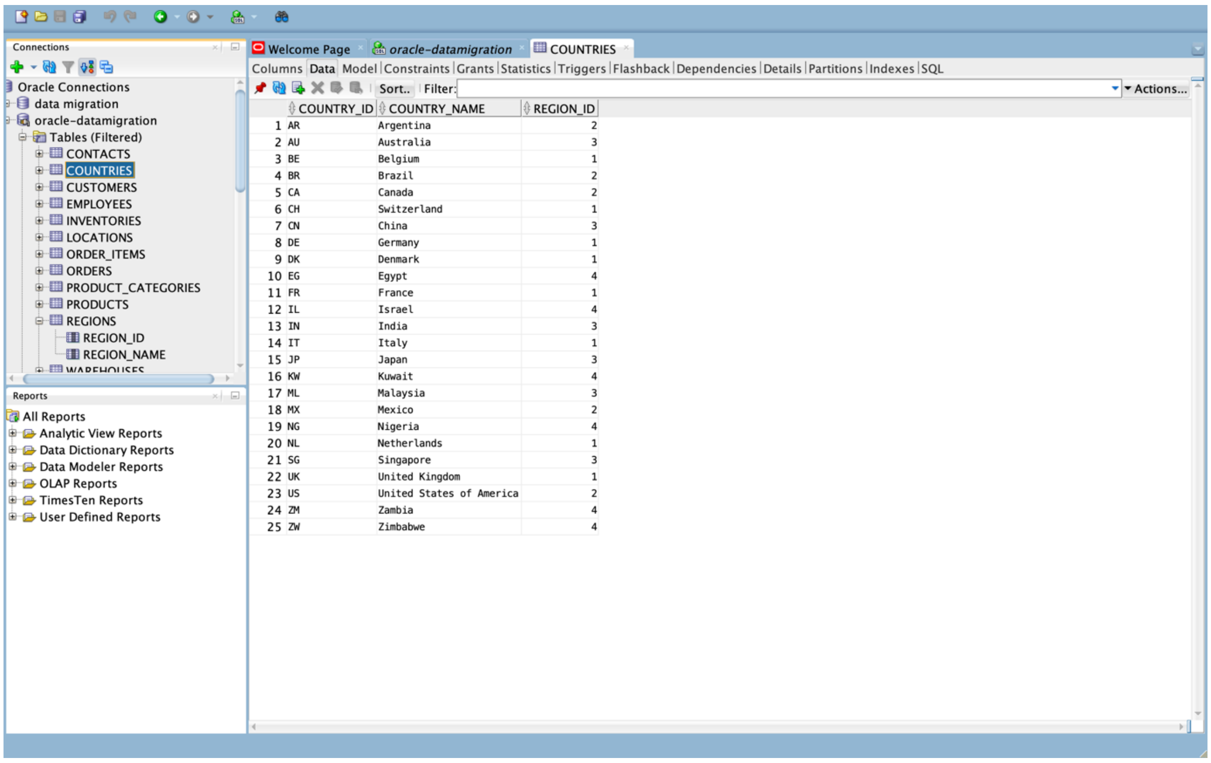
The provided code should not be a script, it should be able to reuse cross-projects or applications. Be sure that the code you sent is correctly tested and well documented. The database to be migrated
https://www.oracletutorial.com/wp-content/uploads/2019/01/oracle-sample-database.zip

- Import the database from the link below into your Oracle database
Obtain the Oracle setup and import scripts from the provided link.



	TABLE_NAME	NUM_ROWS	STATUS	COLUMNS	COMMENTS	INDEXED_COLUMNS	AVG_ROW_LEN	TABLESPACE_NAME	ESTIMATED_SIZE	LAS
1	CONTACTS	(null)	VALID	6	(null)	1	(null)	USERS	(null)	08-06-
2	COUNTRIES	(null)	VALID	3	(null)	1	(null)	USERS	(null)	08-06-
3	CUSTOMERS	(null)	VALID	5	(null)	1	(null)	USERS	(null)	08-06-
4	EMPLOYEES	(null)	VALID	8	(null)	1	(null)	USERS	(null)	08-06-
5	INVENTORIES	(null)	VALID	3	(null)	1	(null)	USERS	(null)	08-06-
6	LOCATIONS	(null)	VALID	6	(null)	1	(null)	USERS	(null)	08-06-
7	ORDERS	(null)	VALID	5	(null)	1	(null)	USERS	(null)	08-06-
8	ORDER_ITEMS	(null)	VALID	5	(null)	1	(null)	USERS	(null)	08-06-
9	PRODUCTS	(null)	VALID	6	(null)	1	(null)	USERS	(null)	08-06-
10	PRODUCT_CATEGORIES	(null)	VALID	2	(null)	1	(null)	USERS	(null)	08-06-
11	REGIONS	(null)	VALID	2	(null)	1	(null)	USERS	(null)	08-06-
12	WAREHOUSES	(null)	VALID	3	(null)	1	(null)	USERS	(null)	08-06-

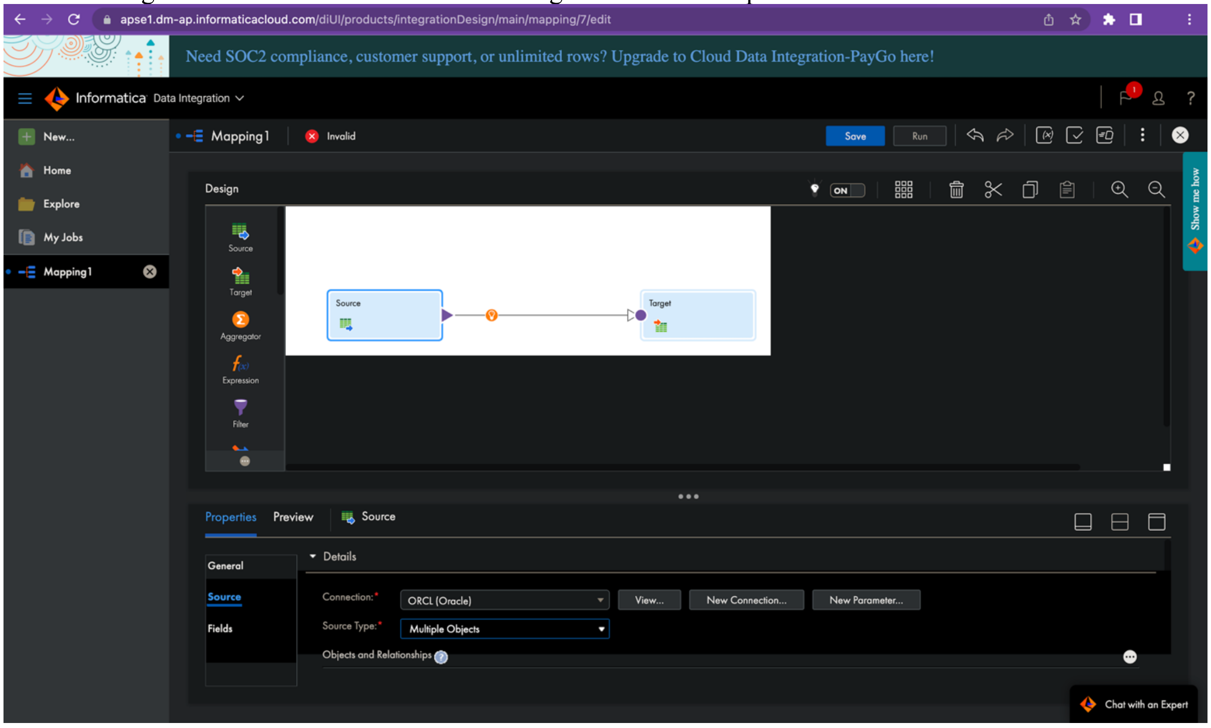
- Analysis of the Oracle database



	COUNTRY_ID	COUNTRY_NAME	REGION_ID
1	AR	Argentina	2
2	AU	Australia	3
3	BE	Belgium	1
4	BR	Brazil	2
5	CA	Canada	2
6	CH	Switzerland	1
7	CN	China	3
8	DE	Germany	1
9	DK	Denmark	1
10	EG	Egypt	4
11	FR	France	1
12	IL	Israel	4
13	IN	India	3
14	IT	Italy	1
15	JP	Japan	3
16	KW	Kuwait	4
17	ML	Malaysia	3
18	MX	Mexico	2
19	NG	Nigeria	4
20	NL	Netherlands	1
21	SG	Singapore	3
22	UK	United Kingdom	1
23	US	United States of America	2
24	ZM	Zambia	4
25	ZW	Zimbabwe	4

- Schema and table mapping

In Informatica, the procedure involves configuring both the oracle as a source and snowflake as a destination database and establishing a mapping that links the two.
We can change the source and destination according to our further requirements.



