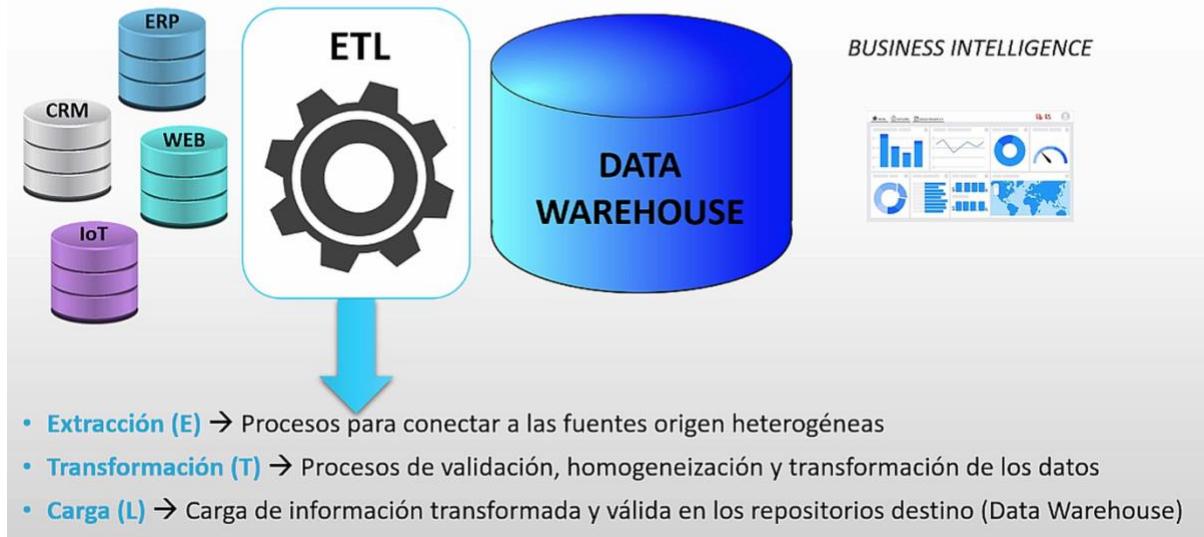


Snowflake



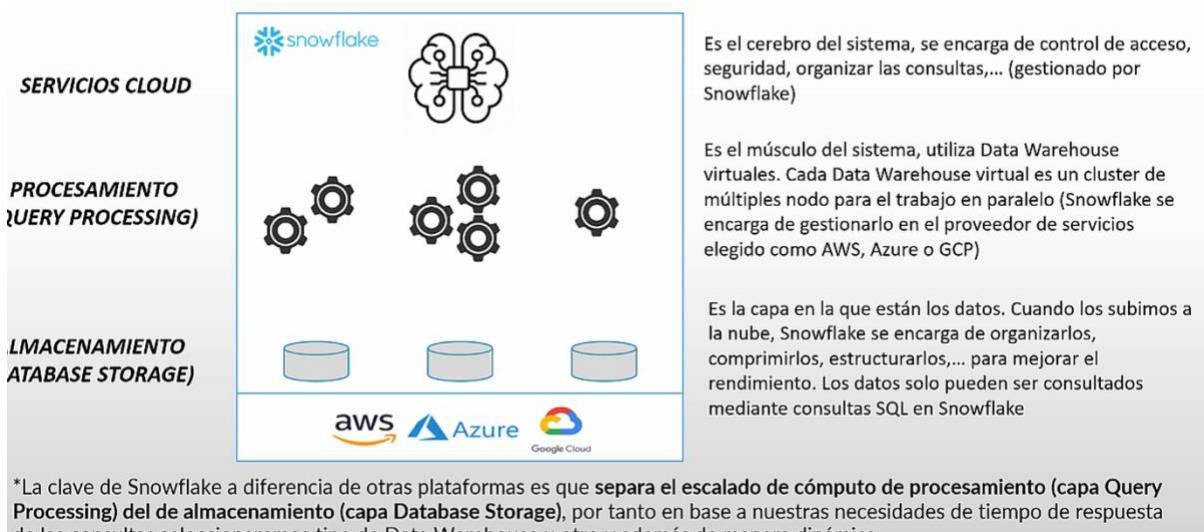
## ¿Qué es un data warehouse?

FUENTES DE INFORMACIÓN



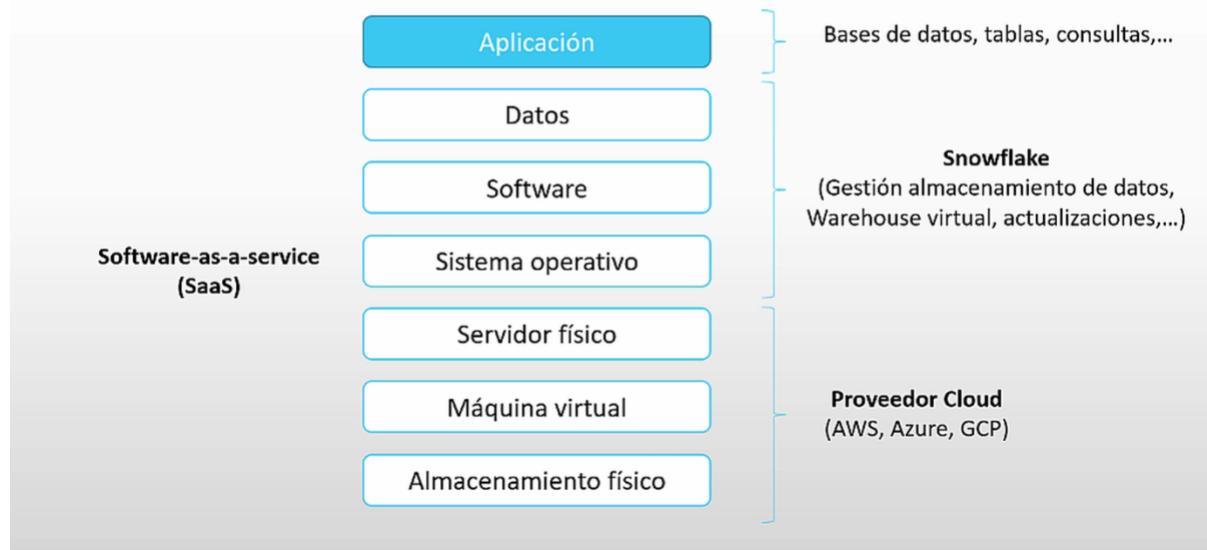
Cuando subimos a la nube, snowflake comprime ficheros

## Arquitectura de Snowflake y sus ventajas



# Arquitectura de Snowflake y sus ventajas

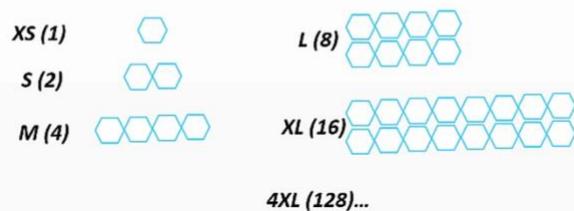
## MODELO DE CLOUD COMPUTING



## Arquitectura de Snowflake y sus ventajas



### Tamaños de Data Warehouse virtual



### Multi-clustering



### Política de escalado

**Standard**  
Favorece la creación warehouse adicional

**Economy**  
Favorece conservar créditos en lugar de crear warehouse adicional

Estructura de costes y precios de Snowflake

SELECCIONA LOS PRECIOS DE TU REGIÓN Y TU PLATAFORMA PREFERIDA

Elige la plataforma y la región para ver los precios

STANDARD	ENTERPRISE	BUSINESS CRITICAL	VIRTUAL PRIVATE SNOWFLAKE (VPS)
 Data warehouse SQL completo Secure Data Sharing en diferentes regiones/nubes Soporte Premier 24 horas al día, 365 días al año  Cifrado de categoría empresarial siempre activo en tránsito y en reposo Warehouses virtuales especiales para el cliente Autenticación federada Réplica de bases de datos Funciones externas  Snowsight Creación de tu propio Data Exchange Acceso a Snowflake Marketplace	 <b>Standard +</b> Warehouse multicíster Hasta 90 días de Time Travel Regeneración anual de claves para todos los datos cifrados Vistas materializadas Servicio de optimización de búsqueda Enmascaramiento de datos dinámicos Tokenización de datos externos	 <b>Enterprise +</b> Asistencia para el cumplimiento de la ley HIPAA  Cumplimiento de PCI Cifrado Triple DES con claves administradas por el cliente Compatibilidad con AWS PrivateLink Compatibilidad con Azure Private Link Failover y fallback de la base de datos para garantizar la continuidad del negocio Funciones externas - Asistencia para terminales privados de AWS API Gateway	 <b>Business Critical +</b> Servidores virtuales dedicados al cliente dondequiera que se encuentre la clave de cifrado en la memoria Almacén de metadatos especial para el cliente
<a href="#">EMPEZAR</a>	<a href="#">EMPEZAR</a>	<a href="#">EMPEZAR</a>	<a href="#">CONTACTO</a>

 CAPACIDAD DE ALMACENAMIENTO



## Ediciones de Snowflake y precios

### ¿Cómo se calcula el coste de Snowflake?

#### CÓMPUTO

- Warehouses activos por hora
- Depende del tamaño de warehouse
- Facturado por segundo (mínimo 1')
- Cálculo con créditos de snowflake (el precio del crédito depende de la edición)

#### ALMACENAMIENTO

- Fijo mensual por almacenamiento
- Basado en el promedio de almacenamiento usado mensual
- Coste calculado después de la compresión de datos (mayor eficiencia)
- Depende del proveedor Cloud

#### Ejemplo Coste Almacenamiento:

BUSINESS CRITICAL / AWS / US EAST  
Pensamos que necesitamos 1 TB de almacenamiento

#### ALMACENAMIENTO ON-DEMAND

#### Escenario 1: 100 GB de almacenamiento usado:

$$0.1 \text{ TB} \times 40\text{€} / \text{TB} = 4\text{€}$$

#### Escenario 2: 800 GB de almacenamiento usado:

$$0.8 \text{ TB} \times 40\text{€} / \text{TB} = 32\text{€}$$

#### ALMACENAMIENTO POR CAPACIDAD

#### Escenario 1: 100 GB de almacenamiento usado:

$$1 \text{ TB} \times 23\text{€} / \text{TB} = 23\text{€}$$

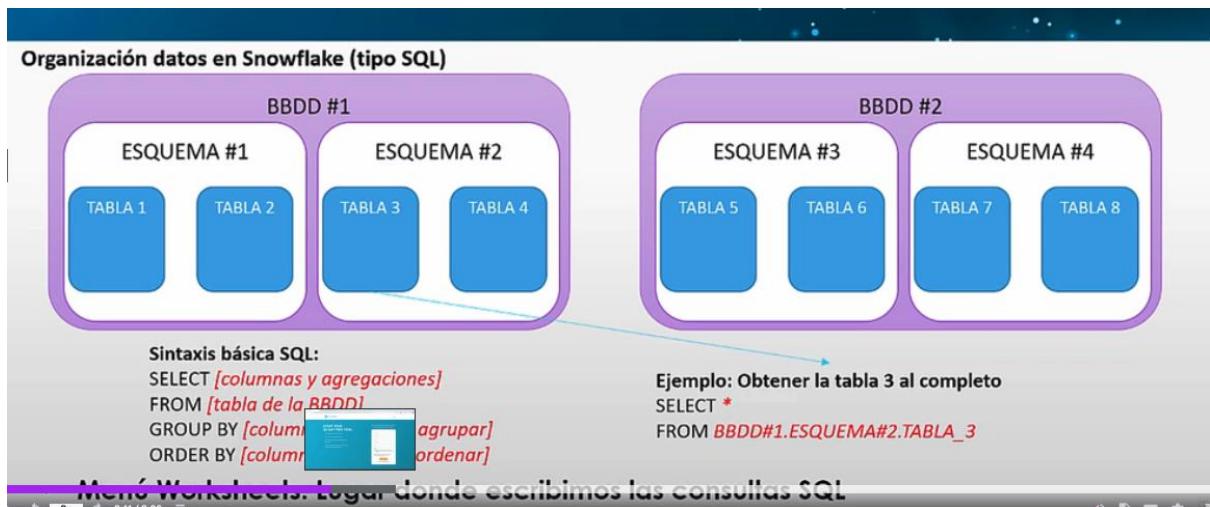
#### Escenario 2: 800 GB de almacenamiento usado:

$$1 \text{ TB} \times 23\text{€} / \text{TB} = 23\text{€}$$

#### RECOMENDACIÓN:

1) Empezar con "On demand storage"

2) Cuando estés seguro del uso pasar a "Capacity storage"



Iván Pinar  
ACCOUNTADMIN

- Worksheets
- Dashboards**
- Data
- Marketplace
- Activity
- Admin
- Help & Support

30 days left in trial  
Upgrade

Welcome to Dashboards

Use Dashboards to create and share beautiful and interactive visualizations of your data.

Worksheets 2024-02-04 9:37am

Databases Worksheets

Pinned (0)

No pinned objects

Search objects

> SNOWFLAKE

> SNOWFLAKE\_SAMPLE\_DATA

No Database selected Settings

1 select col from table where created = :daterange

ACCOUNTADMIN COMPUTE\_WH Share

# Crear un nuevo warehouse mediante interfaz o usando SQL



## Mediante interfaz

The screenshot shows the Snowflake web interface. On the left, there's a sidebar with navigation links like Worksheets, Dashboards, Data, Marketplace, Activity, Admin, Usage, Warehouses (which is selected), Resource Monitors, Users & Roles, Security, Editing & Terms, Contacts, and Accounts. The main area is titled 'Warehouses' and shows a table with one row: '1 Warehouse'. The first column is 'NAME' with 'COMPUTE\_WH', the second is 'STATUS' with 'Suspended', and the third is 'SIZE' with 'X-Small'. A modal window titled 'New Warehouse' is open in the center. It has fields for 'Name' (containing 'I'), 'Size' (set to 'X-Small 1 credit/hour'), and 'Comment (optional)'. There are also sections for 'Query Acceleration' and 'Multi-cluster Warehouse'. At the bottom of the modal are 'Cancel' and 'Create Warehouse' buttons.

## Mediante sentencia SQL

```
1 CREATE WAREHOUSE EJEMPLO_WAREHOUSE
2 WITH
3 WAREHOUSE_SIZE=XSMALL
4 MAX_CLUSTER_COUNT=3
5 AUTO_SUSPEND = 300
6 AUTO_RESUME = TRUE
7 INITIALLY_SUSPENDED = TRUE
8 COMMENT = "Ejemplo de warehouse"
9
10 DROP WAREHOUSE EJEMPLO_WAREHOUSE
```

The screenshot displays two main windows from the Snowflake web interface.

**Top Window: Schema Overview**

- Left Sidebar:** Shows user information (Iván Pinar, ACCOUNTADMIN), navigation links (Worksheets, Dashboards, Data, Databases, Marketplace, Activity, Admin, Help & Support), and a trial status message ("27 days left in trial" with an "Upgrade" button).
- Central Area:** Displays the **SNOWFLAKE\_SAMPLE\_DATA / TPCDS\_SF10TCL** schema details. It shows a **Schema** created 7 months ago with 10 TB of data in a Clustered format, containing **24 Tables**. A table list table is shown below:

NAME	TYPE	OWNER	ROWS	BYTES	CREATED
CALL_CENTER	Table	—	54	21.5KB	7 months ago
CATALOG_PAGE	Table	—	40K	2.1MB	7 months ago
CATALOG RETURNS	Table	—	1.4B	77.0GB	7 months ago
CATALOG SALES	Table	—	14.4B	956.4GB	7 months ago
CUSTOMER	Table	—	65M	2.2GB	7 months ago
CUSTOMER ADDRESS	Table	—	32.5M	589.7MB	7 months ago
CUSTOMER DEMOGRAPHICS	Table	—	1.9M	73MB	7 months ago
DATE DIM	Table	—	73.0K	2.1MB	7 months ago
HOUSEHOLD DEMOGRAPHICS	Table	—	7.2K	27.0KB	7 months ago
INCOME BAND	Table	—	20	1.5KB	7 months ago
INVENTORY	Table	—	1.3B	5.2GB	7 months ago
ITEM	Table	—	402K	37.0MB	7 months ago
PROMOTION	Table	—	2K	82.5KB	7 months ago
REASON	Table	—	70	3.0KB	7 months ago
SHIP MODE	Table	—	20	4.0KB	7 months ago
STORES	Table	—	1.5K	138.5KB	7 months ago

**Bottom Window: New Warehouse Creation Dialog**

This dialog is titled **New Warehouse** and is being created as **ACCOUNTADMIN**.

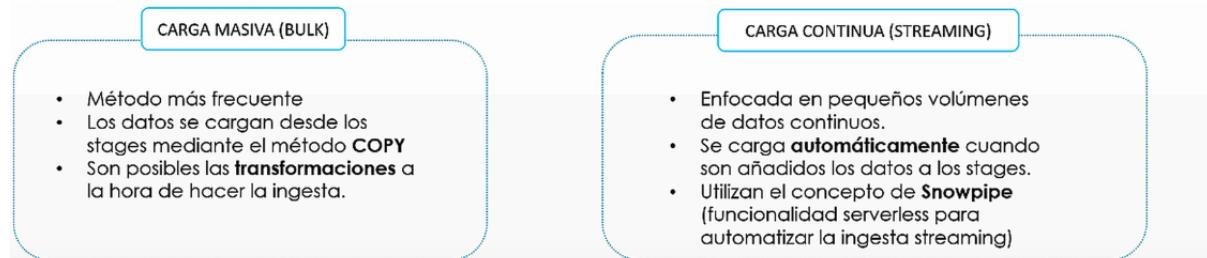
- Auto suspend:** Checked. Description: "Automatically suspends the warehouse if it is inactive for the specified period of time." Suspend After: 5 min(s) of inactivity.
- Multi-cluster Warehouse:** Checked. Description: "Scale compute resources as query concurrency needs change." [Learn more](#)
- Min Clusters:** 1. **Max Clusters:** 3. **Scaling Policy:** Standard.
- Query Acceleration:** Checked. Description: "Accelerate outlier queries with additional flexible compute resources." [Learn more](#)
- Scale Factor:** 8x.

