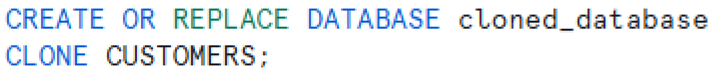
Assignment- 11

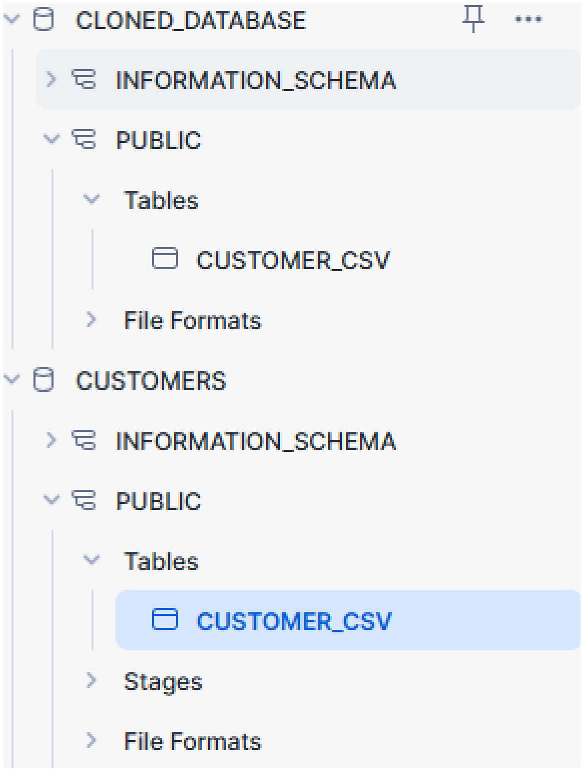
1. Create a Clone of an Existing Database Using Zero-Copy Cloning

Step 1: Clone a Database or Table

Use the CLONE command to create a clone of a database or table without copying data.

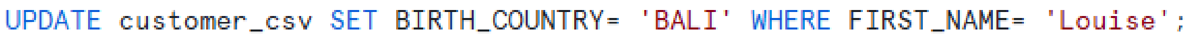


```
CREATE OR REPLACE DATABASE cloned_database
CLONE CUSTOMERS;
```

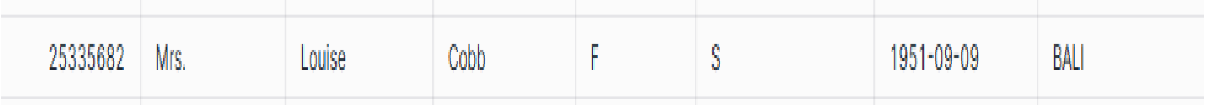


Step 2: Modify the Clone

Modify data in the cloned table to see that it does not affect the original data:



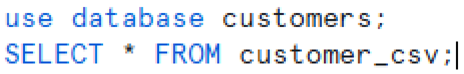
```sql
UPDATE customer_csv SET BIRTH_COUNTRY= 'BALI' WHERE FIRST_NAME= 'Louise';
```



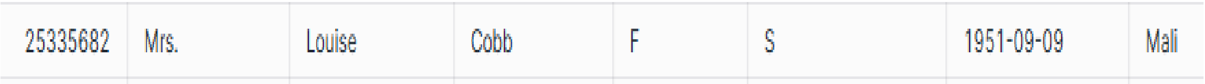
25335682	Mrs.	Louise	Cobb	F	S	1951-09-09	BALI

Step 3: Verify the Original Data

Query the original table and confirm that it has not been affected by changes made in the clone:



```sql
use database customers;
SELECT * FROM customer_csv;
```



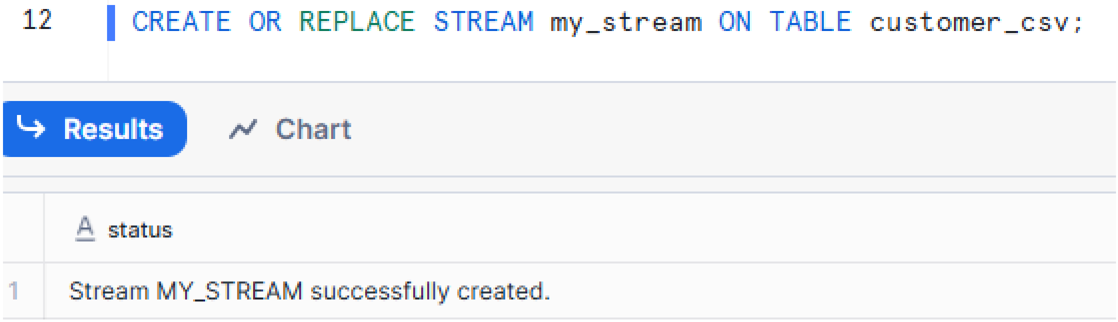
25335682	Mrs.	Louise	Cobb	F	S	1951-09-09	Mali

So we can clearly see no changes in the original table after updating anything in the cloned table or database.