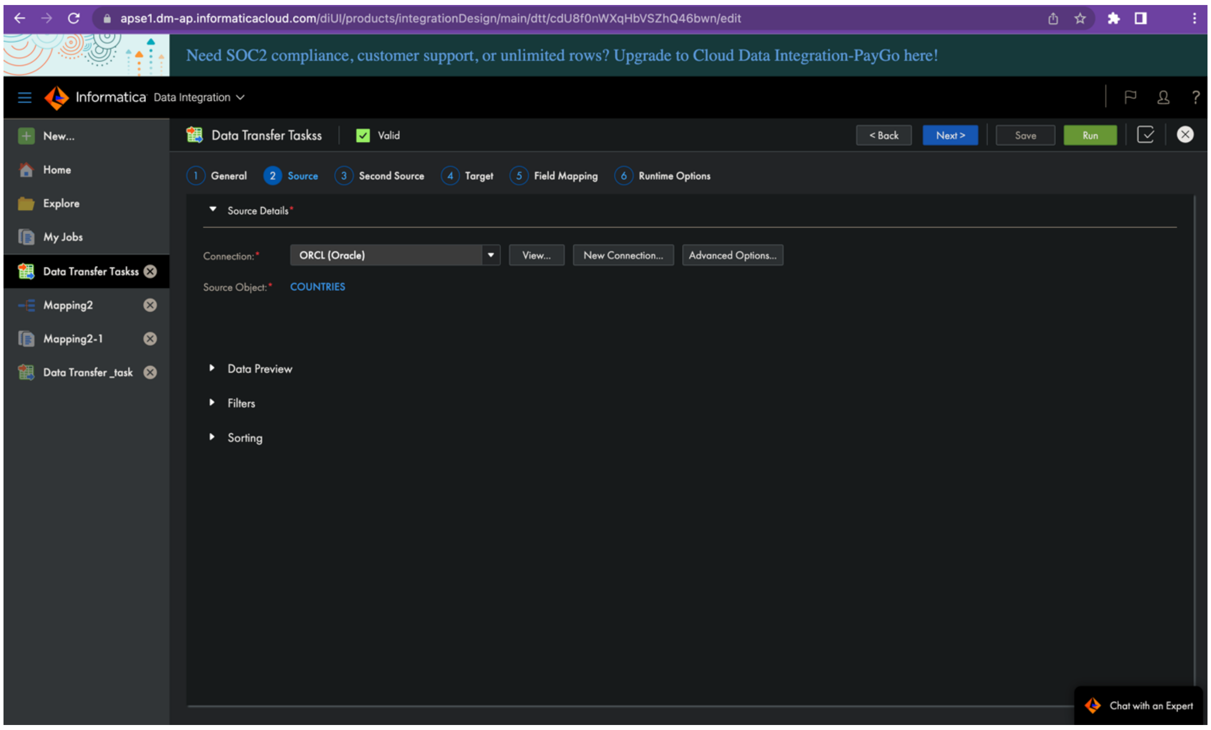
apse1.dm-ap.informaticacloud.com/diUI/products/integrationDesign/main/mapping/7/edit

Need SOC2 compliance, customer support, or unlimited rows? Upgrade to Cloud Data Integration-PayGo here!

Connection:* ORCL (Oracle)

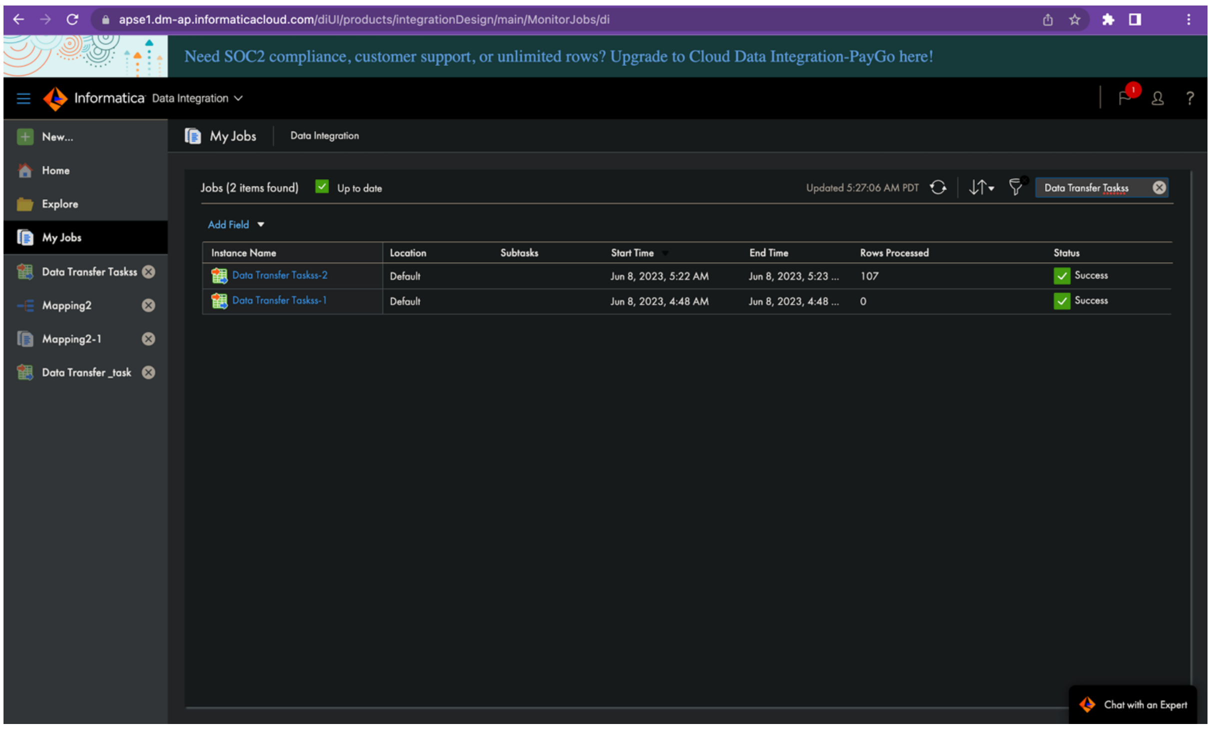
Source Type:* Multiple Objects

- Extraction of data from Oracle



Connection:* ORCL (Oracle)

Source Object:* COUNTRIES

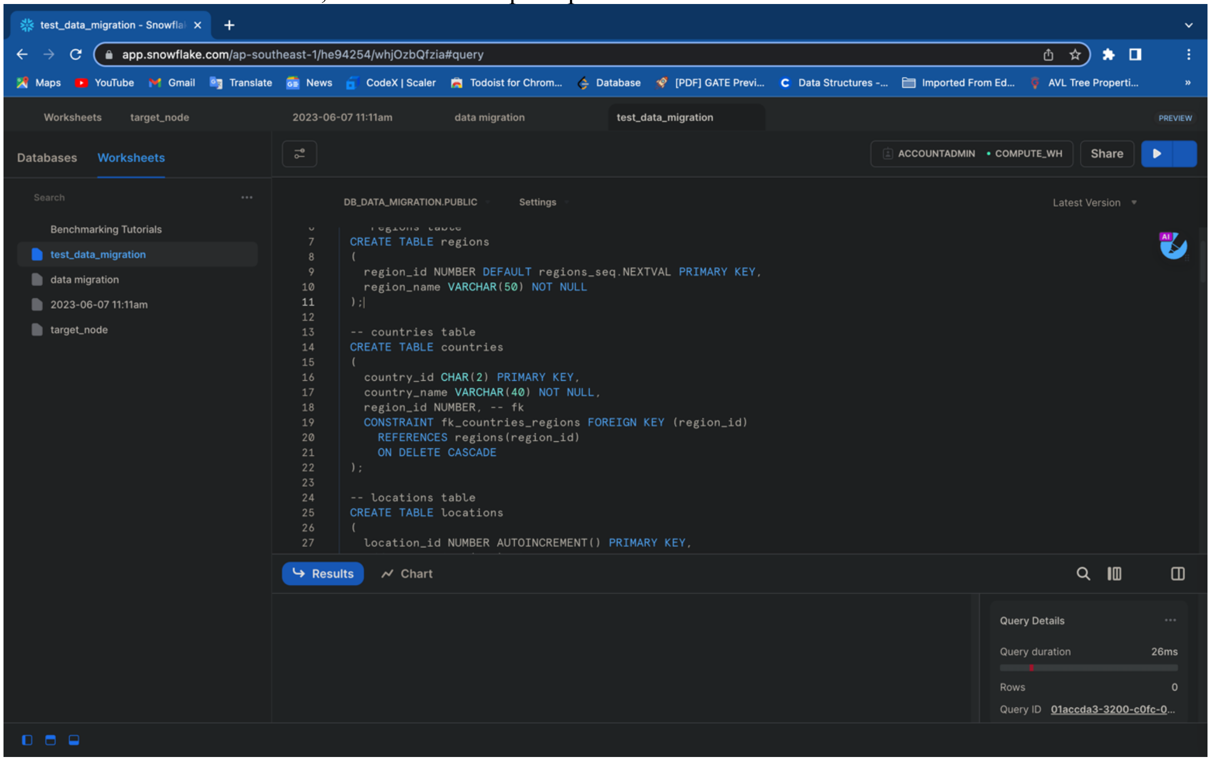

apse1.dm-ap.informaticacloud.com/diUI/products/integrationDesign/main/MonitorJobs/di
Need SOC2 compliance, customer support, or unlimited rows? Upgrade to Cloud Data Integration-PayGo here!
Informatica Data Integration
New...
Home
Explore
My Jobs
Data Transfer Tasks
Mapping2
Mapping2-1
Data Transfer _task
My Jobs
Data Integration
Add Field
Data Transfer Taskss
Chat with an Expert

