

Collibra + Snowflake: Better together with data lineage

Sheeri Cabral

Senior Product Manager
Collibra

Paul Gancz

Partner Solutions Architect
Snowflake



Sheeri Cabral
Senior Product Manager
Collibra



Paul Gancz
Partner Solutions Architect
Snowflake

Agenda

- 1 Data intelligence starts with the data
- 2 Business challenges
- 3 Collibra + Snowflake
- 4 Snowflake lineage with Collibra
- 5 Demo
- 6 Q&A

Data intelligence starts with the data

Personas

Business Analyst

Data Steward

Data Engineer

Data Privacy Manager

Data Scientist

System of Engagement

Team productivity



GTM

servicenow

IT & Ops



Collibra

Data

ATLASSIAN

Engineering



HR

Data Tools

Individual productivity

ETL /
ELT

Data
Modeling

Data
Science

Dashboards
& Analytics

Data Infrastructure

Compute & storage

Data
Warehouse

Data
Lake

Database /
OLTP

Streaming

Query
engines

Real time
Analytics

Hyperscalers

Business challenges



Limited visibility

Lack a comprehensive view of data



No data provenance

Unclear where and how data flows within and between systems



Lack of trust

Low confidence that data can be trusted and ready to use

Impacting your business with lineage

Get a **complete, context-rich view** of where data comes from and trust that it is relevant and trustworthy

Trace data flows and show relationships, so users can quickly **understand your data's history**

Predict the impact of **potential changes** with lineage at the table, column, transformation and SQL query level

The screenshot shows a data lineage tool interface for a table named 'CUSTOMER_DISCOUNTS'. The interface displays a dependency graph where data flows from source tables to a target table. The source tables are:

- FOOD_DELIVERY.CUSTOMERS [OPENLINEAGE::database]** (Attributes: NAME)
- FOOD_DELIVERY.DISCOUNTS [OPENLINEAGE::database]** (Attributes: AMOUNT_OFF, STARTS_AT, ENDS_AT)
- FOOD_DELIVERY.V_ORDERS [OPENLINEAGE::database]** (Attributes: MENU_ITEM, TOTAL_ORDERED)

The target table is **FOOD_DELIVERY.CUSTOMER_DISCOUNTS [OPENLINEAGE::database]** (Attributes: NAME, AMOUNT_OFF, STARTS_AT, ENDS_AT). Arrows indicate the flow of data from the source tables to the target table. The 'CUSTOMERS' table provides the 'NAME' attribute, the 'DISCOUNTS' table provides 'AMOUNT_OFF', 'STARTS_AT', and 'ENDS_AT', and the 'V_ORDERS' table provides 'MENU_ITEM' and 'TOTAL_ORDERED'.

Below the graph, the SQL query for the target table is displayed:

```
1 CREATE VIEW CUSTOMER_DISCOUNTS AS
2 SELECT NAME, AMOUNT_OFF, STARTS_AT, ENDS_AT
3 FROM DISCOUNTS INNER JOIN CUSTOMERS ON (DISCOUNTS.CUSTOMERS_ID=CUSTOMERS.ID)
```


Collibra & Snowflake partnership

- **Snowflake Elite Partner**
Since 2020
- **Snowflake Ventures**
Investment
- **Snowflake Partner Network Competency Program**
Healthcare and Financial Services

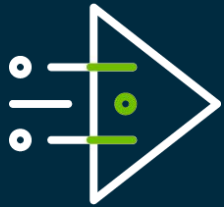


Snowflake platform

Under the hood



Why the Snowflake data cloud?



Fast for any Workload

Run virtually any number or type of job across users and data volumes quickly and reliably.



It just works

Replace manual with automated to operate at scale, optimize costs, and minimize downtime.



Connected to what matters

Extend access and collaboration across teams, workloads, clouds, and data, seamlessly and securely.

Snowflake in action

The screenshot displays the Snowflake Lineage tool interface. At the top, it shows the user 'SYSADMIN' in the 'DEMO_WH' warehouse, with a 'Share' button and a refresh icon. The interface is divided into three main sections: a left sidebar, a central query editor, and a bottom results panel.