Buttons at the bottom: **Create Warehouse** (highlighted in blue) and **Cancel**.

## Carga de datos en Snowflake

## Adquisición e ingestión de datos en Snowflake (stages)

### ¿Qué tipos de carga existen?

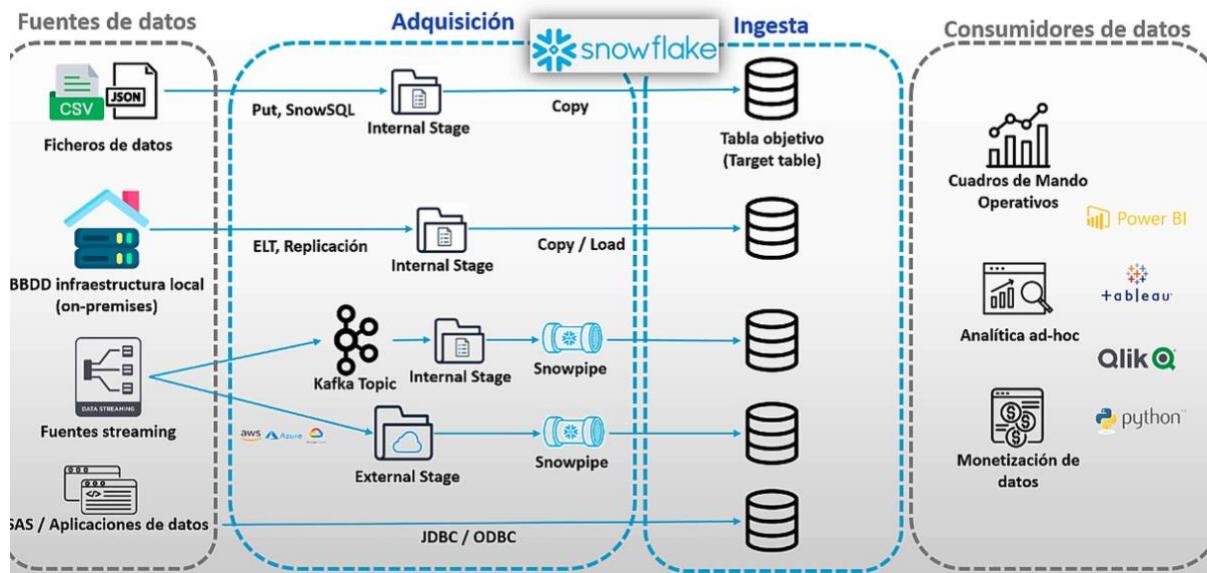


### ¿Qué es un stage?

- Lugar donde se encuentran almacenados los ficheros de datos y desde donde cargar los datos en la tabla objetivo.



## Adquisición e ingestión de datos en Snowflake (stages)



1) Database 2) Schema 3) Table

```

1   create table tabla_ventas (
2       ID_Pedido int
3       ,Unidades int
4       ,Precio_unitario float
5       ,Total_Venta float
6       ,Fecha date
7
8   )
9

```

The screenshot shows a database management interface with two main panes. The left pane, titled 'Databases' and 'Worksheets', contains a sidebar with pinned objects: All Objects, BBDD\_VENTAS, MANAGE\_DB, PRIMERABDD, SNOWFLAKE, and SNOWFLAKE\_SAMPLE\_DATA. The right pane is a code editor with the following SQL script:

```

MANAGE_DB.EXTERNAL_STAGES *      Settings *
18
19 CREATE OR REPLACE TABLE PRIMERABDD.PRIMERESQUEMA.ORDERS_EX (
20     ORDER_ID VARCHAR(30),
21     AMOUNT INT,
22     PROFIT INT,
23     PROFITABLE_FLAG VARCHAR(30)
24
25
26
27 COPY INTO PRIMERABDD.PRIMERESQUEMA.ORDERS_EX
28     FROM (select
29         s.$1,
30         s.$2,
31         s.$3,
32         CASE WHEN CAST(s.$3 as int) < 0 THEN 'not profitable' ELSE 'profitable' END
33         from @MANAGE_DB.external_stages.aws_stage s)
34     file_format= (type = csv field_delimiter='.' skip_header=1)
35     files='OrderDetails.csv';
36
37
38 SELECT * FROM PRIMERABDD.PRIMERESQUEMA.ORDERS_EX
39
40
41 // Ejemplo 3 Seleccionar Substring de una columna - Tabla
42

```

The code editor has tabs for 'Results' and 'Chart'. Below the code editor is a results table with one row:

	status
1	Table ORDERS_EX successfully created.

Errors:

Databases Worksheets

No pinned objects

All objects

- BBDD\_VENTAS
- MANAGE\_DB
  - EXTERNAL\_STAGES
    - Stages
      - AWS\_STAGE
      - AWS\_STAGE\_ERROREX
  - INFORMATION\_SCHEMA
  - PUBLIC
- PRIMERABDD
- SNOWFLAKE
- SNOWFLAKE\_SAMPLE\_DATA

No Database selected    Settings

```

44   FROM @MANAGE_DB.EXTERNAL_STAGES.AWS_STAGE_ERROREX
45   file_format=(type = csv field_delimiter=',' skip_header=1)
46   files = ('OrderDetails_error.csv','OrderDetails_error2.csv')
47   ON_ERROR = 'ABORT_STATEMENT';
48
49
50   // Verificar resultados y truncar (limpiar) la tabla
51   SELECT * FROM PRIMERABDD.PRIMERESQUEMA.ORDERS_EX
52   SELECT COUNT(*) FROM PRIMERABDD.PRIMERESQUEMA.ORDERS_EX
53
54   TRUNCATE TABLE PRIMERABDD.PRIMERESQUEMA.ORDERS_EX;
55
56   // Manejo de errores usando la opción ON_ERROR = SKIP_FILE y los 2 ficheros
57   COPY INTO PRIMERABDD.PRIMERESQUEMA.ORDERS_EX
58   FROM @MANAGE_DB.external_stages.aws_stage_errorex
59   file_format=(type = csv field_delimiter=',' skip_header=1)
60   files = ('OrderDetails_error.csv','OrderDetails_error2.csv')
61   ON_ERROR = 'SKIP_FILE';
62
63
64   // Verificar resultados y truncar (limpiar) la tabla
65   SELECT * FROM PRIMERABDD.PRIMERESQUEMA.ORDERS_EX
66   SELECT COUNT(*) FROM PRIMERABDD.PRIMERESQUEMA.ORDERS_EX

```

Results Chart

	status
1	Statement executed successfully.

Query Details

- Query duration 133ms
- Rows 1

```

85
86   // Manejo de errores usando la opción ON_ERROR = SKIP_FILE_<number> porcentual y los 2 ficheros
87   COPY INTO PRIMERABDD.PRIMERESQUEMA.ORDERS_EX
88   FROM @MANAGE_DB.external_stages.aws_stage_errorex
89   file_format=(type = csv field_delimiter=',' skip_header=1)
90   files = ('OrderDetails_error.csv','OrderDetails_error2.csv')
91   ON_ERROR = 'SKIP_FILE_0.5%';
92
93
94   // Verificar resultados y truncar (limpiar) la tabla
95   SELECT * FROM PRIMERABDD.PRIMERESQUEMA.ORDERS_EX
96   SELECT COUNT(*) FROM PRIMERABDD.PRIMERESQUEMA.ORDERS_EX
97
98   TRUNCATE TABLE PRIMERABDD.PRIMERESQUEMA.ORDERS_EX;
99
100

```

Results Chart

	status	rows_parsed	rows_loaded	error_limit	... errors_seen	first_error	# errors_seen
1	1/OrderDetails_error2.csv	LOADED	285	285	1	0	null
2	1/OrderDetails_error.csv	PARTIALLY_LOADED	1,500	1,498	7	2	Numeri

COPY\_DB.PUBLIC ▾ Settings ▾

Latest Version ▾

```

74   COPY_TYPE = 'CSV', FILENAME_SUFFIX = ' ', DATA_FORMATTER = 'Z'
75   patterns', 'Order.*'
76   VALIDATION_MODE = RETURN_ERRORS;
77
78   // Almacenar registros erróneos en una tabla
79   CREATE OR REPLACE TABLE rechazados AS
80   select rejected_record from table(result_scan(last_query_id()));
81
82   SELECT * FROM rechazados;
83
84
85
86   ---- 2) Transformar registros erróneos ----
87
88   CREATE OR REPLACE TABLE valores_erroneos AS
89   SELECT
90   SPLIT_PART(rejected_record,'.',1) as ORDER_ID,
91   SPLIT_PART(rejected_record,'.',2) as AMOUNT,
92   SPLIT_PART(rejected_record,'.',3) as PROFIT,
93   SPLIT_PART(rejected_record,'.',4) as QUANTITY,
94   SPLIT_PART(rejected_record,'.',5) as CATEGORY,
95   SPLIT_PART(rejected_record,'.',6) as SUBCATEGORY
96   FROM rechazados;
97
98
99   | SELECT * FROM valores_erroneos;

```

↳ Results ▾ Chart

ORDER_ID	AMOUNT	PROFIT	QUANTITY	...	CATEGORY	SUBCATEGORY
1 B-30601	1275	10	7-		Furniture	Bookcases
2 B-30601	8	-244	30		Clothing	Hankerchief
3 B-25601	1275	one thousand	7		Furniture	Bookcases
4 B-25601	66	two hundred twenty	5		Clothing	Stole

Query Details

Query duration

Rows

Query ID: 01ac8008-0000-1

ORDER\_ID

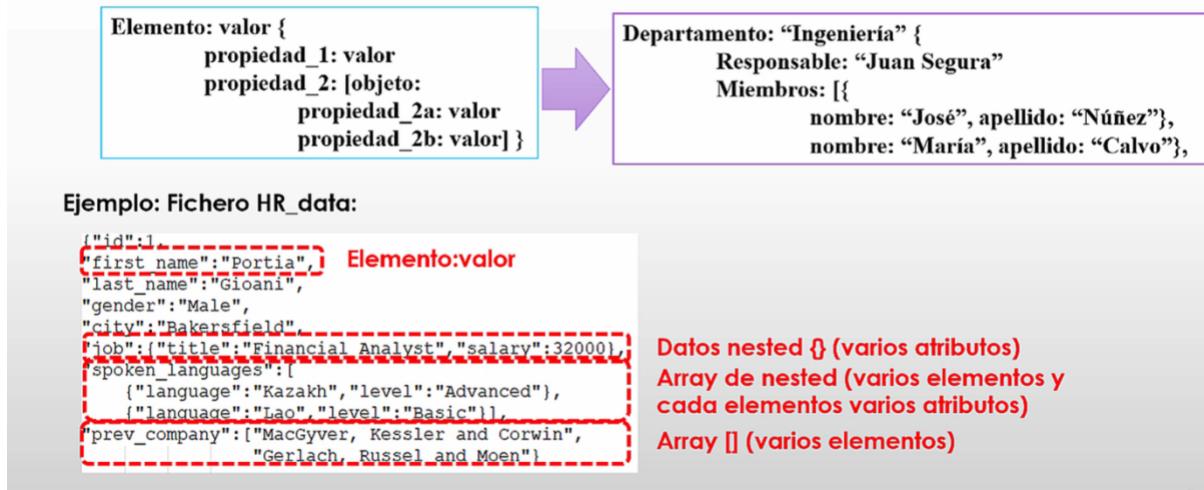
Carga de datos no estructurados

## Lo que aprenderemos en este bloque....

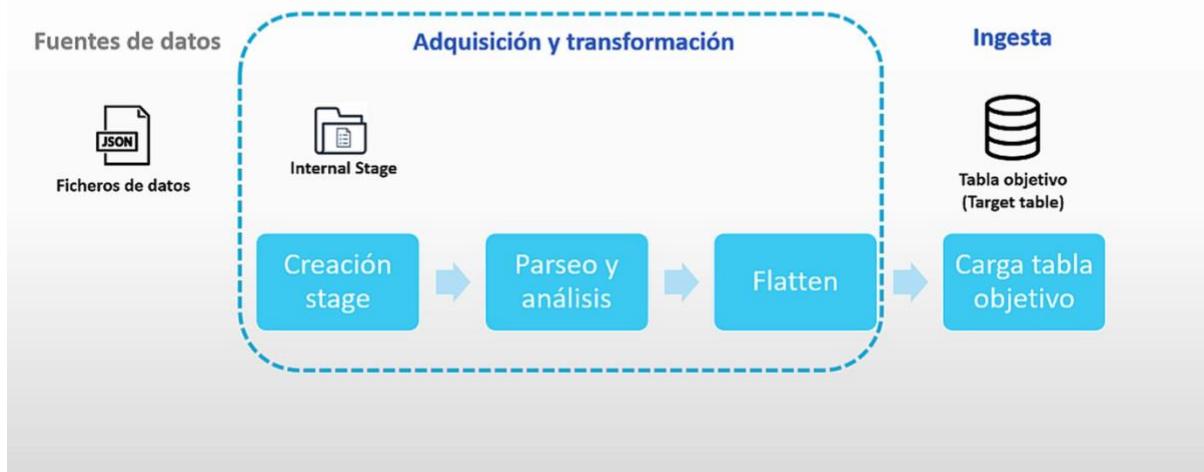
- ✓ ¿Qué es el formato NO estructurado (JSON, XML, Parquet,...)?
- ✓ ¿Cómo crear un stage y cargar los datos brutos no estructurados?
- ✓ ¿Cómo realizar el parseo y transformación de datos no estructurados?
- ✓ ¿Cómo cargar datos no estructurados en la tabla objetivo (estructurado)?

## ¿Qué son los datos NO estructurados?

- ¿Qué es el formato JSON? (Formato NO estructurado más utilizado)
  - JSON (JavaScript Object Notation) es un formato semiestructurado ligero útil para el intercambio de datos. Es fácil de analizar (parsear).



## ¿Qué son los datos NO estructurados?



El proceso, por tanto, que vamos a llevar a cabo para hacer toda esta ingestión de datos JSON y la carga

final en la tabla objetivo va a ser el siguiente.

Vamos a tener nuestra fuente de datos JSON.

Después comenzaremos el bloque de adquisición y transformación, donde haremos la creación del stage, el Internal Stage, puesto que vamos a cargar desde un fichero en local en esta situación.

En este caso vamos a hacer un parseo de análisis de ese fichero JSON.

Y por último vamos a aplicar el flatten, que básicamente es transformar los datos no estructurados en datos estructurados.

Cuando ya los tenemos estructurados, pues finalmente haremos la ingestión en la tabla objetivo.

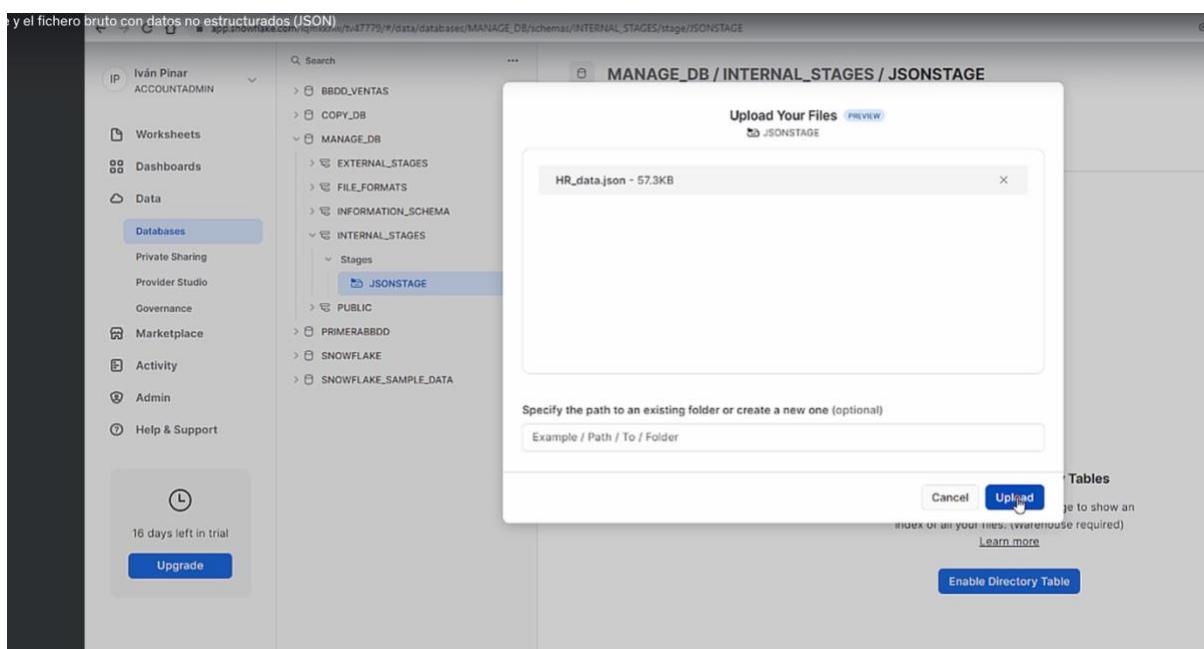
The screenshot shows the Snowflake interface with the following details:

- Left sidebar:** Shows databases like BBDD\_VENTAS, COPY\_DB, and MANAGE\_DB.
- Central pane:** A code editor window titled "MANAGE\_DB.PUBLIC" containing SQL commands to create a schema and an internal stage. The schema creation part is:
 

```
CREATE OR REPLACE SCHEMA MANAGE_DB.INTERNAL_STAGES;
CREATE OR REPLACE stage MANAGE_DB.INTERNAL_STAGES.JSONSTAGE
```

 The stage creation part is:
 

```
CREATE OR REPLACE stage MANAGE_DB.EXTERNAL_STAGES.JSONSTAGE
url='s3://bucketsnowflake-jsondemo';
```
- Details panel:** Shows the newly created schema "INTERNAL\_STAGES" with its definition, parent database "MANAGE\_DB", owner "ACCOUNTADMIN", and creation time "just now".
- Status bar:** Shows the message "Schema INTERNAL\_STAGES successfully created."