Jobs (2 items found) Up to date Updated 5:27:06 AM PDT

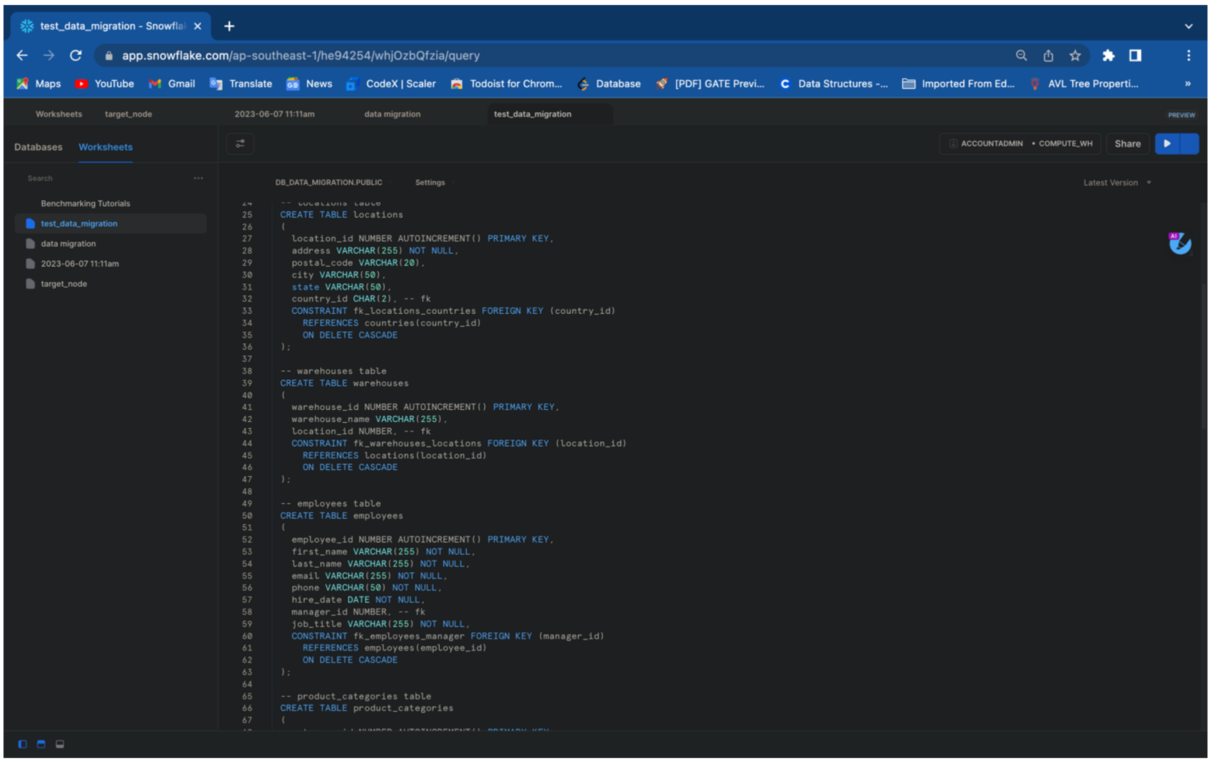
Instance Name	Location	Subtasks	Start Time	End Time	Rows Processed	Status
Data Transfer Taskss-2	Default		Jun 8, 2023, 5:22 AM	Jun 8, 2023, 5:23 ...	107	✓ Success
Data Transfer Taskss-1	Default		Jun 8, 2023, 4:48 AM	Jun 8, 2023, 4:48 ...	0	✓ Success

- Creation of tables in Snowflake

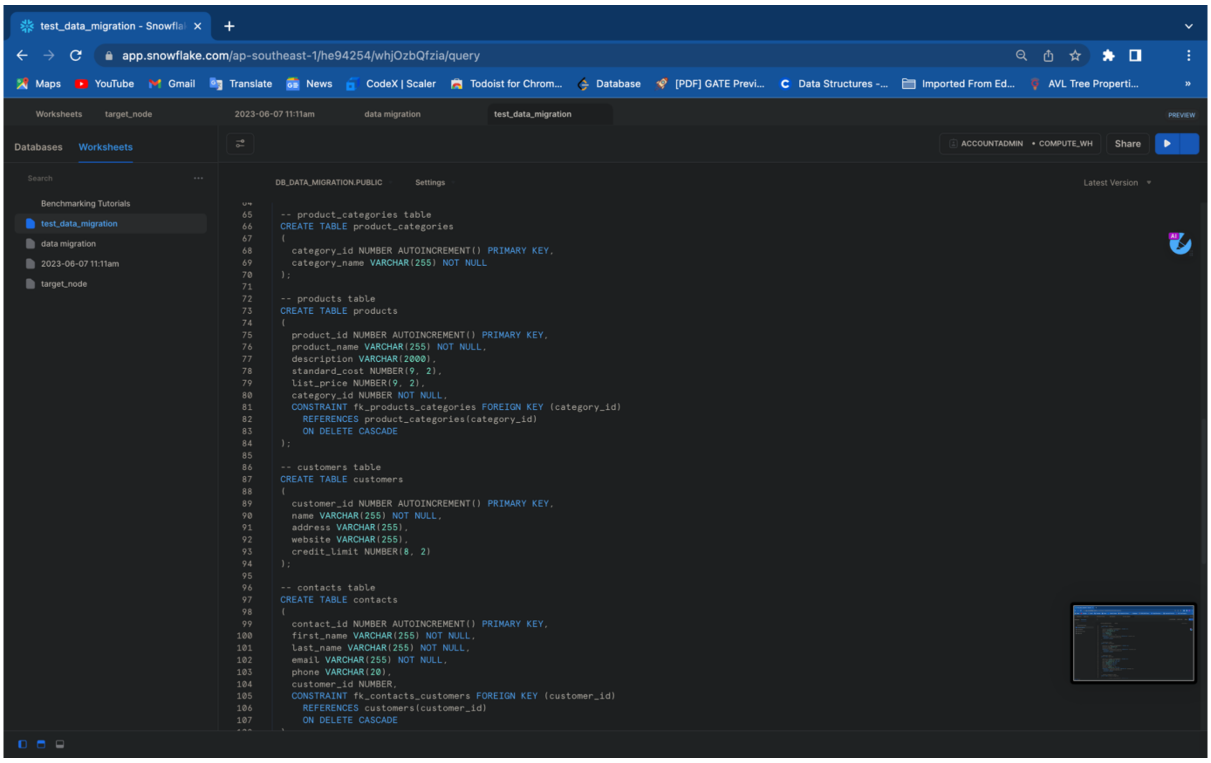
To create tables in Snowflake, I wrote a snowsql script.



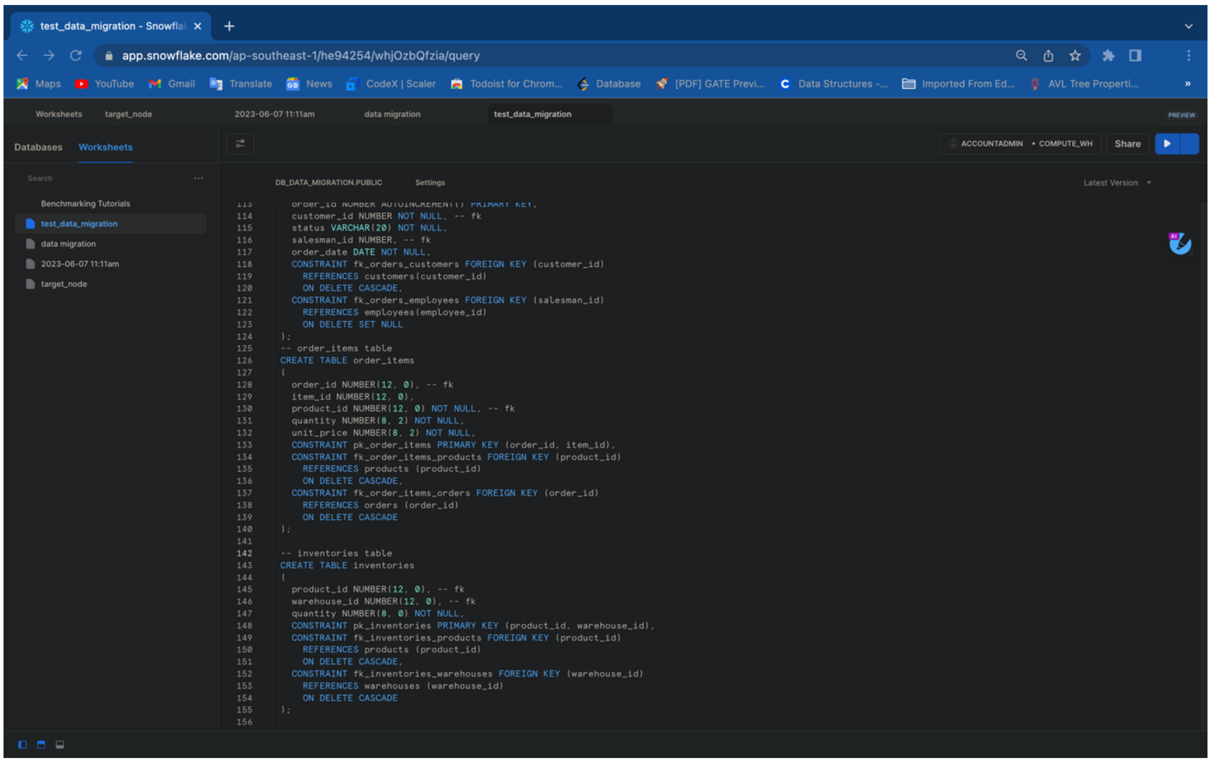
```sql
CREATE TABLE regions
(
  region_id NUMBER DEFAULT regions_seq.NEXTVAL PRIMARY KEY,
  region_name VARCHAR(50) NOT NULL
);

-- countries table
CREATE TABLE countries
(
  country_id CHAR(2) PRIMARY KEY,
  country_name VARCHAR(40) NOT NULL,
  region_id NUMBER, -- fk
  CONSTRAINT fk_countries_regions FOREIGN KEY (region_id)
    REFERENCES regions(region_id)
    ON DELETE CASCADE
);

-- locations table
CREATE TABLE locations
(
  location_id NUMBER AUTOINCREMENT() PRIMARY KEY,
```



```sql
CREATE TABLE locations
(
  location_id NUMBER AUTOINCREMENT() PRIMARY KEY,
  address VARCHAR(255) NOT NULL,
  postal_code VARCHAR(20),
  city VARCHAR(50),
  state VARCHAR(50),
  country_id CHAR(2), -- fk
  CONSTRAINT fk_locations_countries FOREIGN KEY (country_id)
    REFERENCES countries(country_id)
    ON DELETE CASCADE
);

-- warehouses table
CREATE TABLE warehouses
(
  warehouse_id NUMBER AUTOINCREMENT() PRIMARY KEY,
  warehouse_name VARCHAR(255),
  location_id NUMBER, -- fk
  CONSTRAINT fk_warehouses_locations FOREIGN KEY (location_id)
    REFERENCES locations(location_id)
    ON DELETE CASCADE
);

-- employees table
CREATE TABLE employees
(
  employee_id NUMBER AUTOINCREMENT() PRIMARY KEY,
  first_name VARCHAR(255) NOT NULL,
  last_name VARCHAR(255) NOT NULL,
  email VARCHAR(255) NOT NULL,
  phone VARCHAR(50) NOT NULL,
  hire_date DATE NOT NULL,
  manager_id NUMBER, -- fk
  job_title VARCHAR(255) NOT NULL,
  CONSTRAINT fk_employees_manager FOREIGN KEY (manager_id)
    REFERENCES employees(employee_id)
    ON DELETE CASCADE
);

-- product_categories table
CREATE TABLE product_categories
(
```