2. Set Up a Stream and Task for Continuous Integration

Step 1: Create a Stream

Create a stream to track changes in a table.

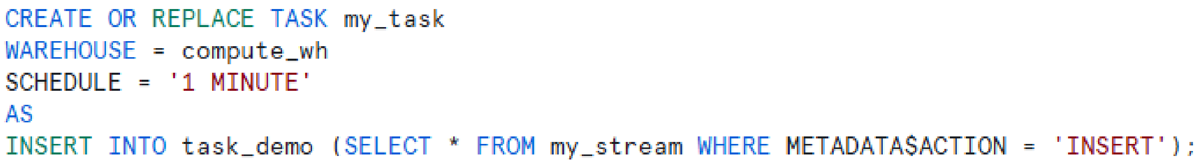


```sql
CREATE OR REPLACE STREAM my_stream ON TABLE customer_csv;
```

	status
1	Stream MY_STREAM successfully created.

Step 2: Create a Task

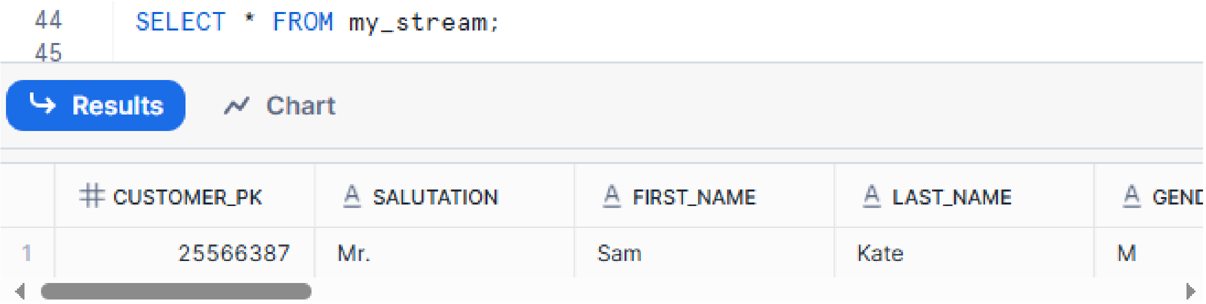
Create a task that processes the stream and inserts changed data into a target table.



```
CREATE OR REPLACE TASK my_task
WAREHOUSE = compute_wh
SCHEDULE = '1 MINUTE'
AS
INSERT INTO task_demo (SELECT * FROM my_stream WHERE METADATA$ACTION = 'INSERT');
```