**Parsear**

Worksheets MODO VALIDACIÓN Y TRA... 3.2 Creación del stage y el ... 3.3 Parseo y análisis del fi... 3.4 Manejo de datos neste... 3.5 Jerarquías y flatten del ... 3.6 Inserción de datos no e... + ACCOUNT

**Databases Worksheets**

Inned (0) Pinned objects All Objects ...

BBDD\_VENTAS COPY\_DB MANAGE\_DB PRIMERABBDD SNOWFLAKE SNOWFLAKE\_SAMPLE\_DATA

```

COPY_DB.PUBLIC Settings
1 // Seleccionar atributo/columna
2
3 SELECT RAW_FILE:city FROM PRIMERABBDD.PRIMERESQUEMA.JSON_RAW
4
5 SELECT $1:first_name FROM PRIMERABBDD.PRIMERESQUEMA.JSON_RAW
6
7
8 // Seleccionar atributo/columna - formateo y conversión
9
10 SELECT RAW_FILE:first_name::string as first_name FROM PRIMERABBDD.PRIMERESQUEMA.JSON_RAW;
11
12 SELECT RAW_FILE:id::int as id FROM PRIMERABBDD.PRIMERESQUEMA.JSON_RAW;
13
14
15 SELECT
16   RAW_FILE:id::int as id,
17   RAW_FILE:first_name::STRING as first_name,
18   RAW_FILE:last_name::STRING as last_name,
19   RAW_FILE:gender::STRING as gender
20
FROM PRIMERABBDD.PRIMERESQUEMA.JSON_RAW;

```

↳ Results ↵ Chart

RAW_FILE:CITY
1 "Bakerfield"
2 "Louny"
3 "Wates"
4 "Umeå"
5 "Shangxian"
6 "Saurimo"
7 "Kanoar"

PRIMERABBDD SNOWFLAKE SNOWFLAKE\_SAMPLE\_DATA

```

10 SELECT RAW_FILE:first_name::string as first_name FROM PRIMERABBDD.PRIMERESQUEMA.JSON_RAW;
11
12 SELECT RAW_FILE:id::int as id FROM PRIMERABBDD.PRIMERESQUEMA.JSON_RAW;
13
14
15 SELECT
16   RAW_FILE:id::int as id,
17   RAW_FILE:first_name::STRING as first_name,
18   RAW_FILE:last_name::STRING as last_name,
19   RAW_FILE:gender::STRING as gender
20
FROM PRIMERABBDD.PRIMERESQUEMA.JSON_RAW;

```

↳ Results ↵ Chart

ID	FIRST_NAME	LAST_NAME	GENDER
1	Portia	Gioani	Male
2	Dag	Croney	Male
3	Heath	Lackmann	Female
4	Dita	Deering	Female
5	Nikki	McCosh	Female
6	Austina	Laux	Female
7	Tessa	Nairns	Female
8	Remington	Lilleman	Female
9	Genovera	Semered	Male

## Manejo de datos nested

```
COPY_DB.PUBLIC *      Settings *

1  | // Manejo de datos nested
2
3  SELECT RAW_FILE:job as job FROM PRIMERABBDD.PRIMERESQUEMA.JSON_RAW;
4
5
6  SELECT
7      RAW_FILE:job.salary::INT as salary
8  FROM PRIMERABBDD.PRIMERESQUEMA.JSON_RAW;
9

10
11
12  SELECT
13      RAW_FILE:first_name::STRING as first_name,
14      RAW_FILE:job.salary::INT as salary,
15      RAW_FILE:job.title::STRING as title
16  FROM PRIMERABBDD.PRIMERESQUEMA.JSON_RAW;
17
18
19  // Manejo de arrays
20
21  SELECT
22      RAW_FILE:prev_company as prev_company
23  FROM PRIMERABBDD.PRIMERESQUEMA.JSON_RAW;
24
25  SELECT
26      RAW_FILE:prev_company[1]::STRING as prev_company
27  FROM PRIMERABBDD.PRIMERESQUEMA.JSON_RAW;
28
29
30  SELECT
31      ARRAY_SIZE(RAW_FILE:prev_company) as prev_company
32  FROM PRIMERABBDD.PRIMERESQUEMA.JSON_RAW;
```

Worksheets MODO VALIDACIÓN Y TRA... 3.2 Creación del stage y el ... 3.3 Parseo y análisis del fic... 3.4 Manejo de datos neste... 3.5 Jerarquías y flatten del ... 3.6 Inserción de datos no e... +

Databases Worksheets

Pinned (0)

No pinned objects

All Objects

- BBDD\_VENTAS
- COPY\_DB
- MANAGE\_DB
- PRIMERABDD
- SNOWFLAKE
- SNOWFLAKE\_SAMPLE\_DATA

COPY\_DB.PUBLIC \* Settings \*

```
1 // Manejo de datos nested
2
3 | SELECT RAW_FILE:job as job FROM PRIMERABDD.PRIMERESQUEMA.JSON_RAW;
4
5
6 SELECT
7   RAW_FILE:job.salary::INT as salary
8   FROM PRIMERABDD.PRIMERESQUEMA.JSON_RAW;
9
10
11
12 SELECT
13   RAW_FILE:first_name::STRING as first_name,
14   RAW_FILE:job.salary::INT as salary,
15   RAW_FILE:job.title::STRING as title
16   FROM PRIMERABDD.PRIMERESQUEMA.JSON_RAW;
17
18
19 // Manejo de arrays
20
```

Results Chart

JOB
{ "salary": 32000, "title": "Financial Analyst" }
{ "salary": 43000, "title": "Clinical Specialist" }
{ "salary": 40800, "title": "Research Assistant I" }
{ "salary": 14400, "title": "Assistant Media Planner" }
{ "salary": 34600, "title": "Administrative Officer" }
{ "salary": 25800, "title": "Compensation Analyst" }
{ "salary": 30600, "title": "Data Coordinator" }
{ "salary": 22700, "title": "Pharmacist" }
{ "salary": 33200, "title": "Compensation Analyst" }

Query Details

Query duration

Rows

Query ID 0iacBaf6-

JOB

100% filled

```

4
5
6   SELECT
7     RAW_FILE:job.salary::INT as salary
8   FROM PRIMERABBDD.PRIMERESQUEMA.JSON_RAW;
9
10
11
12   SELECT
13     RAW_FILE:first_name::STRING as first_name,
14     RAW_FILE:job.salary::INT as salary,
15     RAW_FILE:job.title::STRING as title
16   FROM PRIMERABBDD.PRIMERESQUEMA.JSON_RAW;
17
18
19   // Manejo de arrays
20

```

↳ Results ~ Chart

	SALARY
1	32,000
2	43,000
3	40,800
4	14,400
5	34,600
6	25,800
7	30,600
8	22,700
9	33,200

Query Details  
Query duration  
Rows  
Query ID 01ac8  
  
SALARY  
-600

```

11
12   SELECT
13     RAW_FILE:first_name::STRING as first_name,
14     RAW_FILE:job.salary::INT as salary,
15     RAW_FILE:job.title::STRING as title
16   FROM PRIMERABBDD.PRIMERESQUEMA.JSON_RAW;
17
18
19   // Manejo de arrays
20

```

↳ Results ~ Chart

	FIRST_NAME	...	SALARY	TITLE
1	Portia		32,000	Financial Analyst
2	Dag		43,000	Clinical Specialist
3	Heath		40,800	Research Assistant I
4	Dita		14,400	Assistant Media Planner
5	Nikki		34,600	Administrative Officer
6	Austina		25,800	Compensation Analyst
7	Tessa		30,600	Data Coordinator
8	Remington		22,700	Pharmacist
9	Genovera		33,200	Compensation Analyst

Object

Selected objects

Objects ...

BBDD\_VENTAS

COPY\_DB

MANAGE\_DB

PRIMERABDD

SNOWFLAKE

SNOWFLAKE\_SAMPLE\_DATA

```

COPY_DB.PUBLIC * Settings *
13     RAW_FILE:first_name::STRING as first_name,
14     RAW_FILE:job.salary::INT as salary,
15     RAW_FILE:job.title::STRING as title
16   FROM PRIMERABDD.PRIMERESQUEMA.JSON_RAW;
17
18 // Manejo de arrays
19
20   SELECT
21     RAW_FILE:prev_company as prev_company
22   FROM PRIMERABDD.PRIMERESQUEMA.JSON_RAW;
23
24   SELECT
25     RAW_FILE:prev_company[1]::STRING as prev_company
26   FROM PRIMERABDD.PRIMERESQUEMA.JSON_RAW;
27
28
29   SELECT
30     ARRAY_SIZE(RAW_FILE:prev_company) as prev_company
31   FROM PRIMERABDD.PRIMERESQUEMA.JSON_RAW;
32

```

↳ Results ↵ Chart

PREV_COMPANY
1 null
2 Gerlach, Russel and Moen
3 Reynolds LLC
4 null
5 null
6 null
7 null
8 Lockman, Kunze and Bartoletti
9 null

## Hierarchies

... COPY\_DB.PUBLIC \* Settings \*

```

1   SELECT
2     RAW_FILE:spoken_languages as spoken_languages
3   FROM PRIMERABDD.PRIMERESQUEMA.JSON_RAW;
4
5   SELECT * FROM PRIMERABDD.PRIMERESQUEMA.JSON_RAW;
6
7 // Contamos el número de elementos del array
8   SELECT
9     array_size(RAW_FILE:spoken_languages) as spoken_languages
10    FROM PRIMERABDD.PRIMERESQUEMA.JSON_RAW;
11
12
13   SELECT
14     RAW_FILE:first_name::STRING as first_name,
15     array_size(RAW_FILE:spoken_languages) as spoken_languages
16    FROM PRIMERABDD.PRIMERESQUEMA.JSON_RAW;
17
18
19
20   SELECT

```

↳ Results ↵ Chart

SPOKEN_LANGUAGES
1 [ { "language": "Kazakh", "level": "Advanced" }, { "language": "Lao", "level": "Basic" } ]
2 [ { "language": "Assamese", "level": "Basic" }, { "language": "Papiamento", "level": "Expert" }, { "language": "Telugu", "level": "Basic" } ]
3 [ { "language": "Swati", "level": "Expert" } ]
4 [ { "language": "Chinese", "level": "Advanced" }, { "language": "Mongolian", "level": "Basic" } ]
5 [ { "language": "Filipino", "level": "Basic" }, { "language": "Kazakh", "level": "Basic" } ]
6 [ { "language": "Northern Sotho", "level": "Basic" }, { "language": "Tswana", "level": "Expert" }, { "language": "Georgian", "level": "Advanced" } ]
7 [ { "language": "Korean", "level": "Basic" } ]
8 [ { "language": "Kazakh", "level": "Expert" } ]
9 [ { "language": "Haitian Creole", "level": "Advanced" }, { "language": "Quechua", "level": "Basic" }, { "language": "Thai", "level": "Expert" } ]

## 4: Nube



## ¿Cómo cargamos datos desde la nube?



**Go to AWS S3 and create bucket:**

## Create bucket Info

Buckets are containers for data stored in S3. Learn more [↗](#)

### General configuration

#### Bucket name

Bucket name must be globally unique and must not contain spaces or uppercase letters. See rules for bucket naming [↗](#)

#### AWS Region



#### Copy settings from existing bucket - optional

Only the bucket settings in the following configuration are copied.

### Object Ownership Info

Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

**ACLs disabled (recommended)**

All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

**ACLs enabled**

Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

#### Object Ownership

Bucket owner enforced

### Block Public Access settings for this bucket

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its contents is blocked, turn on Block all public access. This will cause no public access to this bucket.

We upload the file netflix\_titles:

mibucketdecargasnowflake Info

[Objects](#) [Properties](#) [Permissions](#) [Metrics](#) [Management](#) [Access Points](#)

#### Objects (0)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) [↗](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more [↗](#)

[Actions](#)  [Create folder](#)

*Find objects by prefix*

Name	Type	Last modified	Size	Storage class
No objects				

You don't have any objects in this bucket.

Upload

Screenshot of Microsoft Excel showing a spreadsheet titled "netflix\_titles". The table contains data from the "show\_id", "type", "title", "director", "cast", "country", "date\_added", "release\_year", "rating", "duration", "listed\_in", and "description" columns. The data includes various movie and TV show entries with their respective details.

show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
s1,TV Show,3%, "João Miguel, Bianca Comparato, Michel Gomes, Rodolfo Valente, Vanesa Oliveira, Rafael Lozano, Viviane Porto, Mel Fronckowiak, Sergio Mamberti, Zezé Motta, Celso Frateschi",Brazil,"August 14, 2020",2020											
s2,Movie,7,19,Jorge Michel Grau,"Demián Bichir, Iñaki Bonilla, Oscar Serrano, Azalia Ortiz, Octavio Michel, Carmen Deatò",Mexico,"December 23, 2016",2016,TV-MA,93 min,"Dramas, International Movies","After a devastating											
s3,Movie,23,59,Gilbert Chan,"Teddy Chan, Stella Chung, Henley Hii, Lawrence Koh, Tommy Kuan, Josh Lai, Mark Lee, Susan Leong, Benjamin Lim",Singapore,"December 20, 2018",2018,R,78 min,"Horror Movies, International Movie											
s4,Movie,9,Shane Acker,"Elijah Wood, John C. Reilly, Jennifer Connelly, Christopher Plummer, Crispin Glover, Martin Landau, Fred Tatasciore, Alan Oppenheimer, Tom Kane",United States,"November 16, 2017",2009,PG-13,80 min											
s5,Movie,21,Robert Luketic,"Jim Sturgess, Kevin Spacey, Cole Bosworth, Aaron Yoo, Lisa Lapira, Jacob Pitts, Laurence Fishburne, Jack McGee, Josh Gad, Sam Golzari, Helen Carey, Jack Gilpin",United States,"January 1, 2020",2009,PG-13,100 min											
s6,TV Show,46,Serdar Akar,"Erdal Bozkurt, Yasemin Alifen, Melis Birkan, Sayın Soysal, Berkan Aşal, Metin Belgin, Ayşenur Eren, Selin Uludoğan, Ayşe Fecht, Suna Yıldız, Aşkın Yıldız",Turkey,"July 1, 2017",2016,TV-MA,1 Season											
s7,Movie,122,Yasir Al Yasiri,"Amina Khalil, Ahmed Dawood, Tarek Lotfy, Ahmed El Fishawy, Mahmoud Hijazi, Jihane Khalil, Asmaa Galal, Tara Emad",Egypt,"June 1, 2020",2019,TV-MA,95 min,"Horror Movies, International Movies"											
s8,Movie,167,Kevin Reynolds,"Samuel L. Jackson, John Heard, Kelly Rowan, Clifton Collins Jr., Tony Plana",United States,"November 1, 2019",1997,R,119 min,Dramas,"After one of his high school students attacks him, dedicated teacher Samuel L. Jackson must fight back to protect his students."											
s9,Movie,705,Shrawan Kumar,"Divya Dutta, Atul Kulkarni, Mohan Agase, Anupama Shyam, Raayo S. Bakirta, Yashvita Sancheti, Geetika Kansara, Archan Trivedi, Rajiv Pathak",India,"April 1, 2019",2019,TV-14,118 min,"Horror Movies"											
s10,Movie,1920,Vikram Bhatt,"Rajneesh Duggal, Adah Sharma, Indraneil Sengupta, Anjan Alagh, Rajendranath Zutshi, Vipin Sharma, Amin Hajee, Shri Vallabh Vyas",India,"December 15, 2017",2008,TV-MA,143 min,"Horror Movies"											
s11,Movie,1922,Zak Hillitch,"Thomas Jane, Molly Parker, Dylan Schmid, Kathryn Bernard, Bob Frazer, Brian d'Arcy James, Neal McDonough",United States,"October 20, 2017",2017,TV-MA,103 min,"Dramas, Thrillers","A farmer pen											
s12,TV Show,1983,"Robert Więckiewicz, Maciej Musiał, Michałina Olszańska, Andrzej Chyra, Clive Russell, Zofia Wicha, Edyta Olszańska, Mateusz Kościukiewicz, Ewa Bąszczyk, Vu Le Hong, Tomasz Wałosz, Krzysztof											
s13,TV Show,1994,Diego Enrique Osorno,,Mexico,"May 17, 2019",2019,TV-MA,1 Season,"Crime TV Shows, Documentaries, International TV Shows","Archival video and new interviews examine Mexican politics in 1994, a year marked											
s14,Movie,2,213,Nottaporn Boonprakob,Artivara Kongmalai,Thailand,"March 1, 2019",2018,TV-MA,89 min,"Documentaries, International Movies, Sports Movies","This intimate documentary follows rock star Artivara Kongmalai											
s15,Movie,3027,John Suits,"Omar Epps, Kate Walsh, Miranda Cosgrove, Angus Macfadyen, Jordie Fox, Enver Gjokaj, Haaz Sleiman",United States,"March 19, 2020",2019,R,91 min,"Independent Movies, Sci-Fi & Fantasy, Thrillers","St											
s16,Movie,Oct-01,Kunle Afolayan,"Sadig Daba, David Ballie, Kayode Olaiya, Kehinde Bankole, Fabian Adeoye Lojeide, Nick Rhys, Kunle Afolayan, Colin David Reese, Ibrahim Shatta, Femi Adebayo, Kanayo O. Kanayo, Lawrence Stu											
s17,TV Show,Feb-09,, "Shahd El Yaseen, Shaila Sabt, Hale, Hanadi Al-Kandari, Salma Salem, Ibrahim Al-Harbi, Mahmoud Boushrahi, Yousef Al Balushi, Ghorour, Abdullah Al-bloshi",, "March 20, 2019",2018,TV-14,1 Season,"Internati											
s18,Movie,22-Jul,Paul Greengrass,"Andrea Danielsen Lie, Jonny Gardener, Jonas Strand Gravli, Ola G. Furuseth, Maria Bock, Thorbjørn Harr, Jaden Smith",Norway, Iceland, United States,"October 10, 2018",2018,R,144 min,"Drama											
s19,Movie,15-Aug,Swapnaneel Jaykar,"Rahul Pethi, Mrunmayee Deshpande, Adinath Kothare, Vaibhav Mangal, Jaywant Wadkar, Satish Pulekar, Naina Aptekar, Uday Tikekar",India,"March 29, 2019",2019,TV-14,124 min,"Comedie											
s20,Movie,83,,Lee Dixon, Ian Wright, Paul Merson",United Kingdom,"May 16, 2018",2017,TV-PG,87 min,Sports Movies,"Mixing old footage with interviews, this is the story of Arsenal's improbable win versus Liverpool in the final											
s21,Movie,46,Kuch Bheege Alfaaz,Onir,"Geetanjali Thapa, Zain Khan Durran, Shiray Rai Tiwari, Mona Ambegaonkar, Chandreyee Ghosh, Barun Chanda, Saheb Bhattacharjee, Shefali Chauhan",India,"September 1, 2018",2018,T											
s22,Movie,46,Goli Soda 2,Vijay Milton,"Samuthirakan, Bharath Seenu, Vinoth, Esakki Barath, Chemban Vinod Jose, Gautham Menon, Krishna Kurup, Subiksha",India,"September 15, 2018",2018,TV-14,128 min,"Action & Adventure, I											
s23,Movie,46,Maj Rati 44,Keteki,Santvana Bardoloi,"Adil Hussain, Shakil Imtiaz, Mahendra Rabha, Sulakshana Baruah, Rahul Gautam Sarma, Kulada Bhattacharja, Indu Mohan Das, Moumita Talukdar",India,"September 15, 2018",2018,TV-14,100 min,"Dramas, Internati											
s24,Movie,46,Mayurakshi,Alau Ghosh,"Soumitra Chatterjee, Prasenjit Chatterjee, Indrani Haldar, Sudipta Chakraborty, Suman Banerjee, Gargi Roychowdhury",India,"September 15, 2018",2017,TV-14,100 min,"Dramas, Internati											
s25,TV Show,SAINT SEIYA: Knights of the Zodiac.,Brvson Bauzus, Emily Neves, Blake Sheard, Patrick Poole, Luci Christian, Adam Gibbs, Masakazu Morita, Fumiko Oriksa, Takahiro Sakurai, Hiroaki Miura, Satomi Sato, Katsuvu											