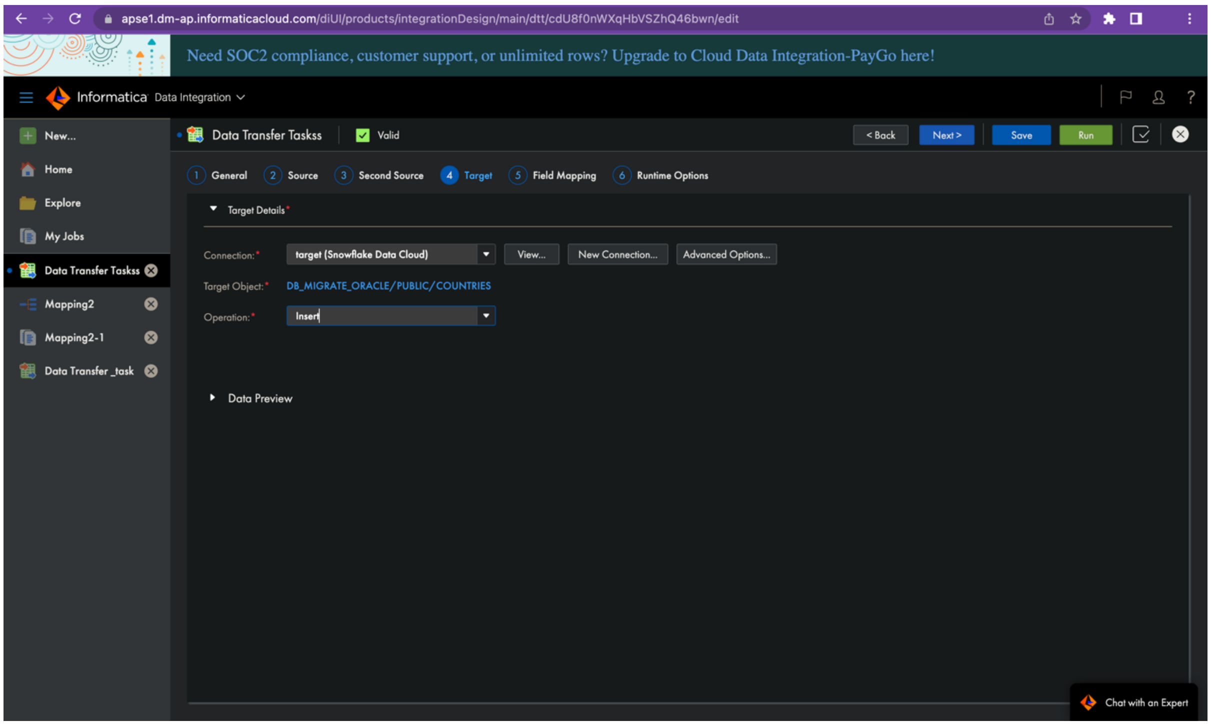


```sql
-- product_categories table
CREATE TABLE product_categories
(
  category_id NUMBER AUTOINCREMENT() PRIMARY KEY,
  category_name VARCHAR(255) NOT NULL
);

-- products table
CREATE TABLE products
(
  product_id NUMBER AUTOINCREMENT() PRIMARY KEY,
  product_name VARCHAR(255) NOT NULL,
  description VARCHAR(2000),
  standard_cost NUMBER(9, 2),
  list_price NUMBER(9, 2),
  category_id NUMBER NOT NULL,
  CONSTRAINT fk_products_categories FOREIGN KEY (category_id)
    REFERENCES product_categories(category_id)
    ON DELETE CASCADE
);

-- customers table
CREATE TABLE customers
(
  customer_id NUMBER AUTOINCREMENT() PRIMARY KEY,
  name VARCHAR(255) NOT NULL,
  address VARCHAR(255),
  website VARCHAR(255),
  credit_limit NUMBER(8, 2)
);

-- contacts table
CREATE TABLE contacts
(
  contact_id NUMBER AUTOINCREMENT() PRIMARY KEY,
  first_name VARCHAR(255) NOT NULL,
  last_name VARCHAR(255) NOT NULL,
  email VARCHAR(255) NOT NULL,
  phone VARCHAR(20),
  customer_id NUMBER,
  CONSTRAINT fk_contacts_customers FOREIGN KEY (customer_id)
    REFERENCES customers(customer_id)
    ON DELETE CASCADE
```



```sql
  order_id NUMBER AUTOINCREMENT() PRIMARY KEY,
  customer_id NUMBER NOT NULL, -- fk
  status VARCHAR(20) NOT NULL,
  salesman_id NUMBER, -- fk
  order_date DATE NOT NULL,
  CONSTRAINT fk_orders_customers FOREIGN KEY (customer_id)
    REFERENCES customers(customer_id)
    ON DELETE CASCADE,
  CONSTRAINT fk_orders_employees FOREIGN KEY (salesman_id)
    REFERENCES employees(employee_id)
    ON DELETE SET NULL
);
-- order_items table
CREATE TABLE order_items
(
  order_id NUMBER(12, 0), -- fk
  item_id NUMBER(12, 0),
  product_id NUMBER(12, 0) NOT NULL, -- fk
  quantity NUMBER(8, 2) NOT NULL,
  unit_price NUMBER(8, 2) NOT NULL,
  CONSTRAINT pk_order_items PRIMARY KEY (order_id, item_id),
  CONSTRAINT fk_order_items_products FOREIGN KEY (product_id)
    REFERENCES products (product_id)
    ON DELETE CASCADE,
  CONSTRAINT fk_order_items_orders FOREIGN KEY (order_id)
    REFERENCES orders (order_id)
    ON DELETE CASCADE
);

-- inventories table
CREATE TABLE inventories
(
  product_id NUMBER(12, 0), -- fk
  warehouse_id NUMBER(12, 0), -- fk
  quantity NUMBER(8, 0) NOT NULL,
  CONSTRAINT pk_inventories PRIMARY KEY (product_id, warehouse_id),
  CONSTRAINT fk_inventories_products FOREIGN KEY (product_id)
    REFERENCES products (product_id)
    ON DELETE CASCADE,
  CONSTRAINT fk_inventories_warehouses FOREIGN KEY (warehouse_id)
    REFERENCES warehouses (warehouse_id)
    ON DELETE CASCADE
);
```