Left Sidebar: Contains navigation options for 'Worksheets' and 'Databases'. Under 'Databases', there are sections for 'Pinned (0)', 'No pinned objects', and 'All Objects'. The 'OPENLINEAGE' database is expanded, showing a tree structure with 'FOOD_DELIVERY' (containing Tables, Views, Stages, Pipes, Streams, Tasks, Functions, and Procedures), 'INFORMATION_SCHEMA' (containing Views), and 'PUBLIC' (containing Tables, Views, Stages, Pipes, Streams, Tasks, and Functions).

Central Query Editor: Displays a SQL query for the 'OPENLINEAGE.FOOD_DELIVERY' database. The query is as follows:

```
4 INSERT INTO food_delivery.delivery_7_days (
5     order_id, order_placed_on, order_dispatched_on, order_delivered_on,
6     customer_email, customer_address, discount_id, menu_id,
7     restaurant_id, restaurant_address, menu_item_id, category_id,
8     driver_id
9 )
10 SELECT o.order_id,
11        o.placed_on AS order_placed_on,
12        (SELECT transitioned_at FROM food_delivery.order_status
13         WHERE order_id = o.order_id
14             AND status = 'DISPATCHED') AS order_dispatched_on,
15        (SELECT transitioned_at FROM food_delivery.order_status
16         WHERE order_id = o.order_id
17             AND status = 'DELIVERED') AS order_delivered_on,
18        c.email AS customer_email,
19        c.address AS customer_address,
20        o.discount_id, o.menu_id, o.restaurant_id,
21        r.address AS restaurant_address,
22        o.menu_item_id, o.category_id,
23        d.id AS driver_id
24 FROM food_delivery.orders_7_days AS o
25      INNER JOIN food_delivery.order_status AS os
26            ON os.order_id = o.order_id
27      INNER JOIN food_delivery.customers AS c
28            ON c.id = os.customer_id
29      INNER JOIN food_delivery.restaurants AS r
30            ON r.id = os.restaurant_id
31      INNER JOIN food_delivery.drivers AS d
32            ON d.id = os.driver_id
33 WHERE os.transitioned_at >= TIMEADD(hour, -168, current_time())
34
```

Bottom Results Panel: Features tabs for 'Objects', 'Editor', 'Results', and 'Chart'. The 'Results' tab is active, showing a table with one row and one column:

number of rows inserted
0

Below the table, the 'Query Details' section shows a 'Query duration' of 134ms, accompanied by a progress bar.

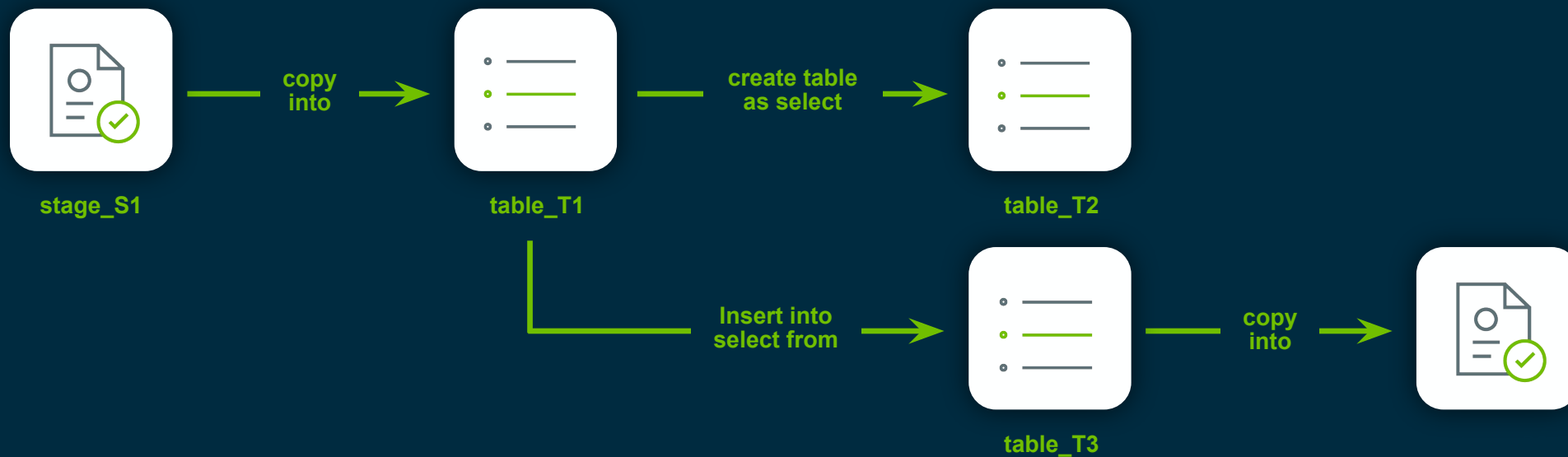
Demo

```
"sources" : [  
  {  
    "id" : "snowOL",  
    "type" : "DatabaseSnowflake",  
    "mode" : "SQL-API",  
    "collibraSystemName" : "Snowflake",  
    "hostname" : "collibra[REDACTED].snowflakecomputing.com",  
    "auth" : {  
      "type" : "Basic",  
      "username" : "CERTIFICATION"  
    },  
    "customConnectionProperties": "role=SYSADMIN",  
    "databaseNames" : [ "OPENLINEAGE", "CERTIFICATION" ],  
    "warehouse": "DEMO_WH"  
  }  
]  
}
```

```
Sheeris-MacBook-Pro ~ ; JAVA_OPTS='--add-opens java.base/java.nio=ALL-UNNAMED' bin/lineage-harvester full-sync
```