## We go now to IAM, Roles and create a new Role:

Screenshot of the AWS IAM console showing the "Roles" page. It displays two existing roles: "AWSServiceRoleForSupport" and "AWSServiceRoleForTrustedAdvisor". Both roles are associated with the "AWS Service: support" and "AWS Service: trustedadvisor" service-linked roles respectively. The "Temporary credentials" section is also visible.

**Step 2**  
Add permissions

**Step 3**  
Name, review, and create

**Trusted entity type**

- AWS service Allow AWS services like EC2, Lambda, or others to perform actions in this account.
- AWS account Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.
- Web identity Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.

**An AWS account**  
Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.

- This account (249207335272)
- Another AWS account

Account ID  
Identifier of the account that can use this role  
249207335272

Account ID is a 12-digit number.

**Options**

**Require external ID (Best practice when a third party will assume this role)**  
You can increase the security of your role by requiring an optional external identifier, which prevents "confused deputy" attacks. This is recommended if you do not own or have administrative access to the account that can assume this role. The external ID can include any characters that you choose. To assume this role, users must be in the trusted account and provide this exact external ID. [Learn more](#)

External ID  
00000

**Important:** The console does not support using an external ID with the Switch Role feature. If you select this option, entities in the trusted account must use the API, CLI, or a custom federation proxy to make cross-account iam:AssumeRole calls. [Learn more](#)

**Require MFA**  
Requires that the assuming entity use multi-factor authentication.

## Add permissions Info

**Permissions policies (Selected 1/847) Info**

Choose one or more policies to attach to your new role.

**Create policy**

Filter policies by property or policy name and press enter. 9 matches

**Clear filters**

Policy name	Type	Description
<input checked="" type="checkbox"/>  AmazonS3FullAccess	AWS m...	Provides full access to all buckets via the AWS Management Console.
<input type="checkbox"/>  AmazonS3ReadOnlyAccess	AWS m...	Provides read only access to all buckets via the AWS Management Console.
<input type="checkbox"/>  AmazonDMSRedshiftS3Role	AWS m...	Provides access to manage S3 settings for Redshift endpoints for DMS.
<input type="checkbox"/>  QuickSightAccessForS3StorageManagementAnalytics	AWS m...	Policy used by QuickSight team to access customer data produced by S3 Storage Management Analytics.
<input type="checkbox"/>  AmazonS3OutpostsFullAccess	AWS m...	Provides full access to Amazon S3 on Outposts via the AWS Management Console.
<input type="checkbox"/>  AmazonS3OutpostsReadOnlyAccess	AWS m...	Provides read only access to Amazon S3 on Outposts via the AWS Management Console.
<input type="checkbox"/>  AmazonS3ObjectLambdaExecutionRolePolicy	AWS m...	Provides AWS Lambda functions permissions to interact with Amazon S3 Object Lambda. Also grants Lambda permissions to write to ...
<input type="checkbox"/>  AWSBackupServiceRolePolicyForS3Restore	AWS m...	Policy containing permissions necessary for AWS Backup to restore a S3 backup to a bucket. This includes read/write permissions to ...
<input type="checkbox"/>  AWSBackupServiceRolePolicyForS3Backup	AWS m...	Policy containing permissions necessary for AWS Backup to backup data in any S3 bucket. This includes read access to all S3 objects...

### ► Set permissions boundary - optional Info

Set a permissions boundary to control the maximum permissions this role can have. This is not a common setting, but you can use it to delegate permission management to others.

IAM > Roles

**Roles (3) Info**

An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

**Create role**

Search

Role name	Trusted entities	Last activity
<input type="checkbox"/> AWSServiceRoleForSupport	AWS Service: support (Service-Linked Role)	-
<input type="checkbox"/> AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service-Linked Role)	-
<input checked="" type="checkbox"/> Snowflake_S3	Account: 249207335272	-

**we create the script for Snowflake:**

Databases Worksheets

Pinned (0)

No pinned objects

All Objects

- BBDD\_VENTAS
- COPY\_DB
- MANAGE\_DB
- PRIMERABDD
- SNOWFLAKE
- SNOWFLAKE\_SAMPLE\_DATA

```

MANAGE_DB.PUBLIC * Settings *

1 // Crear objeto integración
2
3 create or replace storage integration s3_int
4   TYPE = EXTERNAL_STAGE
5   STORAGE_PROVIDER = S3
6   ENABLED = TRUE
7   STORAGE_AWS_ROLE_ARN = ''
8   STORAGE_ALLOWED_LOCATIONS = ('s3://<your-bucket-name>/<your-path>')
9   COMMENT = 'Comentario opcional'
10
11
12 // Verificar propiedades para coger el external_id y actualizarlo en S3
13 DESC integration s3_int;
14
15
16
17 // Crear tabla objetivo
18 CREATE OR REPLACE TABLE PRIMERABDD.PRIMERESQUEMA.movie_titles (
19   show_id STRING,
20   type STRING,
21   title STRING,
22   director STRING,
23   cast STRING,
24   country STRING,
25   date_added STRING,
26   release_year STRING,
27   rating STRING,
28   duration STRING,
29   listed_in STRING,
30   description STRING )
31
32
33 // Crear objeto formato de fichero
34 CREATE OR REPLACE file format MANAGE_DB.file_formats.csv_fileformat
35   type = csv
36   field delimiter = '*'

```

We copy the ARN code from AWS and paste it in snowflake:

Identity and Access Management (IAM) x

IAM > Roles > Snowflake\_S3

Snowflake\_S3

Summary

[Edit](#)

Creation date	May 31, 2023, 11:31 (UTC+02:00)	ARN	<a href="#">arn:aws:iam:249287335272:role/Snowflake_S3</a>
Last activity	None	Maximum session duration	1 hour

[Link to switch roles in console](#)

[https://signin.aws.amazon.com/switchrole?roleName=Snowflake\\_S3&acccode=249287335272](https://signin.aws.amazon.com/switchrole?roleName=Snowflake_S3&acccode=249287335272)

[Permissions](#) [Trust relationships](#) [Tags](#) [Access Advisor](#) [Revoke sessions](#)

**Permissions policies (1) [Info](#)**

You can attach up to 10 managed policies.

[Filter policies by property or policy name and press enter.](#)

Policy name	Type	Description
AmazonS3FullAccess	AWS managed	Provides full access to all buckets via the AWS Management Console.

**Permissions boundary - (not set) [Info](#)**

Set to control the maximum permissions this role can have. This is not a common setting but can be used to limit what the role can do when shared with others.

[Boundary](#)

```

MANAGE_DB.PUBLIC *      Settings *

1 // Crear objeto integración
2
3 create or replace storage integration s3_int
4   TYPE = EXTERNAL_STAGE
5   STORAGE_PROVIDER = S3
6   ENABLED = TRUE
7   STORAGE_AWS_ROLE_ARN = 'arn:aws:iam::249287335272:role/Snowflake_S3'
8   STORAGE_ALLOWED_LOCATIONS = ('s3://<your-bucket-name>/<your-path>/')
9   COMMENT = 'Comentario opcional'
10
11
12 // Verificar propiedades para coger el external_id y actualizarlo en S3
13 DESC integration s3_int;
14
15
16
17 // Crear tabla objetivo
18 CREATE OR REPLACE TABLE PRIMERABBDD.PRIMERESQUEMA.movie_titles (
19   show_id STRING,
20   type STRING,
21   title STRING,
22   director STRING,
23   cast STRING,
24   country STRING,
25   date_added STRING,
26   release_year STRING,
27   rating STRING,
28   duration STRING,
29   listed_in STRING,
30   description STRING )
31
32
33 // Crear objeto formato de fichero
34 CREATE OR REPLACE file format MANAGE_DB.file_formats.csv_fileformat
35   type = CSV
36   field_delimiter = ","
37   skip_header = 1
38   null_if = ('NULL', 'null')

```

## We create an integration object:

The screenshot shows the Snowflake interface with the 'Worksheets' tab selected. On the left, the sidebar lists databases like 'PRIMERABBDD' and objects such as 'COPY\_DB', 'MANAGE\_DB', and 'PRIMERESQUEMA'. The main workspace displays the SQL code for creating the 's3\_int' integration. The code is highlighted in blue, indicating it's being executed. Below the code, the 'Results' tab is active, showing a single row with the status 'Integration S3\_INT successfully created.'

```

abases Worksheets
ed (0)
Sinned objects
All Objects ...
3 Bbdd_VENTAS
3 COPY_DB
3 MANAGE_DB
3 PRIMERABBDD
3 SNOWFLAKE
3 SNOWFLAKE_SAMPLE_DATA
MANAGE_DB.PUBLIC *      Settings *
1 // Crear objeto integración
2
3 create or replace storage integration s3_int
4   TYPE = EXTERNAL_STAGE
5   STORAGE_PROVIDER = S3
6   ENABLED = TRUE
7   STORAGE_AWS_ROLE_ARN = 'arn:aws:iam::249287335272:role/Snowflake_S3'
8   STORAGE_ALLOWED_LOCATIONS = ('s3://mibucketdecargasnowflake/')
9   COMMENT = 'Comentario opcional'
↳ Results ↳ Chart
status
1 Integration S3_INT successfully created.

```

## We copy the ARN value and paste it in IAM roles Snowflake\_3 trust policy:

1Y show\_id SINKING,  
 20 type STRING,  
 21 +-----+  
 ↵ Results ↵ Chart

property	property_type	property_value	property_default	...
ENABLED	Boolean	true	false	
STORAGE_PROVIDER	String	S3		
STORAGE_ALLOWED_LOCATIONS	List	s3://mbucketdecargasnowflake/		
STORAGE_BLOCKED_LOCATIONS	List			
STORAGE_AWS_IAM_USER_ARN	String	arn:aws:iam::220432643186:user/la790000-s		
STORAGE_AWS_ROLE_ARN	String	arn:aws:iam::249287335272:role/Snowflake_S3		
STORAGE_AWS_EXTERNAL_ID	String	KK93714_SFRole=2_2nMKq+jxIXEyN/zEtDDPOH5CMQ=		
COMMENT	String	Comentario opcional		

Announcements    Reviews    Learning tools

IAM > Roles > Snowflake\_S3 > Edit trust policy

### Edit trust policy

1 • {  
 2     "Version": "2012-10-17",  
 3     "Statement": [  
 4       {  
 5         "Effect": "Allow",  
 6         "Principal": {  
 7           "AWS": "arn:aws:iam::220432643186:user/la790000-s"  
 8         },  
 9         "Action": "sts:AssumeRole",  
 10        "Condition": {  
 11          "StringEquals": {  
 12             "sts:ExternalId": "00000"  
 13          }  
 14        }  
 15      }  
 16     }  
 17 }

**Edit statement** Remove

1. Add actions for STS

Filter actions

All actions (sts:\*)

Access level - read

- GetAccessKeyInfo
- GetCallerIdentity
- GetFederationToken
- GetServiceBearerToken
- GetSessionToken

Access level - read or write

- AssumeRole
- AssumeRoleWithSAML
- AssumeRoleWithWebIdentity
- DecodeAuthorizationMessage

2. Add a principal

Add

3. Add a condition (optional)

Add

+ Add new statement

JSON Ln 7, Col 53

Security: 0 Errors: 0 Warnings: 0 Suggestions: 0 Preview external access

We also copy from Snowflake and paste in AWS the ExternalID:

## Edit trust policy

```
1+ {
2    "Version": "2012-10-17",
3+    "Statement": [
4+        {
5            "Effect": "Allow",
6+            "Principal": {
7                "ARN": "arn:aws:iam::220432643186:user/la790000-s"
8            },
9            "Action": "sts:AssumeRole",
10           "Condition": {
11               "StringEquals": {
12                   "sts:ExternalId": "KK93714_SFCRole=2_2nWKq+jx1XEyN/zEtODPOf5CNPQ="
13               }
14           }
15       }
16   ]
17 }
```

I

## We create our objective table in Snowflake:

```
MANAGE_DB.PUBLIC ▾      Settings ▾

16 // Crear tabla objetivo
17 CREATE OR REPLACE TABLE PRIMERABBDD.PRIMERESQUEMA.movie_titles (
18     show_id STRING,
19     type STRING,
20     title STRING,
21     director STRING,
22     cast STRING,
23     country STRING,
24     date_added STRING,
25     release_year STRING,
26     rating STRING,
27     duration STRING,
28     listed_in STRING,
29     description STRING )
30
31
32 // Crear objeto formato de fichero
33 CREATE OR REPLACE file format MANAGE_DB.file_formats.csv_fileformat
34     type = csv
35     field_delimiter = ','
36     skip_header = 1
37     null_if = ('NULL', 'null')
38     empty_field_as_null = TRUE;
39
40
41 // Crear stage con el objeto de integración y el objeto de formato
42 CREATE OR REPLACE stage MANAGE_DB.external_stages.csv_aws
43     URL = 's3://<your-bucket-name>/'
44     STORAGE_INTEGRATION = s3_int
45     FILE_FORMAT = MANAGE_DB.file_formats.csv_fileformat
46
47
48 // Usar comando COPY para cargar en la tabla objetivo
49 COPY INTO PRIMERABBDD.PRIMERESQUEMA.movie_titles
50     FROM @MANAGE_DB.external_stages.csv_aws
51
52
53
54 // Crear de nuevo el objeto formato de fichero para indicar que hay campos que tienen el carácter " y si encuentra cr
```

## We update the bucket in Snowflake with AWS bucket:

```

1 // Crear stage con el objeto de integración y el objeto de formato
2 CREATE OR REPLACE stage MANAGE_DB.external_stages.csv_aws
3   URL = 's3://mibucketdecargasnowflake/'
4   STORAGE_INTEGRATION = s3_int
5   FILE_FORMAT = MANAGE_DB.file_formats.csv_fileformat
6
7
8 // Usar comando COPY para cargar en la tabla objetivo
9 COPY INTO PRIMERABBDD.PRIMERESQUEMA.movie_titles
10   FROM @MANAGE_DB.external_stages.csv_aws
11
12
13 // Crear de nuevo el objeto formato de fichero para indicar que hay campos que tienen el caracte

