Sprint 06 Project task

Understanding SNOWFLAKE

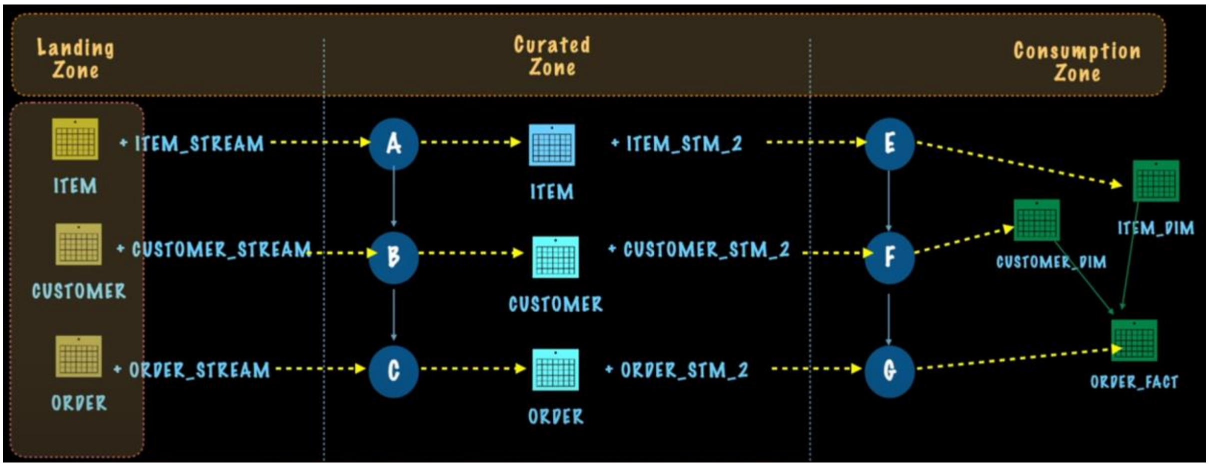
Name – Jatan

Mentor name – Devy

Date – 08/04/2024 to 20/04/2024

OBJECTIVE :- Building ETL pipeline by orchestrating SnowPipe, Stream, and Tasks

Flow of Project :

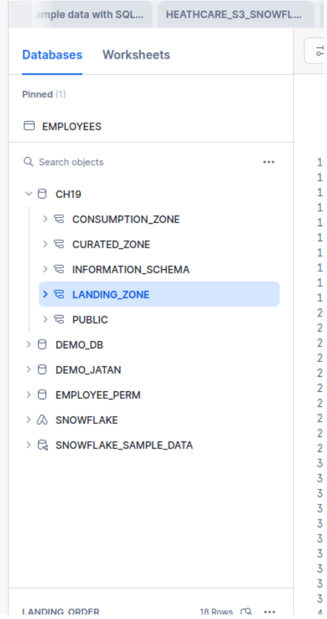


PART 01 Tasks :

1. Create 3 schemas (3 logically layers or zones)
2. Create table (DDL) under landing zone schemas.
3. Initial data load via Web UI.
4. Verify the tables and data.

All the queries are in ETL_IN_SNOWFLAKE_PART01 file

1. Created schemas



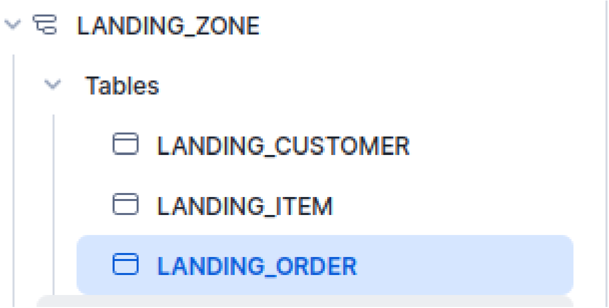
Pinned (1)

EMPLOYEES

- CH19
  - CONSUMPTION_ZONE
  - CURATED_ZONE
  - INFORMATION_SCHEMA
  - LANDING_ZONE
  - PUBLIC
- DEMO_DB
- DEMO_JATAN
- EMPLOYEE_PERM
- SNOWFLAKE
- SNOWFLAKE_SAMPLE_DATA