- Data loading into Snowflake using Informatica

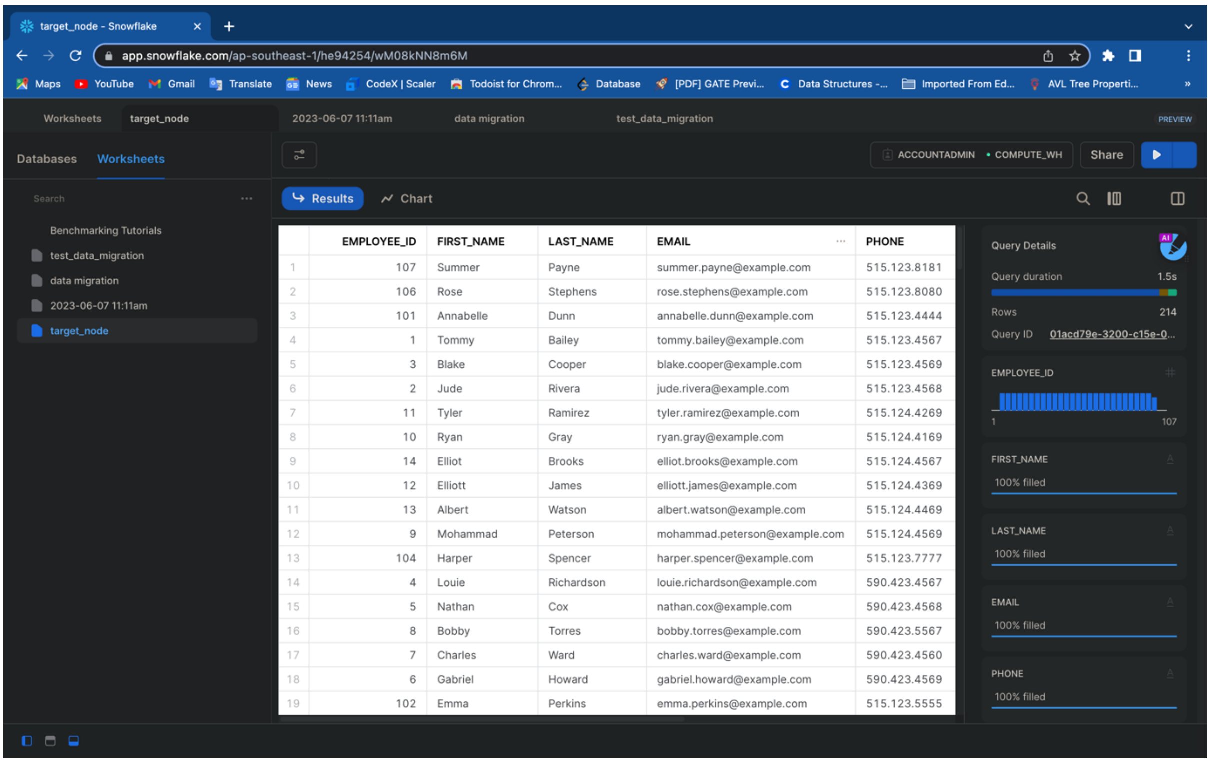

apse1.dm-ap.informaticacloud.com/diUI/products/integrationDesign/main/dtt/cdU8f0nWXqHbVSZhQ46bwn/edit

Need SOC2 compliance, customer support, or unlimited rows? Upgrade to Cloud Data Integration-PayGo here!

Connection: target (Snowflake Data Cloud)

Target Object: DB_MIGRATE_ORACLE/PUBLIC/COUNTRIES

Operation: Insert


app.snowflake.com/ap-southeast-1/he94254/wM08kNN8m6M

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE
1	107	Summer	Payne	summer.payne@example.com	515.123.8181
2	106	Rose	Stephens	rose.stephens@example.com	515.123.8080
3	101	Annabelle	Dunn	annabelle.dunn@example.com	515.123.4444
4	1	Tommy	Bailey	tommy.bailey@example.com	515.123.4567
5	3	Blake	Cooper	blake.cooper@example.com	515.123.4569
6	2	Jude	Rivera	jude.rivera@example.com	515.123.4568
7	11	Tyler	Ramirez	tyler.ramirez@example.com	515.124.4269
8	10	Ryan	Gray	ryan.gray@example.com	515.124.4169
9	14	Elliot	Brooks	elliot.brooks@example.com	515.124.4567
10	12	Elliott	James	elliott.james@example.com	515.124.4369
11	13	Albert	Watson	albert.watson@example.com	515.124.4469
12	9	Mohammad	Peterson	mohammad.peterson@example.com	515.124.4569
13	104	Harper	Spencer	harper.spencer@example.com	515.123.7777
14	4	Louie	Richardson	louie.richardson@example.com	590.423.4567
15	5	Nathan	Cox	nathan.cox@example.com	590.423.4568
16	8	Bobby	Torres	bobby.torres@example.com	590.423.5567
17	7	Charles	Ward	charles.ward@example.com	590.423.4560
18	6	Gabriel	Howard	gabriel.howard@example.com	590.423.4569
19	102	Emma	Perkins	emma.perkins@example.com	515.123.5555

Query Details

Query duration 1.5s

Rows 214

Query ID 01acd79e-3200-c15e-0...

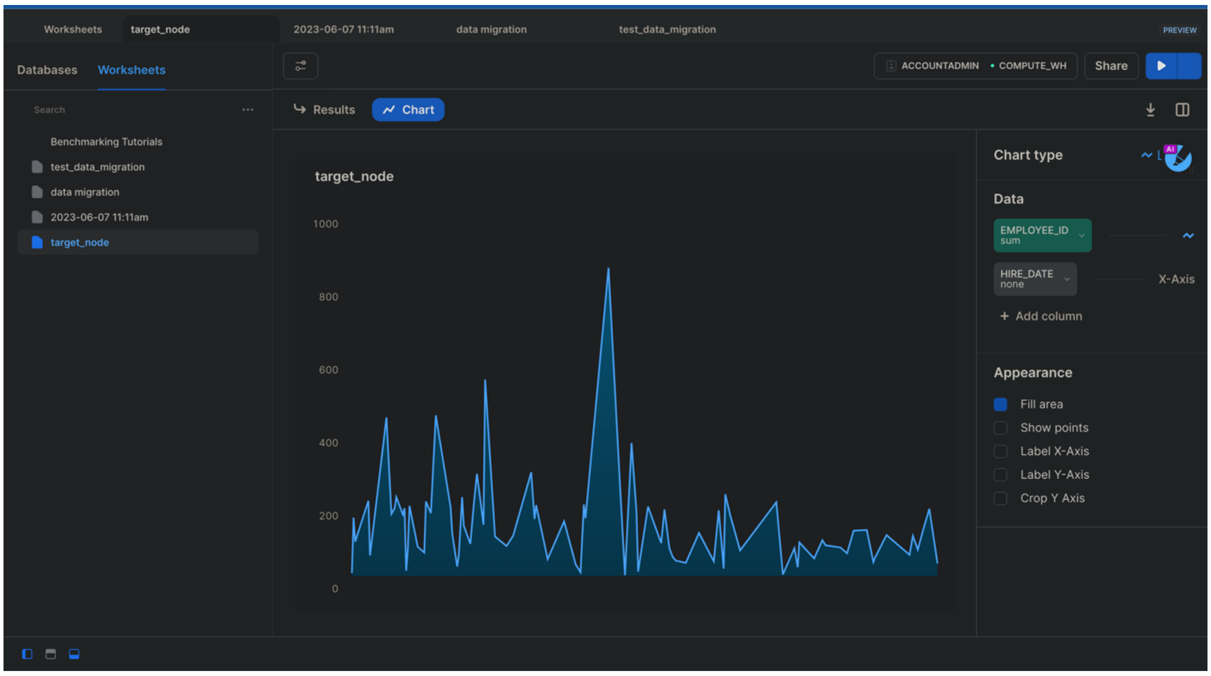
EMPLOYEE_ID 1 107

FIRST_NAME 100% filled

LAST_NAME 100% filled

EMAIL 100% filled

PHONE 100% filled

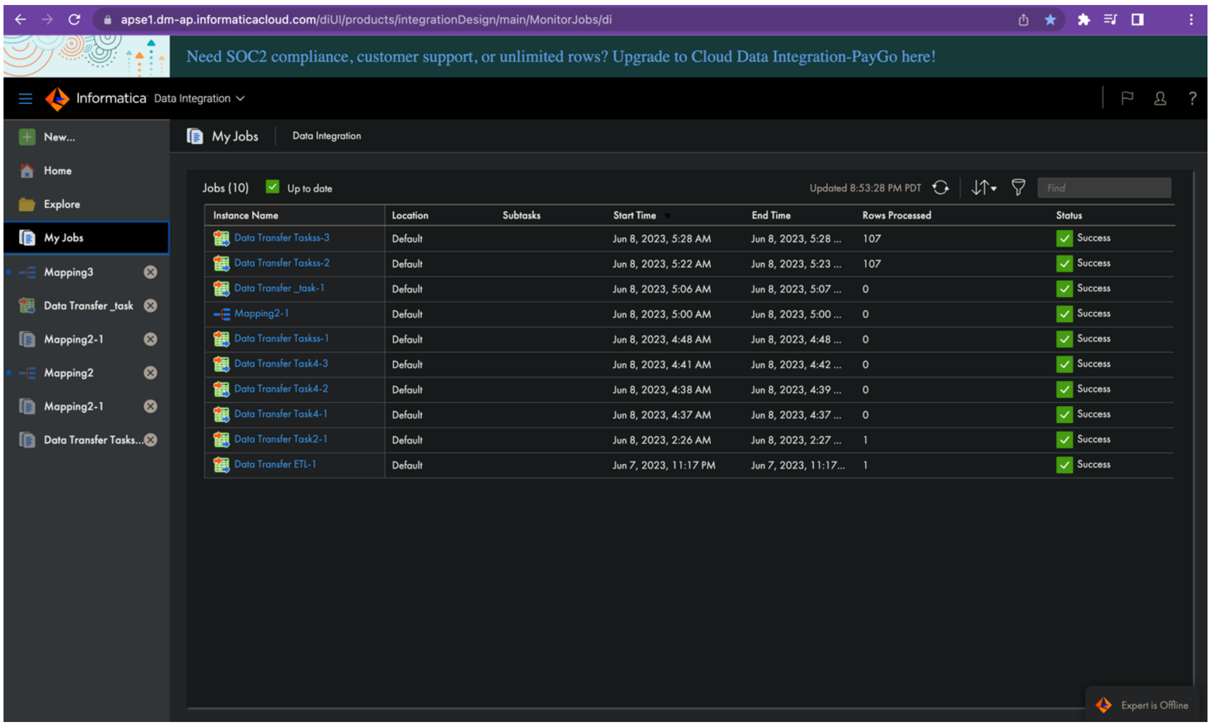


EMPLOYEE_ID sum

HIRE_DATE none — X-Axis

- [✓] Fill area
- [ ] Show points
- [ ] Label X-Axis
- [ ] Label Y-Axis
- [ ] Crop Y Axis