```

We have quotes and commas inside the csv file, we have to tell Snowflake that when he encounters quotes count it as a unit (enclosed):

```

// Crear de nuevo el objeto formato de fichero para indicar que hay campos que tienen el carácter " y si encuentra comas no las tenga en cuenta
CREATE OR REPLACE file format MANAGE_DB.file_formats.csv_fileformat
  type = csv
  field_delimiter = ','
  skip_header = 1
  null_if = ('NULL','null')
  empty_field_as_null = TRUE
  FIELD_OPTIONALLY_ENCLOSED_BY = '"'

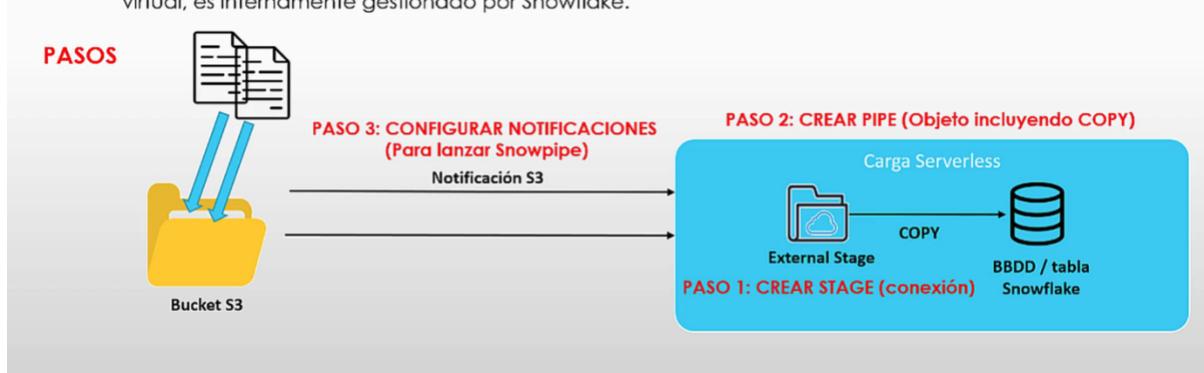
```

A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description	
2	s1	TV Show	3%	"João Miguel, Bianca Comparato, Michel Gomes, Rodolfo Valente, Vaneza Oliveira, Rafael Lozano, Viviane Porto, Mel Fronckowiak, Sergio Mamberti, Zezé Motta, C									
3	Fantasy	"In a future where the elite inhabit an island paradise far from the crowded slums, you get one chance to join the 3% saved from squalor."											
4	s3	Movie	23:59	Gilbert Chan, Tedd Chan, Stella Chung, Henley Hii, Lawrence Koh, Tommy Kuan, Josh Lai, Mark Lee, Susan Leong, Benjamin Lim	Singapore	"December 20, 2018"	2011	R					
5	s4	Movie	9	Shane Acker, Elijah Wood, John C. Reilly, Jennifer Connelly, Christopher Plummer, Crispin Glover, Martin Landau, Fred Tatasciore, Alan Oppenheimer, Tom Kane	United States								
6	s5	Movie	21	Robert Luketic, Jim Sturgess, Kevin Spacey, Kate Bosworth, Aaron Yoo, Liza Lapira, Jacob Pitts, Laurence Fishburne, Jack McGee, Josh Gad, Sam Golzari, Helen Carey, Jack Gi									
7	s6	TV Show	46	Serdar Akar, Erdal Boz, Ayşe Küçük, Yasemin Alton, Melis Birkan, Saygın Soysal, Berkan Aşal, Metin Belgin, Ayşegül Eren, Selin Uludoğan, Ayşay Fecht, Suna Yıldız									
8	s7	Movie	122	Yasir Al Yasiri, Amna Khalil, Ahmed Dawood, Tarek Lotfy, Ahmed El Fishawy, Mahmoud Hijazi, Jihane Khalil, Asmaa Galal, Tara Emad	Egypt	"June 1, 2020"	2019	TV-MA	9:				
9	s8	Movie	167	Kevin Reynolds, Samuel L. Jackson, John Heard, Kelly Rowan, Clifton Collins Jr., Tony Plana	United States	"November 1, 1997"	1997	R	119 min	Dramas	"After one of his h		
10	s9	Movie	706	Shrawan Kumar, Divya Dutta, Atul Kulkarni, Mohan Agashe, Anupam Shyam, Raayo S. Bakhira, Yashvita Sancheti, Greeva Kansara, Archana Trivedi, Rajiv Pathak	India	"April							
11	s10	Movie	1920	Vikram Bhatt, Rajneesh Duggal, Adah Sharma, Indraneil Sengupta, Anjori Alagh, Rajendranath Zutshi, Vipin Sharma, Amin Hajee, Shri Vallabh Vyas	India	"December 1							
12	s11	Movie	1922	Zek Hilditch, Thomas Jane, Molly Parker, Dylan Schmid, Caitlyn Bernard, Bob Frazer, Brian d'Arcy James, Neal McDonough	United States	"October 20, 2017"	2017	TV-M					
13	s12	TV Show	1983	Robert Więckiewicz, Maciej Musiał, Michałina Olszańska, Andrzej Chyra, Clive Russell, Zofia Wicha, Edyta Olszak, Mateusz Kościukiewicz, Ewa Bajaszczak									
14	s13	TV Show	1994	Diego Enrique Osorno	Mexico	"May 17, 2019"	2019	TV-MA	1 Season	Crime TV Shows, Documentaries, International TV Shows	"Archival video and new interviews exa		
15	s14	Movie	"2,215"	Nottapon Boonprakob, Artiware Kongmalai	Thailand	"March 1, 2019"	2018	TV-MA	89 min	Documentaries, International Movies, Sports Movies	"This intimate docu		
16	s15	Movie	3022	John Suits, Omar Epps, Kate Walsh, Miranda Cosgrove, Angus Macfadyen, Jorja Fox, Enver Gjokaj, Haaz Sleiman	United States	"March 19, 2020"	2019	R	91 min	Independence Day: Resurgence	"Independence Day: Resurgence"		
17	s16	Movie	Oct-01	Kunle Afolayan, Sadig Daba, David Baile, Kayode Olaiya, Kehinde Bankole, Fabian Adeoye, Lojede, Nick Rhys, Kunle Afolayan, Colm David Reese, Ibrahim Shatta, Fen									
18	s17	TV Show	Feb-09	Shahid Al Yaseen, Shalla Sabt, Hala, Hanadi Al-Kandari, Salma Salem, Ibrahim Al-Harbi, Mahmoud Boushahri, Yousef Al Balushi, Ghorour, Abdullah Al-bloshi									
19	s18	Movie	22-Jul	Paul Greengrass, Anders Danielsen Lie, Jonny Åberg, Jonas Strand Gravli, Ola G. Furseth, Maria Bock, Thorbjørn Harr, Jaden Smith	Norway, Iceland, United States								
20	s19	Movie	15-Aug	Swapnanee Jaykar, Rahul Pethé, Mrunmayee Deshpande, Adinath Kothare, Vaibhav Mangale, Jaywant Wadkar, Satish Pulekar, Naina Apte, Uday Tikekar	India	"May 16, 2018"	2017	TV-PG	87 min	Sports Movies	"Mixing old footage with interviews, this is the story of Arsenal's		
21	s20	Movie	'69	Lee Dixon, Ian Wright, Paul Merson	United Kingdom	"May 16, 2018"	2017	TV-PG	87 min	Sports Movies	"Mixing old footage with interviews, this is the story of Arsenal's		
22	s21	Movie	„Kuch Bheege Alfaaz	Onir, Geetanjali Thapu, Zain Khan Durran, Shrey Rai Tiwari, Mona Ambegaonkar, Chandreyee Ghosh, Barun Chanda, Saheb Bhattacharjee, Shefali									
23	s22	Movie	„Goli Soda“	Vijay Milton, Samuthirakani, Bharath Seeni, Vinoth, Esakkir Barath, Chemban Vinod Jose, Gautham Menon, Krishna Kurup, Subiksha	India	"September 15, 2018"							
24	s23	Movie	„Maj Rati“	Keteki, Santwana Bardoloi, Adil Hussain, Shakil Imtiaz, Mahendra Rabha, Sulakshana Baruah, Rahul Gautam Samra, Kulade Bhattacharjya, Indu Mohan Das									
25	s24	Movie	„Mayurakshi“	Atanu Ghosh, Soumitra Chatterjee, Prasenjit Chatterjee, Indrani Halder, Sudipta Chakraborty, Suman Banerjee, Gargi Roychowdhury	India	"September 15, 2018"							

## Snowpipe

## ¿Qué es Snowpipe y los pasos para la creación de pipes?

- **Snowpipe** permite la ingesta de datos **tan pronto están disponibles** en el stage (por ejemplo, en el bucket de la plataforma Cloud).
- Muy útil si necesitamos que los datos estén **disponibles** de inmediato para el análisis.
- Snowpipe utiliza funcionalidades "**serverless**", es decir, no precisa de un data warehouse virtual, es internamente gestionado por Snowflake.



We create a stage for Snowpipe:

```
MANAGE_DB.PIPES *      Settings *

1 //5.2 Creación de stage para Snowpipe
2
3 // Creación de la tabla
4 CREATE OR REPLACE TABLE PRIMERABBDD.PRIMERESQUEMA.employees (
5   id INT,
6   first_name STRING,
7   last_name STRING,
8   email STRING,
9   location STRING,
10  department STRING
11 )
12
13
14 // Creación objeto formato fichero
15 CREATE OR REPLACE file format MANAGE_DB.file_formats.csv_fileformat
16   type = csv
17   field_delimiter = ','
18   skip_header = 1
19   null_if = ('NULL','null')
20   empty_field_as_null = TRUE;
21
22
23 // Crear stage con objeto integración y objeto formato fichero
24 CREATE OR REPLACE stage MANAGE_DB.external_stages.csv_folder
25   URL = 's3://mibucketdecargasnowflake/snowpipe'
26   STORAGE_INTEGRATION = s3_int
27   FILE_FORMAT = MANAGE_DB.file_formats.csv_fileformat
28
29
30 // Visualizar los ficheros que hay en el stage
--
```

We will upload the following csv:

A1	B	C	D	E	F	G	H
1	id,first_name,last_name,email,location,department						
2	1,Manda,Birdall,mbirdall0@odnoklassniki.ru,Xinjiang,Research and Development						
3	2,Gaby,Slemonds,gslemonds1@elpais.com,KÃ©falos,Sales						
4	3,Monah,MacKniely,mmackniely2@bloglovin.com,Shaoxing,Engineering						
5	4,Klarika,Corbie,kcorbie3@technorati.com,SaronÃ·da,Legal						
6	5,Sayre,Stubblings,sstubbings4@miltbelan.gov.cn,Itapevi,Product Management						
7	6,Emmi,Sabberton,esabberton5@latimes.com,Chirpan,Business Development						
8	7,Herbie,Abadam,habadam6@dion.ne.jp,Sidi Redouane,Training						
9	8,Cointon,Carverhill,ccarverhill7@hugedomains.com,Nauchnyy Gorodok,Training						
10	9,Martelle,Ghirardi,mghirardi8@mapquest.com,Ocongate,Research and Development						
11	10,Analise,Thackwray,athackwray9@ycombinator.com,Meishan,Accounting						
12	11,Powell,Pomfret,ppomfreta@marriott.com,RyÅgasaki,Business Development						
13	12,Lola,Zincke,lzincke@illinois.edu,Nagbalaye,Engineering						
14	13,Jillarye,Maddern,jmaddernc@smugmug.com,Los Ãngeles,Business Development						
15	14,Bobinette,Bene,bbened@phpbb.com,Baozhu,Legal						
16	15,Marion,Copcutt,mcopcutte@businessweek.com,Javarthushuu,Human Resources						
17	16,Llewellyn,Josephy,ljosephyf@tamu.edu,Canis,Product Management						
18	17,Claretta,Carder,ccarderg@foxnews.com,Krasni Okny,Services						
19	18,Deanne,McCartan,dmccartanh@ifeng.com,Pukkila,Marketing						
20	19,Linoel,Grieve,lgrieve@ucoz.com,Nong Hi,Marketing						
21	20,Drusilla,Borrott,dborrottj@globo.com,Tambak,Accounting						
22	21,Kristi,Stormont,kstormontk@berkeley.edu,Santa MarÃ·a de Caparo,Support						
23	22,Mischa,Klemensiewicz,mklemensiewiczl@alibaba.com,Shixi,Business Development						
24	23,Goddart,Odd,goddm@guardian.co.uk,Fresno,Accounting						
25	24,Muffin,Waycott,mwaycottn@nytimes.com,Ugljevik,Research and Development						
26	25,Minetta,Cogle,mcogleo@springer.com,Riit,Research and Development						
27	26,Claudian,Kinnoch,ckinnochp@typepad.com,Pandan,Engineering						
28	27,Alfie,Golling,agollinq@craigslist.org,Dianbu,Training						
29	28,Bernadina,Ramsier,bramsierr@jigsy.com,DÃ«Äb,Human Resources						
30	29,Fonzie,Thunnercliff,fthunnercliffs@examiner.com,Manas,Product Management						
31	30,Ailina,Burrass,aburraast@psu.edu,Amsterdam Noord west,Legal						
32	31,Ronnie,Baldassi,rbaldassi@narod.ru,NÃ¤ssjÃ¶,Training						
33	32,Cornie,Blanko,cblankov@dropbox.com,Kostanay,Business Development						
34	33,Clea,Trevenu,ctrevenu@booking.com,Algeciras,Engineering						
35	34,Trish,Dommett,tdommettx@wufoo.com,Jilili,Services						

We create table:

```

1 // D.2 CreaciÃ³n de stage para snowpipe
2
3 // CreaciÃ³n de la tabla
4 CREATE OR REPLACE TABLE PRIMERABBDD.PRIMERESQUEMA.employees (
5   id INT,
6   first_name STRING,
7   last_name STRING,
8   email STRING,
9   hire_date STRING

```

Results

	status
1	Table EMPLOYEES successfully created.

We execute the file format:

```
8     email STRING,  
9  
↳ Results  ↳ Chart
```

	status
1	File format CSV_FILEFORMAT successfully created.

we create the stage:

```
// Crear stage con objeto integración y objeto formato fichero  
CREATE OR REPLACE stage MANAGE_DB.external_stages.csv_folder  
    URL = 's3://mibucketdecargasnowflake/snowpipe'  
    STORAGE_INTEGRATION = s3_int  
    FILE_FORMAT = MANAGE_DB.file_formats.csv_fileformat
```

```
↳ Results  ↳ Chart
```

	status	...
1	Stage area CSV_FOLDER successfully created.	

We visualize all files in the stage:

```
// Visualizar los ficheros que hay en el stage  
LIST @MANAGE_DB.external_stages.csv_folder  
  
// Creación de un esquema para organizar los pipes  
CREATE OR REPLACE SCHEMA MANAGE_DB.pipes
```

	name	size	md5
1	s3://mibucketdecargasnowflake/snowpipe/employee_data_1.csv	6,524	566dbe5c262acf70c7a68d62aa71

We create and describe the pipes:

```
// Definir el pipe
CREATE OR REPLACE pipe MANAGE_DB.pipes.employee_pipe
|auto_ingest = TRUE
AS
COPY INTO PRIMERABBDD.PRIMERESQUEMA.employees
FROM @MANAGE_DB.external_stages.csv_folder
```

// Descripción del pipe  
DESC pipe employee\_pipe

SELECT \* FROM PRIMERABBDD.PRIMERESQUEMA.employees

↳ Results    ↵ Chart

	status
1	Pipe EMPLOYEE_PIPE successfully created.

53 // Descripción del pipe  
54 DESC pipe employee\_pipe  
55  
56

↳ Results    ↵ Chart

	...	created_on	name	database_name	schema_name	def
1	2023-06-05 00:25:48,273 -0700		EMPLOYEE_PIPE	MANAGE_DB	PIPES	CO

We create an event notification in our bucket:

## Create event notification Info

To enable notifications, you must first add a notification configuration that identifies the events you want Amazon S3 to publish and the destinations where you want Amazon S3 to send the notifications.

### General configuration

#### Event name

Event name can contain up to 255 characters.

#### Prefix - optional

Limit the notifications to objects with key starting with specified characters.

#### Suffix - optional

Limit the notifications to objects with key ending with specified characters.

### Event types

Specify at least one event for which you want to receive notifications. For each group, you can choose an event type for all events, or you can choose individual events.

## Event types

Specify at least one event for which you want to receive notifications. For each group, you can choose an event type for all events, or you can choose one or more individual events.

### Object creation

- All object create events  
s3:ObjectCreated:"

- Put  
s3:ObjectCreated:Put
- Post  
s3:ObjectCreated:Post
- Copy  
s3:ObjectCreated:Copy
- Multipart upload completed  
s3:ObjectCreated:CompleteMultipartUpload

### Object removal

- All object removal events  
s3:ObjectRemoved:"

- Permanently deleted  
s3:ObjectRemoved:Delete
- Delete marker created  
s3:ObjectRemoved:DeleteMarkerCreated

### Object restore

- All restore object events  
s3:ObjectRestore:"

- Restore initiated  
s3:ObjectRestore:Post
- Restore completed  
s3:ObjectRestore:Completed
- Restored object expired  
s3:ObjectRestore:Delete

### Object ACL

-  Object ACL events

We copy the notification channel from snowflake:

COPY INTO PRTMERARRDD_PRTMERESQUEMA_employees	
51	
1 COUNTADMIN	arn:aws:sqs:eu-west-3:220432643186:sf-snowpipe-AIDATGUWPGRZJMSYI3Y5Y-UtgMHof55GWjUPoouRETg

And paste it in AWS:

**Destination**

① Before Amazon S3 can publish messages to a destination, you must grant the Amazon S3 principal the necessary permissions to call the relevant API to publish messages to an SNS topic, an SQS queue, or a Lambda function. [Learn more](#)

**Destination**  
Choose a destination to publish the event. [Learn more](#)

**Lambda function**  
Run a Lambda function script based on S3 events.

**SNS topic**  
Fanout messages to systems for parallel processing or directly to people.

**SQS queue**  
Send notifications to an SQS queue to be read by a server.

**Specify SQS queue**

**Choose from your SQS queues**

**Enter SQS queue ARN**

**SQS queue**

[Cancel](#) [Save changes](#)

We upload the csv employee\_data\_1.csv in AWS:

**Upload: status**

The information below will no longer be available after you navigate away from this page.