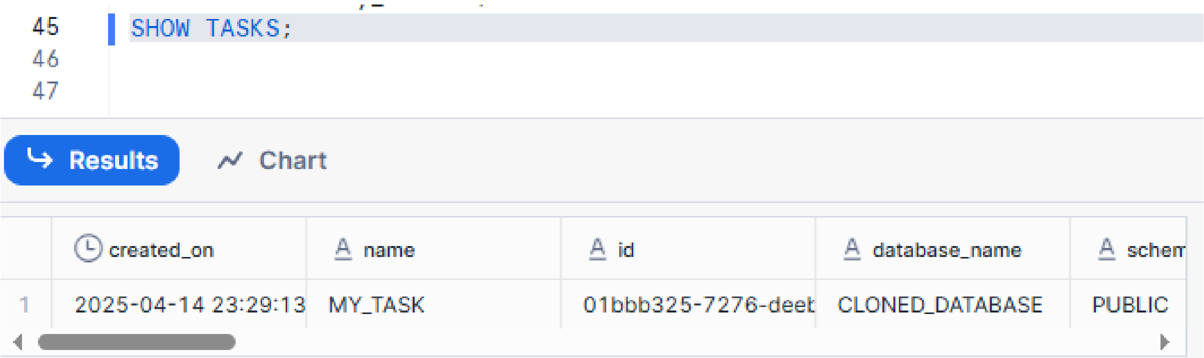
Step 3: Monitor the Stream and Task

View the stream data and task status:



```
SELECT * FROM my_stream;
```

	CUSTOMER_PK	SALUTATION	FIRST_NAME	LAST_NAME	GEND
1	25566387	Mr.	Sam	Kate	M

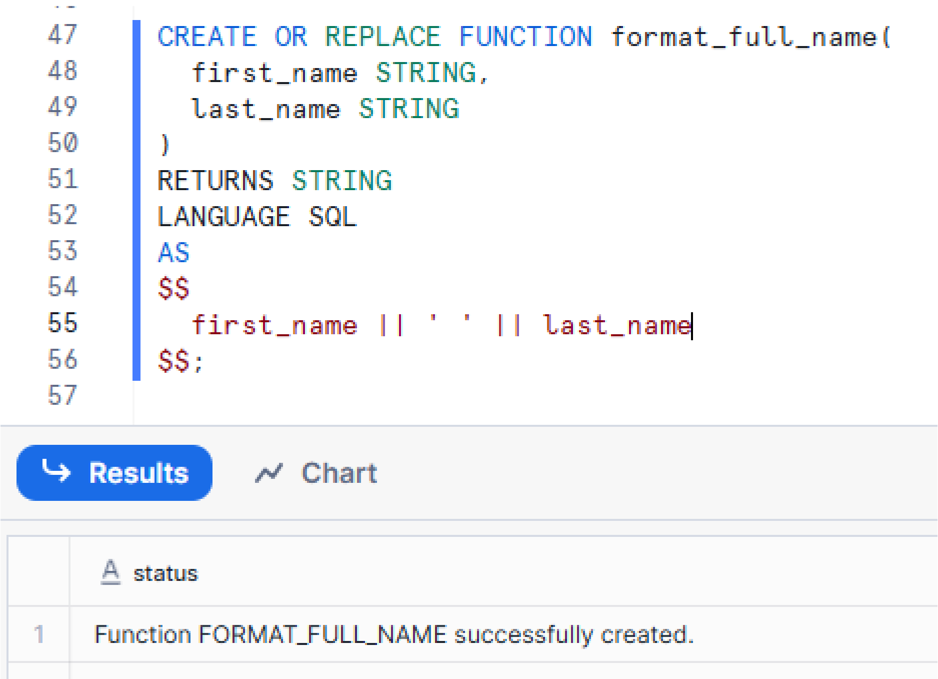


```
SHOW TASKS;
```

	created_on	name	id	database_name	schem
1	2025-04-14 23:29:13	MY_TASK	01bbb325-7276-deet	CLONED_DATABASE	PUBLIC

3. Implement a Simple UDF in SQL

Step 1: Create a SQL UDF

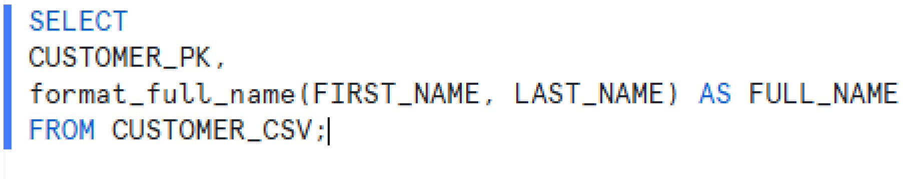


```
CREATE OR REPLACE FUNCTION format_full_name(
  first_name STRING,
  last_name STRING
)
RETURNS STRING
LANGUAGE SQL
AS
$$
  first_name || ' ' || last_name
$$;
```

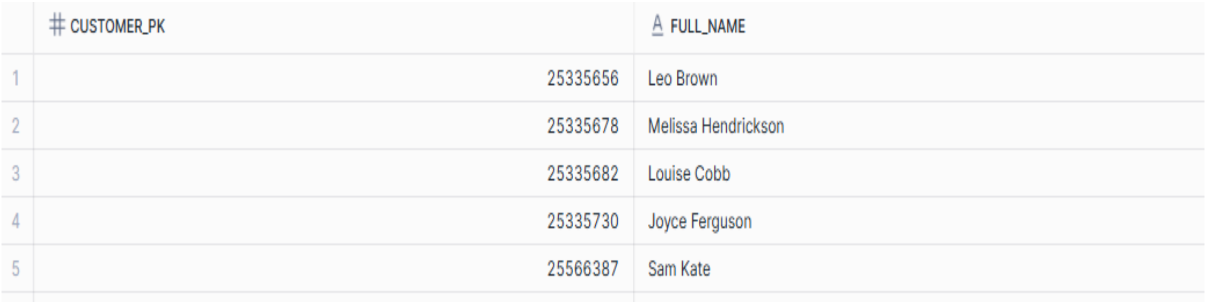
	status
1	Function FORMAT_FULL_NAME successfully created.

Step 2: Use the UDF

Apply the UDF in a query:



```
SELECT
CUSTOMER_PK,
format_full_name(FIRST_NAME, LAST_NAME) AS FULL_NAME
FROM CUSTOMER_CSV;
```



	# CUSTOMER_PK	FULL_NAME
1	25335656	Leo Brown
2	25335678	Melissa Hendrickson
3	25335682	Louise Cobb
4	25335730	Joyce Ferguson
5	25566387	Sam Kate