- Migration of database logic and application code

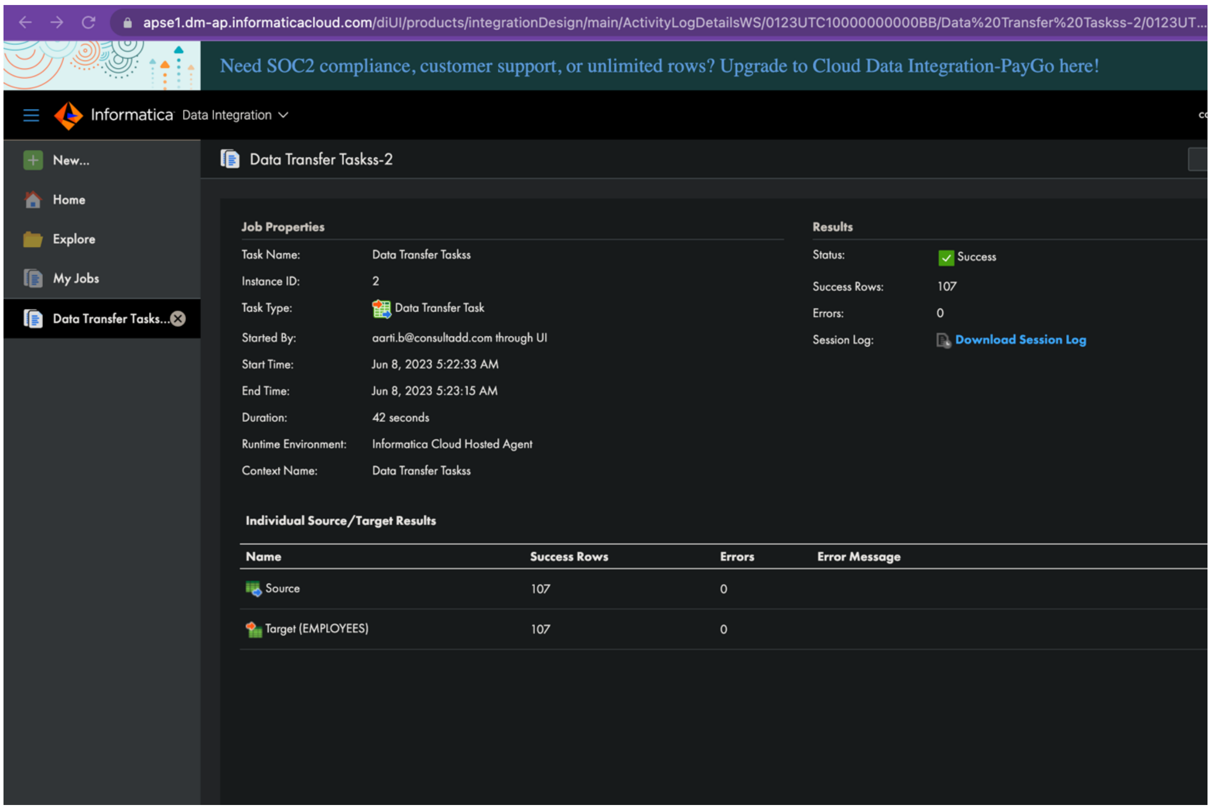


apse1.dm-ap.informaticacloud.com/diUI/products/integrationDesign/main/MonitorJobs/di

Need SOC2 compliance, customer support, or unlimited rows? Upgrade to Cloud Data Integration-PayGo here!

Jobs (10) ✓ Up to date Updated 8:53:28 PM PDT

Instance Name	Location	Subtasks	Start Time	End Time	Rows Processed	Status
Data Transfer Taskss-3	Default		Jun 8, 2023, 5:28 AM	Jun 8, 2023, 5:28 ...	107	✓ Success
Data Transfer Taskss-2	Default		Jun 8, 2023, 5:22 AM	Jun 8, 2023, 5:23 ...	107	✓ Success
Data Transfer _task-1	Default		Jun 8, 2023, 5:06 AM	Jun 8, 2023, 5:07 ...	0	✓ Success
Mapping2-1	Default		Jun 8, 2023, 5:00 AM	Jun 8, 2023, 5:00 ...	0	✓ Success
Data Transfer Taskss-1	Default		Jun 8, 2023, 4:48 AM	Jun 8, 2023, 4:48 ...	0	✓ Success
Data Transfer Task4-3	Default		Jun 8, 2023, 4:41 AM	Jun 8, 2023, 4:42 ...	0	✓ Success
Data Transfer Task4-2	Default		Jun 8, 2023, 4:38 AM	Jun 8, 2023, 4:39 ...	0	✓ Success
Data Transfer Task4-1	Default		Jun 8, 2023, 4:37 AM	Jun 8, 2023, 4:37 ...	0	✓ Success
Data Transfer Task2-1	Default		Jun 8, 2023, 2:26 AM	Jun 8, 2023, 2:27 ...	1	✓ Success
Data Transfer ETL-1	Default		Jun 7, 2023, 11:17 PM	Jun 7, 2023, 11:17...	1	✓ Success



apse1.dm-ap.informaticacloud.com/diUI/products/integrationDesign/main/ActivityLogDetailsWS/0123UTC10000000000BB/Data%20Transfer%20Taskss-2/0123UT...

Need SOC2 compliance, customer support, or unlimited rows? Upgrade to Cloud Data Integration-PayGo here!

Data Transfer Taskss-2

Job Properties

Task Name:	Data Transfer Taskss
Instance ID:	2
Task Type:	Data Transfer Task
Started By:	aarti.b@consultadd.com through UI
Start Time:	Jun 8, 2023 5:22:33 AM
End Time:	Jun 8, 2023 5:23:15 AM
Duration:	42 seconds
Runtime Environment:	Informatica Cloud Hosted Agent
Context Name:	Data Transfer Taskss

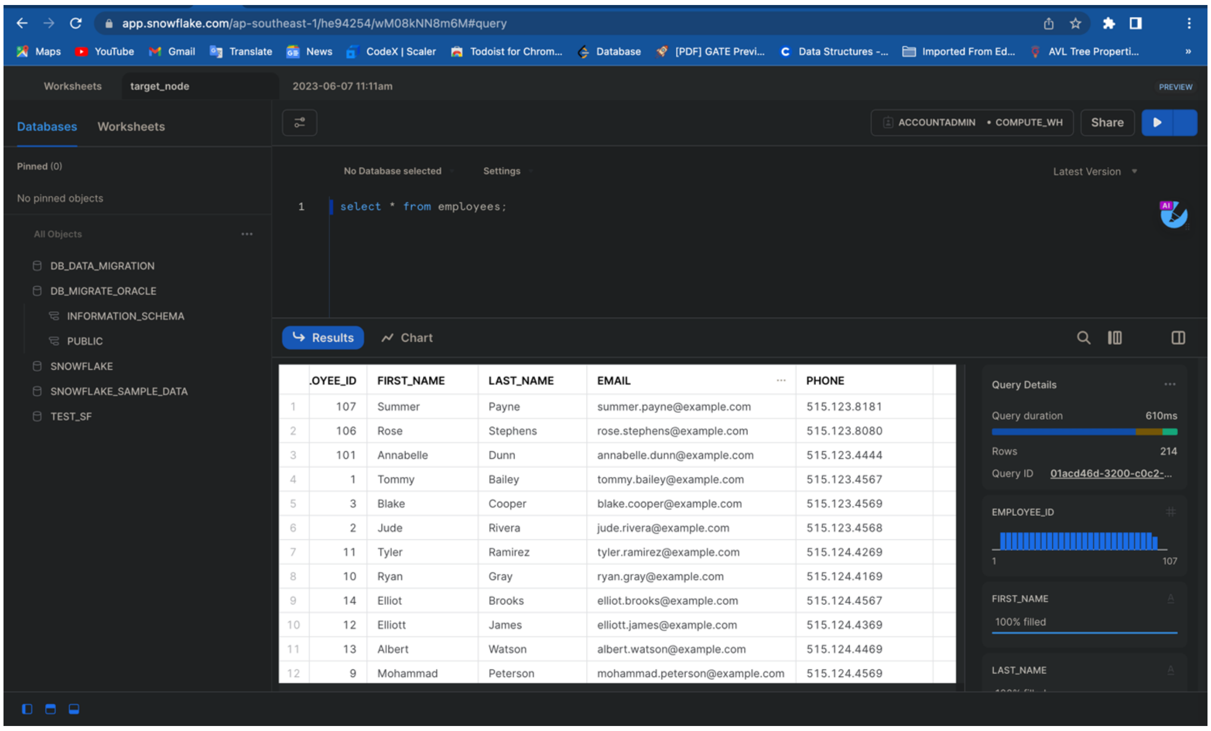
Results

Status:	✓ Success
Success Rows:	107
Errors:	0
Session Log:	Download Session Log

Individual Source/Target Results

Name	Success Rows	Errors	Error Message
Source	107	0	
Target (EMPLOYEES)	107	0	

- Testing and validation



```
select * from employees;
```

	.OYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE
1	107	Summer	Payne	summer.payne@example.com	515.123.8181
2	106	Rose	Stephens	rose.stephens@example.com	515.123.8080
3	101	Annabelle	Dunn	annabelle.dunn@example.com	515.123.4444
4	1	Tommy	Bailey	tommy.bailey@example.com	515.123.4567
5	3	Blake	Cooper	blake.cooper@example.com	515.123.4569
6	2	Jude	Rivera	jude.rivera@example.com	515.123.4568
7	11	Tyler	Ramirez	tyler.ramirez@example.com	515.124.4269
8	10	Ryan	Gray	ryan.gray@example.com	515.124.4169
9	14	Elliot	Brooks	elliot.brooks@example.com	515.124.4567
10	12	Elliott	James	elliott.james@example.com	515.124.4369
11	13	Albert	Watson	albert.watson@example.com	515.124.4469
12	9	Mohammad	Peterson	mohammad.peterson@example.com	515.124.4569

Query Details

Query duration	610ms
Rows	214
Query ID	01acd46d-3200-c0c2-...