Snowflake access history (writes)

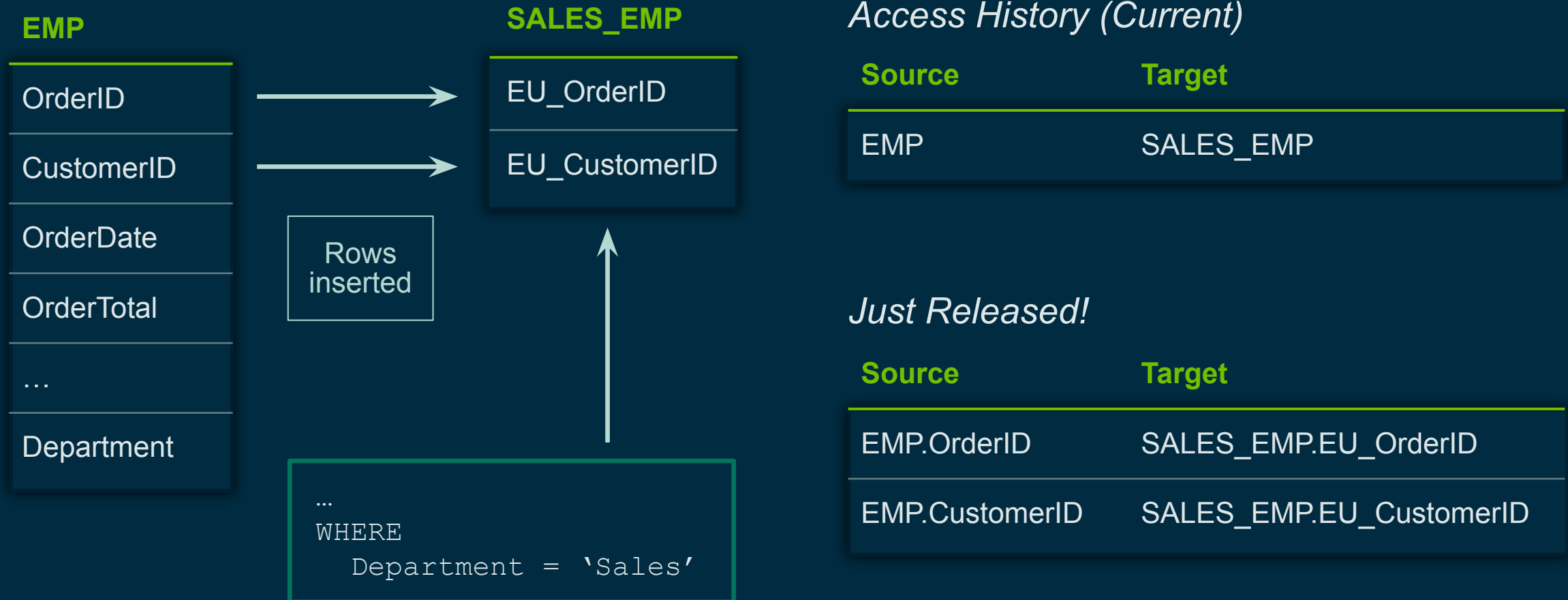
Know the lineage of data



PATH	TARGET_NAME	TARGET_DOMAIN	TARGET_COLUMNS
stage_S1->table_T1	table_T1	TABLE	["CONTENT"]
stage_S1->table_T1->table_T2	table_T2	TABLE	["ID","NAME"]
stage_S1->table_T1->table_T3	table_T3	TABLE	["NAME","ID"]
stage_S1->table_T1->table_T3->stage_S2	stage_S2	STAGE	[]

Column lineage

Track flow of sensitive columns, recorded in Access History



DELIVERY_7_DAYS

Table Candidate | 0 | 0 | 5%

Add to Data Set Actions

Attributes

The diagram illustrates the lineage of data for the **DELIVERY_7_DAYS** table. It shows a flow from various temporary tables (e.g., **FOOD_DELIVERY.TMP_ORDERS**, **FOOD_DELIVERY.TMP_MENU_ITEMS**) through intermediate tables (e.g., **FOOD_DELIVERY.ORDERS**, **FOOD_DELIVERY.MENU_ITEMS**) to the final target table. The final table, **FOOD_DELIVERY.DELIVERY_7_DAYS**, contains columns: ORDER_ID, ORDER_PLACED_ON, ORDER_DISPATCHED_ON, ORDER_DELIVERED_ON, CUSTOMER_EMAIL, CUSTOMER_ADDRESS, DISCOUNT_ID, MENU_ID, RESTAURANT_ID, RESTAURANT_ADDRESS, MENU_ITEM_ID, CATEGORY_ID, and DRIVER_ID.

```
1 INSERT INTO food_delivery.orders_7_days ( order_id, placed_on, discount_id, menu_id, restaurant_id, menu_item_id, category_id)
2 SELECT o.id AS order_id,
3 o.placed_on,