**Summary**

Destination	Succeeded	Failed
s3://mibucketdecargasnowflake/snowpipe/	1 file, 6.4 KB (100.00%)	0 files, 0 B (0%)

[Files and folders](#) [Configuration](#)

**Files and folders (1 Total, 6.4 KB)**

Name	Folder	Type	Size	Status	Error
employee_data_1.csv	-	text/csv	6.4 KB	Succeeded	-

wait around 1min and execute the command:

```

59
60
61 //----5.4 Configuración de notificaciones en AWS / Azure / GCP
62 | SELECT * FROM PRIMERABDD.PRIMERESQUEMA.employees
63
64
65 // 5.5 Gestión de pipes
66
67
68 DESC pipe MANAGE_DB.pipes.employee_pipe;

47 // Definir el pipe
48 CREATE OR REPLACE pipe MANAGE_DB.pipes.employee_pipe

```

↳ Results ~ Chart

	ID	FIRST_NAME	LAST_NAME	EMAIL	LOCATION	...	DEPARTMENT
1	1	Manda	Birdall	mbirdall0@odnoklassniki.ru	Xinjiang	✉	Research and Development
2	2	Gaby	Slemonds	gslemonds1@elpais.com	Kéfalos		Sales
3	3	Monah	MackNiely	mmackniely2@bloglovin.com	Shaoxing		Engineering
4	4	Klarika	Corbie	kcorbie3@technorati.com	Saronida		Legal
5	5	Sayre	Stubblings	sstubbings4@miiitbeian.gov.cn	Itapevi		Product Management
6	6	Emmi	Sabberton	esabberton5@latimes.com	Chirpan		Business Development
7	7	Herbie	Abadam	habadam6@dion.ne.jp	Sidi Redouane		Training
8	8	Cointon	Carverhill	ccarverhill7@hugedomains.com	Nauchnyy Gorodok		Training
9	9	Martelle	Ghirardi	mghirardi8@mapquest.com	Ocongate		Research and Development
10	10	Analise	Thackwray	athackwray9@ycombinator.com	Meishan		Accounting
11	11	Powell	Pomfret	ppomfreta@marriott.com	Ryūgasaki		Business Development
12	12	Lola	Zincke	lzincke@illinois.edu	Nagbalaye		Engineering

we upload a second file employee\_data\_2.csv:

Files and folders (1 Total, 6.2 KB)

Find by name

Name	Folder	Type	Size	Status
employee_data_2.csv	-	text/csv	6.2 KB	✔ Succeeded

We want a description of all pipes from our folder:

```
MANAGE_DB.PIPES * Settings *
```

```
--  
64  
65 // 5.5 Gestión de pipes  
66  
67  
68 DESC pipe MANAGE_DB.pipes.employee_pipe;  
69  
70 SHOW PIPES;  
71
```

↳ Results ↵ Chart

	...	created_on	name	database_name
1	2023-06-05 00:25:43.273 -0700		EMPLOYEE_PIPE	MANAGE_DB

We execute the following commands:

```
--  
64  
65 // 5.5 Gestión de pipes  
66  
67  
68 DESC pipe MANAGE_DB.pipes.employee_pipe;  
69  
70 SHOW PIPES;  
71  
72 SHOW PIPES like '%employee%'  
73  
74 SHOW PIPES in database MANAGE_DB  
75  
76 SHOW PIPES in schema MANAGE_DB.pipes  
77  
78 SHOW PIPES like '%employee%' in Database MANAGE_DB  
79
```

↳ Results ↵ Chart

	...	created_on	name
1	2023-06-05 00:25:43.273 -0700		EMPLOYEE_PIPE

We execute the table:

```
MANAGE_DB.PIPES ▾      Settings ▾

80
81
82 -- Modificar pipe (alterar stage o file format) --
83
84 // Preparación de la tabla 2
85 CREATE OR REPLACE TABLE PRIMERABBDD.PRIMERESQUEMA.employees2 (
86     id INT,
87     first_name STRING,
88     last_name STRING,
89     email STRING,
90     location STRING,
91     department STRING
92 )
93
94
95 // Pausar pipe
96 ALTER PIPE MANAGE_DB.pipes.employee_pipe SET PIPE_EXECUTION_PAUSED = true
97
98
99 // Verificar que el pipe está en pausa y pendingFileCount = 0
100 SELECT SYSTEM$PIPE_STATISTICS FROM INFORMATION_SCHEMA.PIPES
```

↳ Results ⚡ Chart

	status
1	Table EMPLOYEES2 successfully created.

We execute the following commands:

```

MANAGE_DB.PIPES Settings

94
95 // Pausar pipe
96 ALTER PIPE MANAGE_DB.pipes.employee_pipe SET PIPE_EXECUTION_PAUSED = true
97
98
99 // Verificar que el pipe está en pausa y pendingFileCount = 0
100 SELECT SYSTEM$PIPE_STATUS('MANAGE_DB.pipes.employee_pipe')
101
102 // Recrear pip y cambiar definición en el COPY
103 CREATE OR REPLACE pipe MANAGE_DB.pipes.employee_pipe
104 auto_ingest = TRUE
105 AS
106 COPY INTO PRIMERABBDD.PRIMERESQUEMA.employees2
107 FROM @MANAGE_DB.external_stages.csv_folder
108
109 ALTER PIPE MANAGE_DB.pipes.employee_pipe refresh
110
111 // Listar ficheros en el stage
112 LIST @MANAGE_DB.external_stages.csv_folder
113

```

↳ Results ~ Chart

SYSTEM\$PIPE\_STATUS('MANAGE\_DB.PIPES.EMPLOYEE\_PIPE')

1 {"executionState": "PAUSED", "pendingFileCount": 0, "lastIngestedTimestamp": "2023-01-11T10:30:00.000Z"}

```

// Listar ficheros en el stage
LIST @MANAGE_DB.external_stages.csv_folder

SELECT * FROM PRIMERABBDD.PRIMERESQUEMA.employees2

// Recargar manualmente los ficheros que estaban en el bucket previamente
COPY INTO PRIMERABBDD.PRIMERESQUEMA.employees2
FROM @MANAGE_DB.external_stages.csv_folder

// Reactivar pipe
ALTER PIPE MANAGE_DB.pipes.employee_pipe SET PIPE_EXECUTION_PAUSED = false

// Verificar que el pipe está activo de nuevo
SELECT SYSTEM$PIPE_STATUS('MANAGE_DB.pipes.employee_pipe')

```

### We reactive the pipe:

```

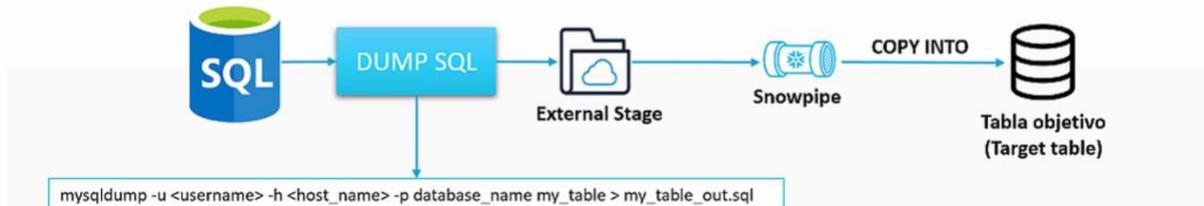
9
0
1 // Reactivar pipe
2 ALTER PIPE MANAGE_DB.pipes.employee_pipe SET PIPE_EXECUTION_PAUSED = false
3
4 // Verificar que el pipe está activo de nuevo
5 SELECT SYSTEM$PIPE_STATUS('MANAGE_DB.pipes.employee_pipe')
6

```

## Migration and load from SQL:

# Migración y Carga desde una BBDD SQL

### Método 1 (Carga de dump CSV desde Cloud)



### Método 2 (Drivers JDBC/ODBC o complementos a instalar en servidor SQL)

- **SQLpipe:** <https://www.sqlpipe.com/blog/how-to-load-data-from-mysql-to-snowflake>
- **HevoData:** <https://hevodata.com/blog/mysql-to-snowflake-data-migration-steps/>



## Different Snowflake table in Snowflake



To use the time travel function, we execute the 3 first commands:

```
MANAGE_DB.PIPES ▾ Settings ▾

1 | SELECT * FROM PRIMERABBDD.PRIMERESQUEMA.employees
2
3 // Insertamos un error
4 UPDATE PRIMERABBDD.PRIMERESQUEMA.employees
5 SET FIRST_NAME = 'Juan'
6
7 // // // Time travel: Método 1 (volver 2 minutos atrás)
8
9 SELECT * FROM PRIMERABBDD.PRIMERESQUEMA.employees at (OFFSET => -60*2)
10
11 // Restaurar tabla original a través de una tabla backup para no eliminar los metadatos de la tabla original
12 CREATE OR REPLACE TABLE PRIMERABBDD.PRIMERESQUEMA.employees_backup as
13 SELECT * FROM PRIMERABBDD.PRIMERESQUEMA.employees at (OFFSET => -60*2)
14
15 // Limpiamos la tabla original
16 TRUNCATE PRIMERABBDD.PRIMERESQUEMA.employees
17
18 // Insertamos los datos de la tabla backup en la tabla original
19 INSERT INTO PRIMERABBDD.PRIMERESQUEMA.employees
20 SELECT * FROM PRIMERABBDD.PRIMERESQUEMA.employees_backup
21
22
23 SELECT * FROM PRIMERABBDD.PRIMERESQUEMA.employees
24
25 DROP PRIMERABBDD.PRIMERESQUEMA.employees_backup
26
27 // Método 2 time stamp: SELECT * FROM PRIMERABBDD.PRIMERESQUEMA.employees before (timestamp => '2023-08-16 07:30:47.145'::timestamp)
28 // Método 3 query ID: SELECT * FROM PRIMERABBDD.PRIMERESQUEMA.employees before (statement => '019b9ee5-0500-8473-0043-4d8300073062')
29
30
31
```

with this command we go back 2 min before:

```
// // // Time travel: Método 1 (volver 2 minutos atrás)
```

```
| SELECT * FROM PRIMERABBDD.PRIMERESQUEMA.employees at (OFFSET => -60*2)
```

We create a backup table:

```
// Restaurar tabla original a través de una tabla backup para no eliminar los metadatos de la tabla original
| CREATE OR REPLACE TABLE PRIMERABBDD.PRIMERESQUEMA.employees_backup as
SELECT * FROM PRIMERABBDD.PRIMERESQUEMA.employees at (OFFSET => -60*2)
```

We insert data to original table:

```
// Insertamos los datos de la tabla backup en la tabla original
INSERT INTO PRIMERABBDD.PRIMERESQUEMA.employees
SELECT * FROM PRIMERABBDD.PRIMERESQUEMA.employees_backup

| SELECT * FROM PRIMERABBDD.PRIMERESQUEMA.employees
DROP PRIMERABBDD.PRIMERESQUEMA.employees_backup

// Método 2 time stamp: SELECT * FROM PRIMERABBDD.PRIMERESQUEMA.employees before (timestamp => '2023-08-16 07:30:47.145'::timestamp)
// Método 3 query ID: SELECT * FROM PRIMERABBDD.PRIMERESQUEMA.employees before (statement => '019b9ee5-0500-8473-0043-4d8300073062')
```

## Time Travel y Fail Safe

### Ciclo de vida de la protección de datos

- No hay operaciones de usuario
- Recuperación después de Time Travel
- Solo mediante soporte Snowflake

SELECT ... AT | BEFORE

Acceso y consulta de datos

**Fail Safe**(Tablas Transient: 0 días  
Tablas permanentes: 7 días)**Time Travel**

(1 – 90 días)

Almacenamiento actual

### Types of table

## Tipos de tabla en Snowflake

**PERMANENTES****CREATE TABLE**

- Utilizada para datos permanentes.
- Permanece hasta DROP
- Time Travel 0-90 días
- Fail Safe

**TRANSIENT****CREATE TRANSIENT TABLE**

- Datos que no necesitan ser protegidos.
- Permanece hasta DROP
- Time Travel 0-1 día
- No hay Fail Safe

**TEMPORARY****CREATE TEMPORARY TABLE**

- Datos no permanentes
- Permanecen durante la sesión
- Time Travel 0-1 día
- No hay Fail Safe

Share our data with another Snowflake account:

We will share the table Movie\_Titles:

MANAGE\_DB.PIPES Settings

```
1 -----EN LA CUENTA DEL PRODUCTOR DE DATOS-----
2
3 // Crear objeto share
4 CREATE OR REPLACE SHARE ORDERS_SHARE;
5
```

↳ Results  Chart

	status
1	Share ORDERS_SHARE successfully created.

### Give permissions:

```
// Permiso a BBDD
GRANT USAGE ON DATABASE PRIMERABBDD TO SHARE ORDERS_SHARE;

// Permiso a esquema
GRANT USAGE ON SCHEMA PRIMERABBDD.PRIMERESQUEMA TO SHARE ORDERS_SHARE;

// Permiso a tabla
GRANT SELECT ON TABLE PRIMERABBDD.PRIMERESQUEMA.ORDERS TO SHARE ORDERS_SHARE;
```

### We verify permissions and add a consumer account:

```
// Verificar permisos
SHOW GRANTS TO SHARE ORDERS_SHARE;

---- Añadir la cuenta del consumidor ---
ALTER SHARE ORDERS_SHARE ADD ACCOUNT <cuenta_consumidor>;
```

```

MANAGE_DB.PRIMERESQUEMA *      Settings *

1 // Ver las particiones conmigo
2 SHOW SHARES
3
4 // Detalle del share
5 DESC SHARE ID.ORDERS.SHARE
6
7 // Crear BBDD en la cuenta consumidor usando el share
8 CREATE DATABASE PRIMERABBDD FROM SHARE ID.ORDERS.SHARE
9
10 // Validar resultados
11 SELECT * FROM PRIMERABBDD.PRIMERESQUEMA.MOVIE_T.TITLES

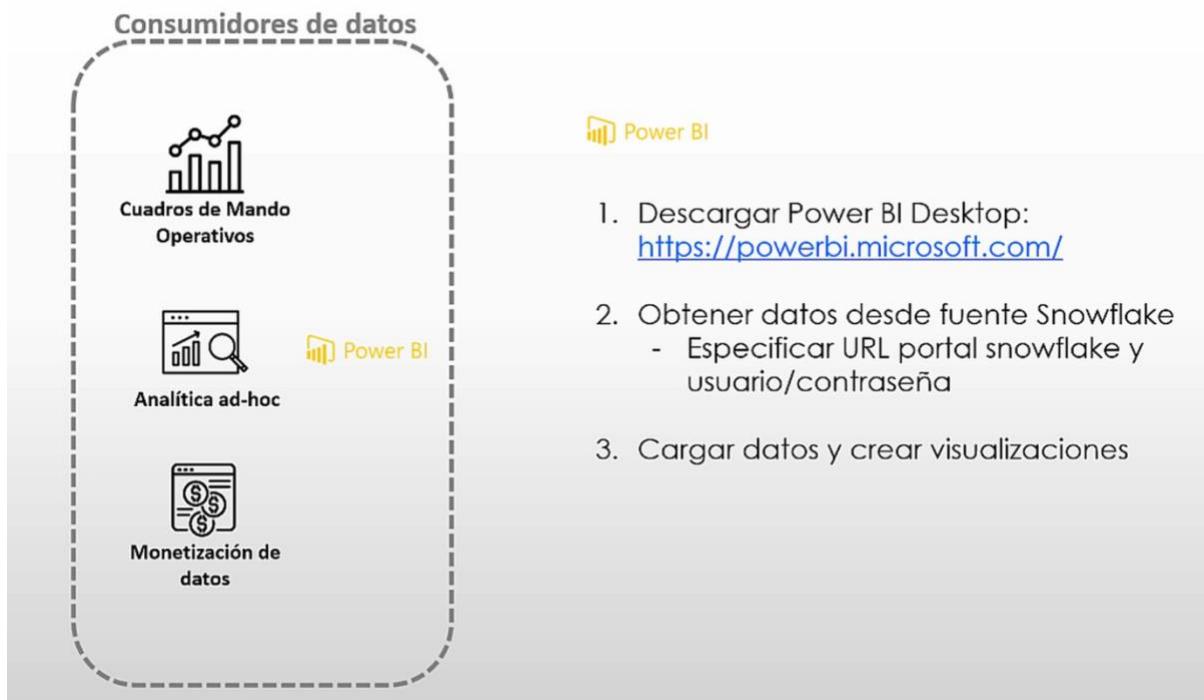
```

We could also create a Direct share:

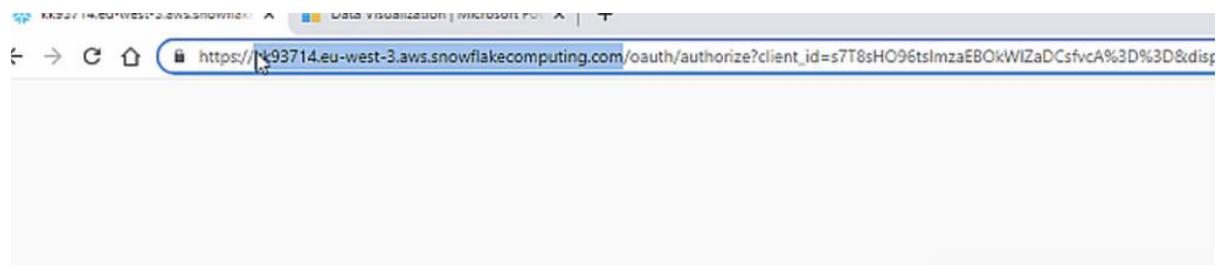
The screenshot shows the Snowflake interface with a modal dialog titled "Share Data". The dialog lists tables under the schema "BBDD\_VENTAS", specifically "TABLA\_VENTAS". The "DATA" section provides details about the table: Created 2 weeks ago, Rows 1,314, Size 0.0MB, and Owner ACCOUNTADMIN. A "Done" button is visible at the bottom right of the dialog.