EMPLOYEE_ID 1 107

FIRST_NAME 100% filled

LAST_NAME

Logs:

```
Agent Group Id: 025000000000002
Agent Group Name: Informatica Cloud Hosted Agent
Agent Id: 00800000000003Q
Task Name: Data Transfer Taskss
Task Type: DTT
Task Id: 0123UT49000000000005
Run Id: 3
Service Id: 935056
Started By: pateldhwani386@gmail.com
Run Context Type: ICS_UI
06/08/2023 08:28:27 **** Importing Connection: Conn_0123UT0B000000000006 ...
06/08/2023 08:28:27 **** Importing Connection: Conn_0123UT0B000000000007 ...
06/08/2023 08:28:27 **** Importing Source Definition: EMPLOYEES ...
06/08/2023 08:28:27 **** Importing Target Definition: EMPLOYEES ...
06/08/2023 08:28:27 **** Importing SessionConfig: default_session_config ...
    <Warning> :  The Error Log DB Connection value should have Relational: as the prefix.
    <Warning> :  Invalid value  for attribute Error Log DB Connection. Will use the default value
    Validating Source Definition  EMPLOYEES...
    Validating Target Definition  EMPLOYEES...
06/08/2023 08:28:27 **** Importing Mapping: Mapping0 ...
Validating transformations of mapping Mapping0...
Validating mapping variable(s).
06/08/2023 08:28:27 **** Importing Workflow: wf_mtt_0123UT49000000000005 ...
    <Warning> :  Invalid value false for attribute Truncate Target Table. Will use the default value NO
[Session< s_mtt_0123UT49000000000005_0dde8e8d65fb458dae7a7391c64c738e >  -->
SnowflakeCloudDataWarehouseV2_table Writer< SnowflakeCloudDataWarehouseV2_table Writer > ]
    <Warning> :  Invalid value None for attribute Pushdown Optimization Context. Will use the default value
None      [Session< s_mtt_0123UT49000000000005_0dde8e8d65fb458dae7a7391c64c738e > ]
    <Warning> :  The value entered is not a valid integer.
    <Warning> :  Invalid value NO for attribute Fail task after wait time. Will use the default value
Successfully extracted session instance [s_mtt_0123UT49000000000005_0dde8e8d65fb458dae7a7391c64c738e].
Starting repository sequence id is [1834661625]
DIRECTOR> VAR_27085 [2023-06-08 08:28:27.775] Parameter file
[/data2/home/cldagnt/SystemAgent/apps/Data_Integration_Server/data/0123UT0E00000000005P/0123UT/parameters/s_m
tt_0123UT49000000000005_0dde8e8d65fb458dae7a7391c64c738e.param] is opened for [session
[wf_mtt_0123UT49000000000005.s_mtt_0123UT49000000000005_0dde8e8d65fb458dae7a7391c64c738e]].
DIRECTOR> VAR_27062 [2023-06-08 08:28:27.775] Warning! Cannot find section for worklet
[wf_mtt_0123UT49000000000005] and folder [] in parameter file
[/data2/home/cldagnt/SystemAgent/apps/Data_Integration_Server/data/0123UT0E00000000005P/0123UT/parameters/s_m
tt_0123UT49000000000005_0dde8e8d65fb458dae7a7391c64c738e.param].
DIRECTOR> TM_6014 [2023-06-08 08:28:27.776] Initializing session
[s_mtt_0123UT49000000000005_0dde8e8d65fb458dae7a7391c64c738e] at [Thu Jun 08 08:28:27 2023].
DIRECTOR> TM_6683 [2023-06-08 08:28:27.776] Repository Name: [XMLRepository]
DIRECTOR> TM_6684 [2023-06-08 08:28:27.776] Server Name: [rDTM]
DIRECTOR> TM_6686 [2023-06-08 08:28:27.776] Folder: []
DIRECTOR> TM_6685 [2023-06-08 08:28:27.776] Workflow: [wf_mtt_0123UT49000000000005] Run Instance Name: [] Run
Id: [0]
DIRECTOR> TM_6992 [2023-06-08 08:28:27.776] Operating System Type [Linux]
DIRECTOR> TM_6101 [2023-06-08 08:28:27.776] Mapping name: Mapping0.
DIRECTOR> TM_6964 [2023-06-08 08:28:27.776] Date format for the Session is [MM/DD/YYYY HH24:MI:SS.US]
DIRECTOR> TM_6703 [2023-06-08 08:28:27.776] Session
[s_mtt_0123UT49000000000005_0dde8e8d65fb458dae7a7391c64c738e] is run by 64-bit Integration Service  [],
version [10.2.0 HotFix2], build [0320].
MANAGER> PETL_24091 [2023-06-08 08:28:27.787] Thread [MANAGER] has the ID [1].
MANAGER> PETL_24058 [2023-06-08 08:28:27.787] Running Partition Group [1].
MANAGER> PETL_24000 [2023-06-08 08:28:27.787] Parallel Pipeline Engine initializing.
MONITOR> PETL_24091 [2023-06-08 08:28:27.787] Thread [MONITOR] has the ID [2].
MANAGER> PETL_24001 [2023-06-08 08:28:27.788] Parallel Pipeline Engine running.
MANAGER> PETL_24003 [2023-06-08 08:28:27.788] Initializing session run.
MAPPING> PETL_24091 [2023-06-08 08:28:27.788] Thread [MAPPING] has the ID [3].
MAPPING> CMN_1569 [2023-06-08 08:28:27.790] Server Mode: [UNICODE]
MAPPING> CMN_1570 [2023-06-08 08:28:27.790] Server Code page: [ISO 8859-1 Western European]
MAPPING> TM_6151 [2023-06-08 08:28:27.790] The session sort order is [Binary].
MAPPING> TM_6185 [2023-06-08 08:28:27.790] Warning. Code page validation is disabled in this session.
MAPPING> CMN_65110 [2023-06-08 08:28:27.790] Current Timezone:[UTC -5:0]
MAPPING> CMN_65111 [2023-06-08 08:28:27.790] Current Process ID:[5719]
MAPPING> TM_6155 [2023-06-08 08:28:27.790] Using HIGH precision processing.
MAPPING> TM_6180 [2023-06-08 08:28:27.790] Deadlock retry logic will not be implemented.
MAPPING> TM_6187 [2023-06-08 08:28:27.790] Session target-based commit interval is [10000].
MAPPING> PMJVM_42020 [2023-06-08 08:28:27.793] [INFO] Loaded library :
/data2/home/cldagnt/SystemAgent/jdk/jre/lib/amd64/server/libjvm.so.
MAPPING> PMJVM_42009 [2023-06-08 08:28:27.850] [INFO] Created Java VM successfully.
MAPPING> SDKS_38029 [2023-06-08 08:28:27.900] Loaded plug-in 605501: [plugin description].
MAPPING> SDKS_38024 [2023-06-08 08:28:28.294] Plug-in 605501 initialization complete.
MAPPING> SDKS_38017 [2023-06-08 08:28:28.294] Writer SDK plug-in initialization complete.
MAPPING> CONNECTOR_10000 [2023-06-08 08:28:29.961] [INFO] The source or target object at runtime is
overridden with the parameter name EMPLOYEES= table name EMPLOYEES.
MAPPING> SDKS_38509 [2023-06-08 08:28:30.197] SDK target and group initialization complete.
MAPPING> TM_6307 [2023-06-08 08:28:30.197] DTM error log disabled.
MAPPING> TE_7022 [2023-06-08 08:28:30.197] TShmWriter: Initialized
MAPPING> TM_6007 [2023-06-08 08:28:30.199] DTM initialized successfully for session
[s_mtt_0123UT49000000000005_0dde8e8d65fb458dae7a7391c64c738e]
DIRECTOR> PETL_24033 [2023-06-08 08:28:30.199] All DTM Connection Info: [<NONE>].
MANAGER> PETL_24004 [2023-06-08 08:28:30.200] Starting pre-session tasks. : (Thu Jun 08 08:28:30 2023)
PRE-SESS> PETL_24091 [2023-06-08 08:28:30.200] Thread [PRE-SESS] has the ID [4].
MANAGER> PETL_24027 [2023-06-08 08:28:30.201] Pre-session task completed successfully. : (Thu Jun 08 08:28:30
2023)
DIRECTOR> PETL_24006 [2023-06-08 08:28:30.201] Starting data movement.
MAPPING> TM_6660 [2023-06-08 08:28:30.203] Total Buffer Pool size is 8314384 bytes and Block size is 459688
bytes.
READER_1_1_1> PETL_24091 [2023-06-08 08:28:30.204] Thread [READER_1_1_1] has the ID [5].
READER_1_1_1> DBG_21438 [2023-06-08 08:28:30.204] Reader: Source is
[odbc://dbtype=Oracle;host=testdb.chg7lglnyg4b.us-east-1.rds.amazonaws.com;port=1521;database=orcl], Type