4 o.discount_id,
5 m.id AS menu_id,
6 m.restaurant_id,
7 mi.id AS menu_item_id,
8 c.id AS category_id
9 FROM food_delivery.orders AS o
10 inner join food_delivery.menu_items AS mi
11 ON mi.id = o.menu item id
```

All code

Browse Settings

Search

- All data objects
- DATABASE
 - Snowflake
 - OPENLINEAGE
 - FOOD_DELIVERY
 - BUSINESS_HOURS
 - CATEGORIES
 - CITIES
 - CUSTOMERS
 - CUSTOMER_DISCOUNTS
 - DELIVERY_7_DAYS**
 - DISCOUNTS
 - DRIVERS
 - MENUS
 - MENU_ITEMS
 - ORDERS
 - ORDERS_7_DAYS
 - ORDER_STATUS
 - RESTAURANTS
 - TMP_CATEGORIES
 - TMP_CUSTOMERS
 - TMP_DRIVERS
 - TMP_MENU
 - TMP_MENU_ITEMS
 - TMP_ORDERS
 - TMP_ORDER_STATUS
 - TMP_RESTAURANTS
 - V_ORDERS

Stats

- Tables: 23
- Done: 73
- Parsing errors: 0
- Analyze errors: 0

Questions?

Special thanks:
Raja Balakrishnan
Daphnee Geyer
Mihail Istrate

THE DATA INTELLIGENCE CONFERENCE

DATA
CITIZENS **22**

Next generation of Snowflake lineage

Snowflake SQL-API Now in public beta!

- Dedicated Snowflake scanner for integration
- View technical lineage including column-level lineage and transformations done by:
 - Views
 - Storage procedures
 - Queries/streams
 - Snowpipes

Snowflake SQL - GA

- Leverages SQL scanner for Snowflake integration
- Stored procedures are not parsed
- Scanner is one-directional