Connect Snowflake to Power BI:

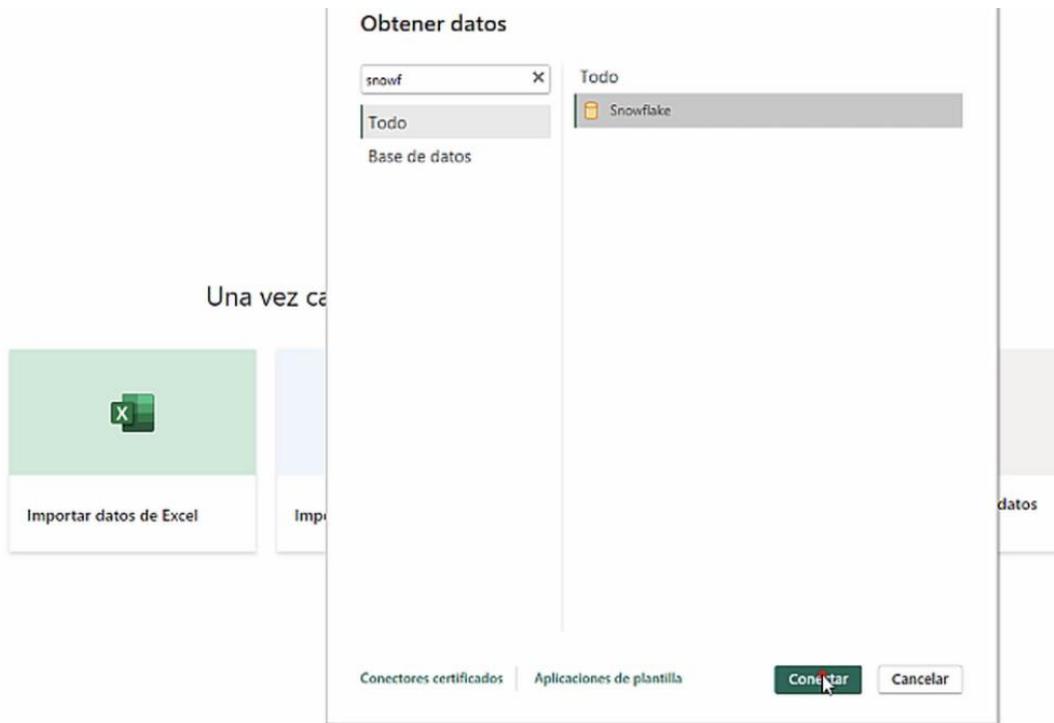
# ¿Cómo conectar Snowflake y Power BI?



We copy the url from Snowflake:



We click connect data in Power BI and snowflake:



We go to Snowflake, Admin, Warehouses:

The screenshot shows the Snowflake Admin interface. On the left, the navigation sidebar includes 'Worksheets', 'Dashboards', 'Data', 'Marketplace', 'Activity', 'Admin' (selected), 'Usage', and 'Warehouses' (selected). The main area is titled 'Warehouses' and displays a table with two rows:

NAME	SIZE	STATUS	CLUSTERS	RUNNING	QUEUED
COMPUTE_WH	XS	Suspended	1	0	0
WAREHOUSE_CURSO	XS	Suspended	0	0	0

Copy the Compute\_WH in Power BI:



Obtener datos de otro origen →

Importar datos de Excel

Navegador

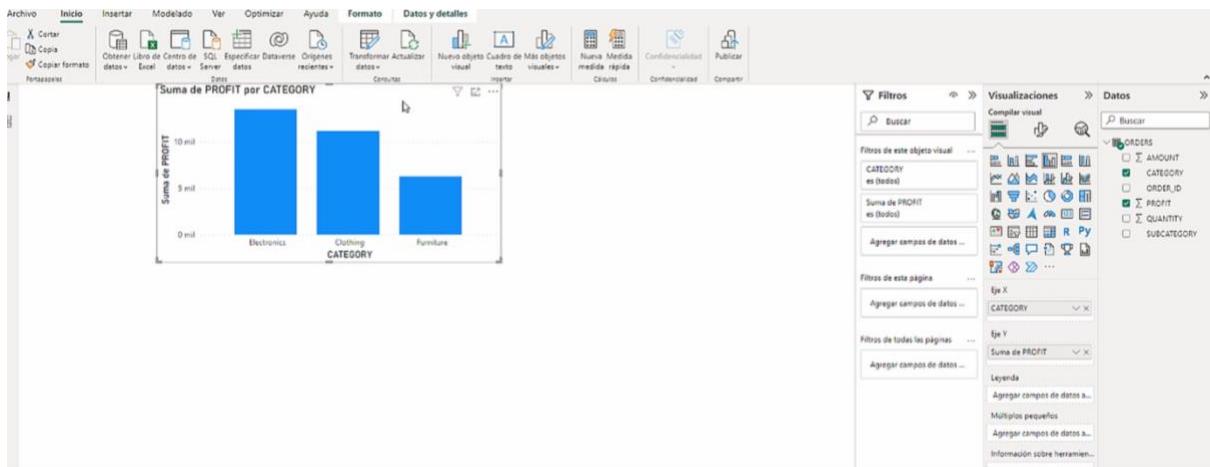
ORDERS

ORDER_ID	AMOUNT	PROFIT	QUANTITY	CATEGORY
B-25601	1275	-1148	7	Furniture
B-25601	66	-12	5	Clothing
B-25601	8	-2	3	Clothing
B-25601	80	-56	4	Electronics
B-25602	168	-111	2	Electronics
B-25602	424	-272	5	Electronics
B-25602	2617	1151	4	Electronics
B-25602	561	212	3	Clothing
B-25602	119	-5	8	Clothing
B-25603	1355	-60	5	Clothing
B-25603	24	-30	1	Furniture
B-25603	193	-186	3	Clothing
B-25603	180	5	3	Clothing
B-25603	116	16	4	Clothing
B-25603	107	36	6	Clothing
B-25603	12	1	2	Clothing
B-25603	38	18	1	Clothing
B-25604	65	17	2	Clothing
B-25604	157	5	9	Clothing
B-25605	75	0	7	Clothing
B-25606	87	4	2	Clothing
B-25607	50	15	4	Clothing
B-25608	1364	864	5	Furniture

Seleccionar tablas relacionadas Crear Transformar datos Cancelar



Obtener datos de otro origen →



How to connect snowflake with Python:



## ¿Cómo conectar Snowflake y Python?

python

- Instalar librería conector para Python y SQL (en la terminal CMD)

```
pip install snowflake-connector-Python
pip install sqlalchemy
```

- Importar librerías en script Python

```
import snowflake.connector
from snowflake.sqlalchemy import URL
from sqlalchemy import create_engine

engine = create_engine(URL(
    user='',
    password='',
    account_identifier='',
    database = 'PRIMERABDD',
    schema = 'PRIMERESQUEMA',
    warehouse = 'COMPUTE_WH',
    role='ACCOUNTADMIN',
))
```

- Definir el motor de conexión a Snowflake

```
#Iniciar la conexión
connection = engine.connect()

#Qué hacer si la tabla ya existe? replace, append, o fail?
#df = ...
#with engine.connect() as con:
#    df.to_sql(name=<nombre_tabla_snowflake>.lower(), con=con, if_exists=if_exists, method=pd_writer)

df = pd.read_sql_query("SELECT * FROM <nombre_tabla_snowflake>", engine)
```

- Leer de una tabla de Snowflake

```
#Iniciar la conexión
connection = engine.connect()

#Qué hacer si la tabla ya existe? replace, append, o fail?
#df = ...
#with engine.connect() as con:
#    df.to_sql(name=<nombre_tabla_snowflake>.lower(), con=con, if_exists=if_exists, method=pd_writer)

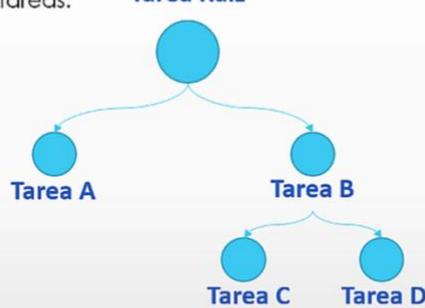
df = pd.read_sql_query("SELECT * FROM <nombre_tabla_snowflake>", engine)
```

- Escribir en una tabla de Snowflake

## Create tasks and streams:

## ¿Qué son las tareas en Snowflake?

- Tareas:** Programar las ejecuciones SQL de Snowflake con la periodicidad que precisemos.
- Se pueden crear árboles de tareas:



Todas las tareas tienen un "parent" salvo la raíz

- Creación de una tarea:

```
CREATE TASK ...
  AFTER <parent task>
  AS ...
```

## we create task\_db database:

TASK\_DB.PUBLIC \* Settings \*

```

1 // 7.2 Creación de tareas
2
3 // Crear BBDD transient
4 CREATE OR REPLACE TRANSIENT DATABASE TASK_DB;
5 USE TASK_DB
6
7

```

**↳ Results** ↵ Chart

status
1 Database TASK_DB successfully created.

We create the table and the task:

```

// Preparar tabla
CREATE OR REPLACE TABLE CUSTOMERS (
    CUSTOMER_ID INT AUTOINCREMENT START = 1 INCREMENT =1,
    FIRST_NAME VARCHAR(40) DEFAULT 'Iván' ,
    CREATE_DATE DATE)

// Create Tarea
CREATE OR REPLACE TASK CUSTOMER_INSERT
    WAREHOUSE = COMPUTE_WH
    SCHEDULE = '1 MINUTE' // Programar siempre en minutos
    AS
        INSERT INTO CUSTOMERS(CREATE_DATE) VALUES(CURRENT_TIMESTAMP); //Podría ser un COPY INTO

```

We start the task:

```

// Mostrar tareas
SHOW TASKS;

```

	created_on	name	id	database_name	schema_name
1	-09 00:24:29.380 -0700	CUSTOMER_INSERT	01acd8dc-e821-fb0c-0000-000000000003	TASK_DB	PUBLIC

Results Chart

nt	warehouse	schedule	predecessors	state	definition
1	COMPUTE_WH	1 MINUTE	[]	started	INSERT INTO CUSTOMERS(CREATE_DATE) VALUES(CURRENT_TIMESTAMP)

We suspend the task:

```
// Mostrar tareas
SHOW TASKS;

// Empezar la tarea y suspenderla
ALTER TASK CUSTOMER_INSERT RESUME;
ALTER TASK CUSTOMER_INSERT SUSPEND;
```

Use of CRON for flexible frequency:

```
34 // 7.3 Uso de CRON para crear tareas con periodicidad flexible
35
36 CREATE OR REPLACE TASK CUSTOMER_INSERT
37   WAREHOUSE = COMPUTE_WH
38   SCHEDULE = 'USING CRON * * * * UTC'
39   AS
40     INSERT INTO CUSTOMERS(CREATE_DATE) VALUES(CURRENT_TIMESTAMP);
41
42   # ----- minute (0-59)
43   # | ----- hour (0-23)
44   # | | ----- day of month (1-31, or L)
45   # | | | ----- month (1-12, JAN-DEC)
46   # | | | | --- day of week (0-6, SUN-SAT, or L)
47   # | | | |
48   # | | | |
49   # | | | |
50   # * * * * *
51
52
53 // Ejemplo Cada minuto
54 SCHEDULE = 'USING CRON * * * * UTC'
55
56 // Ejemplo cada día a las 6:00 UTC
57 SCHEDULE = 'USING CRON 0 6 * * * UTC'
58
59 // Ejemplo cada día a las 9:00 y a las 17:00 UTC
60 SCHEDULE = 'USING CRON 0 9,17 * * * UTC'
61
62 // Ejemplo cada hora desde las 9:00 hasta las 17:00 los domingos utilizando la zona horaria América/Los Ángeles
63 SCHEDULE = 'USING CRON 0 9-17 * * SUN America/Los_Angeles'
```

Tree tasks:

```
1 | SELECT * FROM CUSTOMERS;
2 |
3 | // Preparar 2a tabla
4 | CREATE OR REPLACE TABLE CUSTOMERS2 (
5 |     CUSTOMER_ID INT,
6 |     FIRST_NAME VARCHAR(40),
7 |     CREATE_DATE DATE)
8 |
9 |
10 | // Suspender tarea "parent"
11 | ALTER TASK CUSTOMER_INSERT SUSPEND;
12 |
13 | // Crear tarea "child" CUSTOMER_INSERT2
14 | CREATE OR REPLACE TASK CUSTOMER_INSERT2
15 |     WAREHOUSE = COMPUTE_WH
16 |     AFTER CUSTOMER_INSERT
17 |     AS
18 |         INSERT INTO CUSTOMERS2 SELECT * FROM CUSTOMERS;
19 |
20 |
21 | // Preparar 3a tabla
22 | CREATE OR REPLACE TABLE CUSTOMERS3 (
23 |     CUSTOMER_ID INT,
24 |     FIRST_NAME VARCHAR(40),
25 |     CREATE_DATE DATE,
26 |     INSERT_DATE DATE DEFAULT DATE(CURRENT_TIMESTAMP))
```

```
// Crear tarea "child" CUSTOMER_INSERT3
CREATE OR REPLACE TASK CUSTOMER_INSERT3
    WAREHOUSE = COMPUTE_WH
    AFTER CUSTOMER_INSERT2
    AS
        INSERT INTO CUSTOMERS3 (CUSTOMER_ID,FIRST_NAME,CREATE_DATE) SELECT * FROM CUSTOMERS2; 
SHOW TASKS;

ALTER TASK CUSTOMER_INSERT
SET SCHEDULE = '1 MINUTE'

// Reactivar tareas (primero la tarea raiz)
ALTER TASK CUSTOMER_INSERT RESUME;
ALTER TASK CUSTOMER_INSERT2 RESUME;
ALTER TASK CUSTOMER_INSERT3 RESUME;

SELECT * FROM CUSTOMERS2
```

```

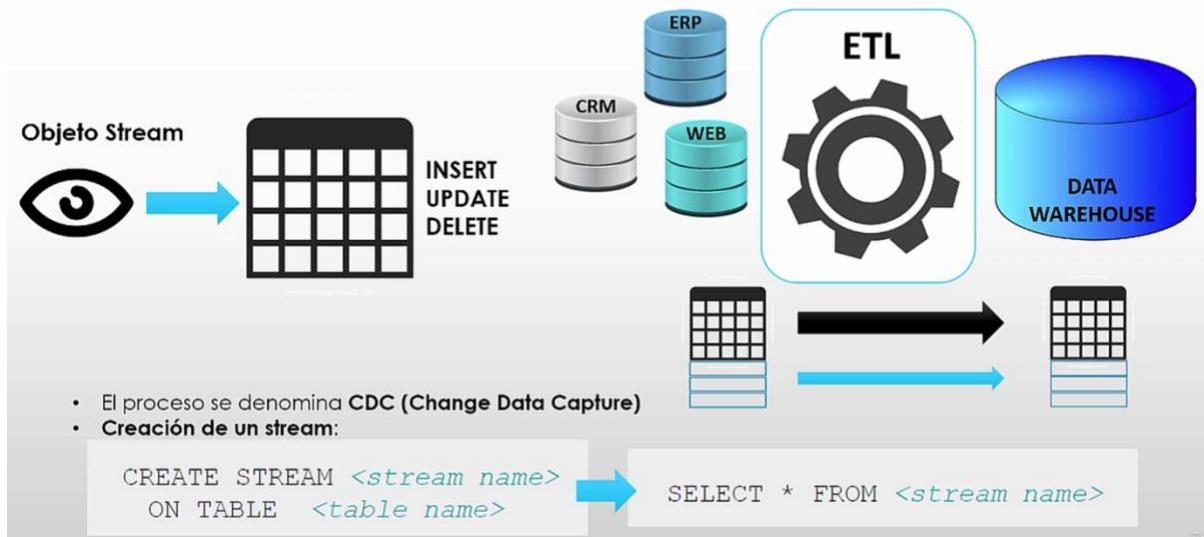
l14
l15
l16 SHOW TASKS;
l17
l18 ALTER TASK CUSTOMER_INSERT
l19 SET SCHEDULE = '1 MINUTE'
l20
l21 ALTER TASK CUSTOMER_INSERT SUSPEND;
l22 // Reactivar tareas (primero la tarea child)
l23 ALTER TASK CUSTOMER_INSERT RESUME;
l24 ALTER TASK CUSTOMER_INSERT2 RESUME;
l25 ALTER TASK CUSTOMER_INSERT3 RESUME;
l26
l27
l28 SELECT * FROM CUSTOMERS2
l29
l30 SELECT * FROM CUSTOMERS3
l31
l32 // Suspender tareas
l33 ALTER TASK CUSTOMER_INSERT SUSPEND;
l34 ALTER TASK CUSTOMER_INSERT2 SUSPEND;
l35 ALTER TASK CUSTOMER_INSERT3 SUSPEND;
l36
l37
l38 // Verificar histórico de tareas
l39 SHOW TASKS;
l40
l41 // Usar función "TASK_HISTORY()"
l42 select *
l43   from table(information_schema.task_history())
l44   order by scheduled_time desc;

```



## ¿Qué son los Streams en Snowflake?

- **Streams:** Objetos que observan si hay cambios en una tabla para producir una acción.