[ODBC], User [admin]
READER_1_1_1> BLKR_16051 [2023-06-08 08:28:30.207] Source database connection [Conn_0123UT0B000000000006]
code page: [MS Windows Latin 1 (ANSI), superset of Latin1]
READER_1_1_1> CMN_1021 [2023-06-08 08:28:30.207] Database driver event...
CMN_1021 [Driver Manager used is [Data Direct].]

READER_1_1_1> CMN_1021 [2023-06-08 08:28:32.878] Database driver event...
CMN_1021 [Reading driver properties from :
/data2/home/cldagnt/SystemAgent/apps/Data_Integration_Server/65.0.1.1/ICS/main/bin/rdtm/../../../../drivers/d
river.properties
DTM flag to determine driver name is not found hence default to [DataDirect CLOSED 8.0 Oracle Wire Protocol
for Informatica - R40] driver.
ODBC Event Using array fetches.
ODBC Event Using Single Row Inserts. connect string = [Driver=DataDirect CLOSED 8.0 Oracle Wire Protocol for
Informatica - R40;Host=testdb.chg7lglnyg4b.us-east-
1.rds.amazonaws.com;ServiceName=orcl;Port=1521;UID=admin;AS=6000000;EBL=0;ENS=1;IACP=4;ETWT=1;WorkArounds2=2;
AllowedOpenSSLVersions=1.0.2;EnableEmptyLobs=1]. userid = [admin]
ODBC Version:     03.52.0000
.
DBMS Name:        Oracle
.
DBMS Version:     19.00.0000 Oracle 19.0.0.0.0
.
Driver Name:      DWora28.so
.
Driver Version:   08.02.2432 (B0798, U0577)
.
Driver ODBC Version:      03.52
.]
```

```
READER_1_1_1> BLKR_16003 [2023-06-08 08:28:32.878] Initialization completed successfully.
TRANSF_1_1_1> PETL_24091 [2023-06-08 08:28:32.879] Thread [TRANSF_1_1_1] has the ID [6].
WRITER_1_*_1> PETL_24091 [2023-06-08 08:28:32.879] Thread [WRITER_1_*_1] has the ID [7].
WRITER_1_*_1> CONNECTOR_10000 [2023-06-08 08:28:32.890] [INFO] Normal Mode (Non-Bulk Processing) is enabled
to Write data
WRITER_1_*_1> SNOWFLAKECLOUDDATAWAREHOUSE_10000 [2023-06-08 08:28:36.022] [INFO] The Snowflake Connector uses
the following JDBC URL to connect to Snowflake: jdbc:snowflake://qkwpabt-
gk41830.snowflakecomputing.com/?warehouse=warehouse_target&role=ACCOUNTADMIN&SSL=on&application=INFA_DI_Cloud
WRITER_1_*_1> WRT_8270 [2023-06-08 08:28:37.242] Target connection group #1 consists of target(s) [EMPLOYEES:
Partition 1]
WRITER_1_*_1> WRT_8003 [2023-06-08 08:28:37.242] Writer initialization complete.
WRITER_1_*_1> WRT_8005 [2023-06-08 08:28:37.242] Writer run started.
READER_1_1_1> BLKR_16007 [2023-06-08 08:28:37.242] Reader run started.
WRITER_1_*_1> WRT_8158 [2023-06-08 08:28:37.242]

*****START LOAD SESSION*****

Load Start Time: Thu Jun 08 08:28:36 2023

Target tables:

     EMPLOYEES: Partition 1


READER_1_1_1> RR_4010 [2023-06-08 08:28:37.243] SQ instance [Source] SQL Query [SELECT
ADMIN.EMPLOYEES.EMPLOYEE_ID, ADMIN.EMPLOYEES.FIRST_NAME, ADMIN.EMPLOYEES.LAST_NAME, ADMIN.EMPLOYEES.EMAIL,
ADMIN.EMPLOYEES.PHONE, ADMIN.EMPLOYEES.HIRE_DATE, ADMIN.EMPLOYEES.MANAGER_ID, ADMIN.EMPLOYEES.JOB_TITLE FROM
ADMIN.EMPLOYEES]
READER_1_1_1> RR_4049 [2023-06-08 08:28:37.243] SQL Query issued to database : (Thu Jun 08 08:28:37 2023)
READER_1_1_1> RR_4050 [2023-06-08 08:28:37.668] First row returned from database to reader : (Thu Jun 08
08:28:37 2023)
READER_1_1_1> BLKR_16019 [2023-06-08 08:28:37.669] Read [107] rows, read [0] error rows for source table
[EMPLOYEES] instance name [EMPLOYEES]
READER_1_1_1> BLKR_16008 [2023-06-08 08:28:37.669] Reader run completed.
TRANSF_1_1_1> DBG_21216 [2023-06-08 08:28:37.669] Finished transformations for Source Qualifier [Source].
Total errors [0]
WRITER_1_*_1> WRT_8167 [2023-06-08 08:28:37.669] Start loading table [EMPLOYEES: Partition 1] at: Thu Jun 08
08:28:36 2023
WRITER_1_*_1> WRT_8168 [2023-06-08 08:28:37.693] End loading table [EMPLOYEES: Partition 1] at: Thu Jun 08
08:28:36 2023
WRITER_1_*_1> SNOWFLAKECLOUDDATAWAREHOUSE_10000 [2023-06-08 08:28:37.693] [INFO] The Snowflake Connector
completed writing data to the target.
WRITER_1_*_1> WRT_8035 [2023-06-08 08:28:45.200] Load complete time: Thu Jun 08 08:28:44 2023

LOAD SUMMARY
============

WRT_8036 Target: EMPLOYEES: Partition 1 (Instance Name: [EMPLOYEES])
WRT_8038 Inserted rows - Requested: 107        Applied: 107        Rejected: 0          Affected: 107


WRITER_1_*_1> WRT_8043 [2023-06-08 08:28:45.200] *****END LOAD SESSION*****
WRITER_1_*_1> WRT_8006 [2023-06-08 08:28:45.200] Writer run completed.
MANAGER> PETL_24031 [2023-06-08 08:28:45.201]
***** RUN INFO FOR TGT LOAD ORDER GROUP [1], CONCURRENT SET [1] *****
Thread [READER_1_1_1] created for [the read stage] of partition point [Source] has completed.
         Total Run Time = [0.627216] secs
         Total Idle Time = [0.000000] secs
         Busy Percentage = [100.000000]
Thread [TRANSF_1_1_1] created for [the transformation stage] of partition point [Source] has completed. The
total run time was insufficient for any meaningful statistics.
Thread [WRITER_1_*_1] created for [the write stage] of partition point [EMPLOYEES] has completed.
         Total Run Time = [7.531312] secs
         Total Idle Time = [0.000000] secs
         Busy Percentage = [100.000000]

MANAGER> PETL_24005 [2023-06-08 08:28:45.202] Starting post-session tasks. : (Thu Jun 08 08:28:45 2023)
POST-SESS> PETL_24091 [2023-06-08 08:28:45.202] Thread [POST-SESS] has the ID [8].
MANAGER> PETL_24029 [2023-06-08 08:28:45.202] Post-session task completed successfully. : (Thu Jun 08
08:28:45 2023)
MAPPING> SDKS_38510 [2023-06-08 08:28:45.203] SDK target and group deinitialized with status [0].
MAPPING> SDKS_38025 [2023-06-08 08:28:45.203] Plug-in 605501 deinitialized and unloaded with status [0].
MAPPING> SDKS_38019 [2023-06-08 08:28:45.203] Writer SDK plug-ins deinitialized with status [0].
MAPPING> TM_6018 [2023-06-08 08:28:45.203] The session completed with [0] row transformation errors.
MANAGER> PETL_24002 [2023-06-08 08:28:45.897] Parallel Pipeline Engine finished.
DIRECTOR> PETL_24012 [2023-06-08 08:28:45.897] Session run completed successfully.
DIRECTOR> TM_6022 [2023-06-08 08:28:45.897]
```

```
SESSION LOAD SUMMARY
================================================

DIRECTOR> TM_6252 [2023-06-08 08:28:45.897] Source Load Summary.
DIRECTOR> CMN_1740 [2023-06-08 08:28:45.897] Table: [Source] (Instance Name: [Source] Instance UI Name:
[Source])
         Output Rows [107], Affected Rows [107], Applied Rows [107], Rejected Rows [0]
DIRECTOR> TM_6253 [2023-06-08 08:28:45.897] Target Load Summary.
DIRECTOR> CMN_1740 [2023-06-08 08:28:45.897] Table: [EMPLOYEES] (Instance Name: [EMPLOYEES] Instance UI Name:
[Target])
         Output Rows [107], Affected Rows [107], Applied Rows [107], Rejected Rows [0]
DIRECTOR> TM_6023 [2023-06-08 08:28:45.897]
===================================================

DIRECTOR> TM_6020 [2023-06-08 08:28:45.897] Session
[s_mtt_0123UT49000000000005_0dde8e8d65fb458dae7a7391c64c738e] completed at [Thu Jun 08 08:28:45 2023].
```