### Stream creation and INSERT:

STREAMS\_DB.PUBLIC \* Settings \*

```
1 ----- Ejemplo Stream: INSERT -----
2 CREATE OR REPLACE TRANSIENT DATABASE STREAMS_DB;
3
4 -- Crear tabla ejemplo "ventas"
5 create or replace table sales_raw_staging(
6   id varchar,
7   product varchar,
8   price varchar,
9   amount varchar,
10  store_id varchar);
11
12 -- Insertar valores
13 INSERT INTO sales_raw_staging
14   values
15     (1,'Banana',1.99,1,1),
16     (2,'Lemon',0.99,1,1),
17     (3,'Apple',1.79,1,2),
18     (4,'Orange Juice',1.89,1,2),
19     (5,'Cereals',5.98,2,1);
20
21 -- Crear tabla ejemplo "tiendas"
22 create or replace table store_table(
23   store_id number,
24   location varchar,
25   employees number);
26
27 -- Insertar valores
28 INSERT INTO STORE_TABLE VALUES(1,'Chicago',33);
29 INSERT INTO STORE_TABLE VALUES(2,'London',12);
--
```

```
-- Crear tabla final global
create or replace table sales_final_table(
  id int,
  product varchar,
  price number,
  amount int,
  store_id int,
  location varchar,
  employees int);

-- Insertar tabla final
INSERT INTO sales_final_table
  SELECT
    SA.id,
    SA.product,
    SA.price,
    SA.amount,
    ST.STORE_ID,
    ST.LOCATION,
    ST.EMPLOYEES
  FROM SALES_RAW_STAGING SA
  JOIN STORE_TABLE ST ON ST.STORE_ID=SA.STORE_ID ;
```

We create the Stream object:

```

-- Crear objeto stream sobre la tabla bruta de "ventas"
create or replace stream sales_stream on table sales_raw_staging;

SHOW STREAMS;

DESC STREAM sales_stream;

-- Verificar estado tablas y stream
select * from sales_stream;

select * from sales_raw_staging;

-- Insertar valores en tabla bruta de "ventas"
insert into sales_raw_staging
values
(6,'Mango',1.99,1,2),
(7,'Garlic',0.99,1,1);

-- Obtener cambios en los datos usando el Stream (INSERTS)
select * from sales_stream;

select * from sales_raw_staging;

```

-- Crear objeto stream sobre la tabla bruta de "ventas"

The screenshot shows a table with the following data:

ID	...	PRODUCT	PRICE	AMOUNT	STORE_ID	METADATAACTION	METADATASISUPDATE
1	6	Mango	1.99	1	2	INSERT	FALSE
2	7	Garlic	0.99	1	1	INSERT	FALSE

### original table:

The screenshot shows a table with the following data:

ID	...	PRODUCT	PRICE	AMOUNT	STORE_ID	LOCATION	EMPLOYEES
1	1	Banana	2	1	1	Chicago	33
2	2	Lemon	1	1	1	Chicago	33
3	3	Apple	2	1	2	London	12
4	4	Orange Juice	2	1	2	London	12
5	5	Cereals	6	2	1	Chicago	33

```

| select * from sales_stream;
| select * from sales_raw_staging;
| select * from sales_final_table;

-- Consumir objeto stream (quedará vacío posteriormente)
INSERT INTO sales_final_table
    SELECT
        SA.id,
        SA.product,
        SA.price,
        SA.amount,
        ST.STORE_ID,
        ST.LOCATION,
        ST.EMPLOYEES
    FROM SALES_STREAM SA
    JOIN STORE_TABLE ST ON ST.STORE_ID=SA.STORE_ID ;

```

```

-- Verificar estado tablas y stream (tras ser consumido)
select * from sales_stream;
SELECT * FROM SALES_FINAL_TABLE;

```

```
| SELECT * FROM SALES_RAW_STAGING;
```

### final table:

84 -- Consumir objeto stream (quedará vacío posteriormente)

	...	ID	PRODUCT	PRICE	AMOUNT	STORE_ID	LOCATION	EMPLOYEES
1		1	Banana	2	1	1	Chicago	33
2		2	Lemon	1	1	1	Chicago	33
3		3	Apple	2	1	2	London	12
4		4	Orange Juice	2	1	2	London	12
5		5	Cereals	6	2	1	Chicago	33
6		7	Garlic	1	1	1	Chicago	33
7		6	Mango	2	1	2	London	12

### Update:

```
-- 7.7 Uso de Stream para operación UPDATE

SELECT * FROM SALES_RAW_STAGING;

SELECT * FROM SALES_STREAM;

UPDATE SALES_RAW_STAGING
SET PRODUCT ='Potato' WHERE PRODUCT = 'Banana'

// Combinar los resultados para actualizar en la tabla final
merge into SALES_FINAL_TABLE F      -- Tabla objetivo donde consolidar los cambios de la tabla bruta
using SALES_STREAM S              -- Stream que ha capturado los cambios
  on f.id = s.id
when matched
  and S.METADATA$ACTION = 'INSERT'
  and S.METADATA$ISUPDATE = 'TRUE'      -- Indica que el registro ha sido actualizado
then update
  set f.product = s.product,
      f.price = s.price,
      f.amount= s.amount,
      f.store_id=s.store_id;

SELECT * FROM SALES_FINAL_TABLE

SELECT * FROM SALES_RAW_STAGING;

SELECT * FROM SALES_STREAM;
```

↳ Results ↵ Chart

	ID	...	PRODUCT	PRICE	AMOUNT	STORE_ID
1	1		Banana	1.99	1	1
2	2	↳	Lemon	0.99	1	1
3	3		Apple	1.79	1	2
4	4		Orange Juice	1.89	1	2
5	5		Cereals	5.98	2	1
6	6		Mango	1.99	1	2
7	7		Garlic	0.99	1	1

We substitute banana for potato:

↳ Results ↵ Chart

	ID	PRODUCT	...	PRICE	AMOUNT	STORE_ID
1	1	Potato	↳	1.99	1	1
2	2	Lemon		0.99	1	1
3	3	Apple		1.79	1	2
4	4	Orange Juice		1.89	1	2
5	5	Cereals		5.98	2	1
6	6	Mango		1.99	1	2
7	7	Garlic		0.99	1	1

## DELETE:

```
-- 7.8 Uso de Stream para operación DELETE

SELECT * FROM SALES_FINAL_TABLE
SELECT * FROM SALES_RAW_STAGING;
SELECT * FROM SALES_STREAM;

DELETE FROM SALES_RAW_STAGING
WHERE PRODUCT = 'Lemon';

-- ***** Procesar stream y consolidar resultados tras eliminar *****
merge into SALES_FINAL_TABLE F      -- Tabla objetivo donde consolidar los cambios de la tabla bruta
using SALES_STREAM S                -- Stream que ha capturado los cambios
on f.id = s.id
when matched
    and S.METADATA$ACTION = 'DELETE'
    and S.METADATA$ISUPDATE = 'FALSE'
    then delete

// 7.9 Procesamiento de todos los cambios de datos en Streams

merge into SALES_FINAL_TABLE F      -- Tabla objetivo donde consolidar los cambios de la tabla bruta
USING ( SELECT STRE.* ,ST.location,ST.employees
        ,STRE
        ,STRE )
```

↳ Results ↵ Chart

	ID	PRODUCT	PRICE	...	AMOUNT	STORE_ID	METADATASACTION	METADATASISUPDATE
1	2	Lemon	0.99		1	1	DELETE	FALSE

we delete row lemon from final table:

```
145
146   | SELECT * FROM SALES_FINAL_TABLE
147
```

↳ Results ↵ Chart

	...	ID	PRODUCT	PRICE	AMOUNT	STORE_ID	LOCATION	EMPLOYEES
1		1	Potato	2	1	1	Chicago	33
2		3	Apple	2	1	2	London	12
3		4	Orange Juice	2	1	2	London	12
4		5	Cereals	6	2	1	Chicago	33
5		7	Garlic	1	1	1	Chicago	33
6		6	Mango	2	1	2	London	12

Processing all the data changes in Streams:

```

167
168 // 7.9 Procesamiento de todos los cambios de datos en Streams
169
170 merge into SALES_FINAL_TABLE F      -- Tabla objetivo donde consolidar los cambios de la tabla bruta
171 USING ( SELECT STRE.* ,ST.location,ST.employees
172     FROM SALES_STREAM STRE
173     JOIN STORE_TABLE ST
174     ON STRE.store_id = ST.store_id
175     ) S
176 ON F.id=S.id
177 when matched          -- Condición DELETE
178     and S.METADATA$ACTION = 'DELETE'
179     and S.METADATA$ISUPDATE = 'FALSE'
180     then delete
181 when matched          -- Condición UPDATE
182     and S.METADATA$ACTION = 'INSERT'
183     and S.METADATA$ISUPDATE = 'TRUE'
184     then update
185     set f.product = s.product,
186         f.price = s.price,
187         f.amount= s.amount,
188         f.store_id=s.store_id
189 when not matched
190     and S.METADATA$ACTION = 'INSERT'
191     then insert
192     (id.product,price,store_id,amount,employees,location)
193     values
194     (s.id, s.product,s.price,s.store_id,s.amount,s.employees,s.location)
195
196
197
198 -- Verificar estado tablas y stream (tras ser consumido)

-- Verificar estado tablas y stream (tras ser consumido)

SELECT * FROM SALES_RAW_STAGING;

| SELECT * FROM SALES_STREAM;

SELECT * FROM SALES_FINAL_TABLE;

// Aplicar cambios INSERT
| INSERT INTO SALES_RAW_STAGING VALUES (2,'Lemon',0.99,1,1);

-- Verificar estado tablas y stream (tras ser consumido)

SELECT * FROM SALES_RAW_STAGING;

SELECT * FROM SALES_STREAM;

SELECT * FROM SALES_FINAL_TABLE;

```

```

207
208 // Aplicar cambios INSERT
209
210 INSERT INTO SALES_RAW_STAGING VALUES (2,'Lemon',0.99,1,1);
211
212 -- Verificar estado tablas y stream (tras ser consumido)
213

```

↳ Results ↵ Chart

	ID	PRODUCT	PRICE	...	AMOUNT	STORE_ID	LOCATION	EMPLOYEES
1	2	Lemon	1		1	1	Chicago	33
2	1	Potato	2		1	1	Chicago	33
3	3	Apple	2		1	2	London	12
4	4	Orange Juice	2		1	2	London	12
c	c	Cereals	c		c	c	Chicago	c

# ID  
2

// Aplicar cambios UPDATE

```

UPDATE SALES_RAW_STAGING
SET PRODUCT = 'Lemonade'
WHERE PRODUCT = 'Lemon'

```

-- Verificar estado tablas y stream (tras ser consumido)

SELECT \* FROM SALES\_RAW\_STAGING;

SELECT \* FROM SALES\_STREAM;

SELECT \* FROM SALES\_FINAL\_TABLE;

// Aplicar cambios DELETE

```

DELETE FROM SALES_RAW_STAGING

```

DELETE FROM SALES\_STREAM

DELETE FROM SALES\_FINAL\_TABLE

↳ Results ↵ Chart

	ID	PRODUCT	...	PRICE	AMOUNT	STORE_ID	LOCATION	EMPLOYEES
1	2	Lemonade	0	1	1	1	Chicago	33
2	1	Potato	2	2	1	1	Chicago	33
3	3	Apple	2	2	1	2	London	12
4	4	Orange Juice	2	2	1	2	London	12
5	5	Cereals	6	6	2	1	Chicago	33
6	7	Garlic	1	1	1	1	Chicago	33
7	6	Mango	2	2	1	2	London	12

```

// Aplicar cambios DELETE

DELETE FROM SALES_RAW_STAGING
WHERE PRODUCT = 'Lemonade';

-- Verificar estado tablas y stream (tras ser consumido)

SELECT * FROM SALES_RAW_STAGING;

SELECT * FROM SALES_STREAM;

SELECT * FROM SALES_FINAL_TABLE;

```

↳ Results    ↵ Chart

	ID	PRODUCT	...	PRICE	AMOUNT	STORE_ID	LOCATION	EMPLOYEES
1	1	Potato		2	1	1	Chicago	33
2	3	Apple		2	1	2	London	12
3	4	Orange Juice		2	1	2	London	12
4	5	Cereals		6	2	1	Chicago	33
5	7	Garlic		1	1	1	Chicago	33
6	6	Mango		2	1	2	London	12

### Combine tasks in Streams:

```

246
247
248 // 7.10 Combinar Tareas y Streams en Snowflake
249
250 // Crear la tarea de verificación de cambios cada minuto
251
252 CREATE OR REPLACE TASK all_data_changes
253   WAREHOUSE = COMPUTE_WH
254   SCHEDULE = '1 MINUTE'
255   WHEN SYSTEM$STREAM_HAS_DATA('SALES_STREAM')
256   AS -- Aplicar la lógica que procesa todos los cambios
257   merge into SALES_FINAL_TABLE F
258   USING ( SELECT STRE.* ,ST.location,ST.employees
259         FROM SALES_STREAM STRE
260         JOIN STORE_TABLE ST
261         ON STRE.store_id = ST.store_id
262         ) S
263   ON F.id=S.id
264   when matched
265     and S.METADATA$ACTION = 'DELETE'
266     and S.METADATA$ISUPDATE = 'FALSE'
267     then delete
268   when matched
269     and S.METADATA$ACTION = 'INSERT'
270     and S.METADATA$ISUPDATE = 'TRUE'
271     then update
272       set f.product = s.product,
273           f.price = s.price,
274           f.amount= s.amount,
275           f.store_id=s.store_id
276   when not matched
277     and S.METADATA$ACTION = 'INSERT'
278     then insert

```

	...	created_on	name	id	database_name	sc
1	2023-06-09 00:56:27.984 -0700		ALL_DATA_CHANGES	01acd8fc-ef23-e32d-0000-0000000000a	STREAMS_DB	PL

```

// Aplicar INSERT

INSERT INTO SALES_RAW_STAGING VALUES (11,'Milk',1.99,1,2);
INSERT INTO SALES_RAW_STAGING VALUES (12,'Chocolate',4.49,1,2);
INSERT INTO SALES_RAW_STAGING VALUES (13,'Cheese',3.89,1,1); I

-- Verificar estado tablas y stream (tras ser consumido)

SELECT * FROM SALES_RAW_STAGING;
SELECT * FROM SALES_STREAM;
SELECT * FROM SALES_FINAL_TABLE;

```

	ID	PRODUCT	PRICE	AMOUNT	STORE_ID	LOCATION	...	EMPLOYEES
1	13	Cheese	4	1	1	Chicago		33
2	11	Milk	2	1	2	London		12
3	12	Chocolate	4	1	2	London		12
4	1	Potato	2	1	1	Chicago		33
5	3	Apple	2	1	2	London		12
6	4	Orange Juice	2	1	2	London		12
7	5	Cereals	6	2	1	Chicago		33
8	7	Garlic	1	1	1	Chicago		33
9	6	Mango	2	1	2	London		12

```

UPDATE SALES_RAW_STAGING
SET PRODUCT = 'Chocolate bar'
WHERE PRODUCT = 'Chocolate';

-- Verificar estado tablas y stream (tras ser consumido)

SELECT * FROM SALES_RAW_STAGING;

SELECT * FROM SALES_STREAM; ○

SELECT * FROM SALES_FINAL_TABLE;

```

	ID	...	PRODUCT	PRICE	AMOUNT	STORE_ID	METADATASACTION	METADATASISUPDATE
1	12		Chocolate bar	4.49	1	2	INSERT	TRUE
2	12		Chocolate	4.49	1	2	DELETE	TRUE

```

-- // Aplicar DELETE
315
316
317    DELETE FROM SALES_RAW_STAGING
318 WHERE PRODUCT = 'Mango';
319
320
321 -- Verificar estado tablas y stream (tras ser consumido)
322
323 SELECT * FROM SALES_RAW_STAGING;
324
325 SELECT * FROM SALES_STREAM;
326
327 SELECT * FROM SALES_FINAL_TABLE;
328

```

	ID	PRODUCT	...	PRICE	AMOUNT	STORE_ID	METADATASACTION	METADATASISUPDATE
1	6	Mango		1.99	1	2	DELETE	FALSE

	...	ID	PRODUCT	PRICE	AMOUNT	STORE_ID	LOCATION	EMPLOYEES
1		1	Potato	2	1	1	Chicago	33
2		3	Apple	2	1	2	London	12
3		4	Orange Juice	2	1	2	London	12
4		5	Cereals	6	2	1	Chicago	33
5		7	Garlic	1	1	1	Chicago	33
6		13	Cheese	4	1	1	Chicago	33
7		11	Milk	2	1	2	London	12
8		12	Chocolate bar	4	1	2	London	12