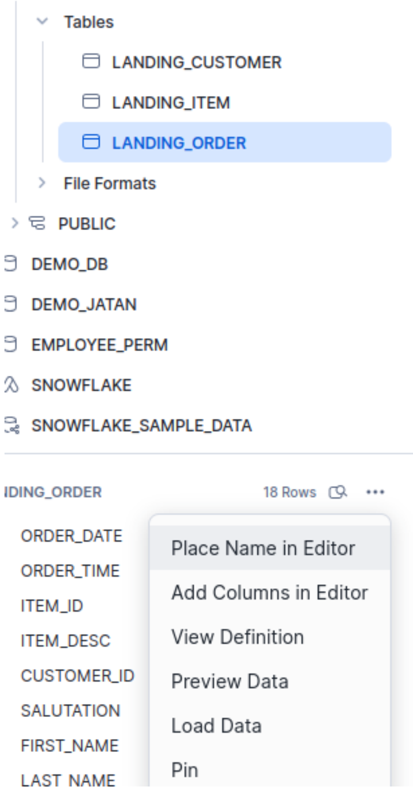
LANDING_ORDER 18 Rows

2. Created table under landing zone Schemas

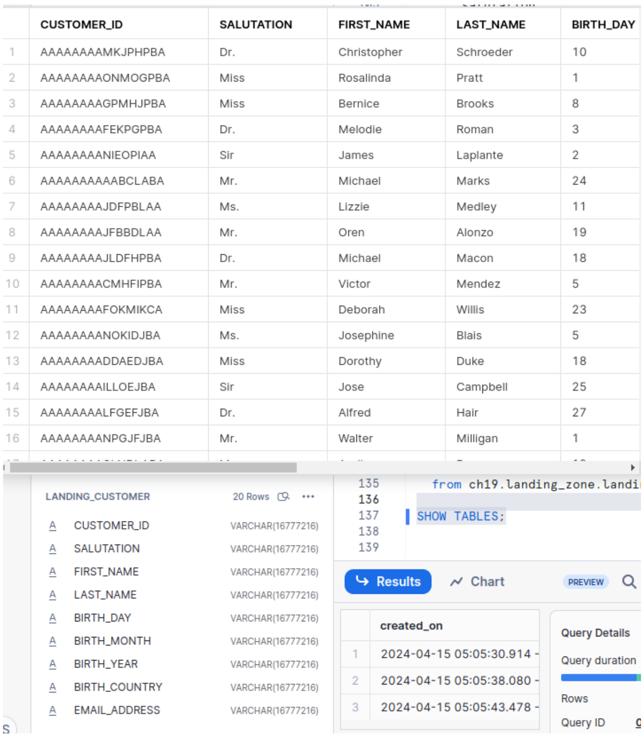


- LANDING_ZONE
  - Tables
    - LANDING_CUSTOMER
    - LANDING_ITEM
    - LANDING_ORDER

3. Initial data load via Web UI.



4. Verify the data



	CUSTOMER_ID	SALUTATION	FIRST_NAME	LAST_NAME	BIRTH_DAY
1	AAAAAAAAMKJPHPBA	Dr.	Christopher	Schroeder	10
2	AAAAAAAAONMOGPBA	Miss	Rosalinda	Pratt	1
3	AAAAAAAAGPMHJPBA	Miss	Bernice	Brooks	8
4	AAAAAAAAFEKPGPBA	Dr.	Melodie	Roman	3
5	AAAAAAAANIEOPIAA	Sir	James	Laplante	2
6	AAAAAAAAAABCLABA	Mr.	Michael	Marks	24
7	AAAAAAAAJDFPBLAA	Ms.	Lizzie	Medley	11
8	AAAAAAAAJFBBDLAA	Mr.	Oren	Alonzo	19
9	AAAAAAAAJLDFHPBA	Dr.	Michael	Macon	18
10	AAAAAAAACMHFIPBA	Mr.	Victor	Mendez	5
11	AAAAAAAAFOKMIKCA	Miss	Deborah	Willis	23
12	AAAAAAAANOKIDJBA	Ms.	Josephine	Blais	5
13	AAAAAAAADDAEDJBA	Miss	Dorothy	Duke	18
14	AAAAAAAAILLOEJBA	Sir	Jose	Campbell	25
15	AAAAAAAALFGEFJBA	Dr.	Alfred	Hair	27
16	AAAAAAAANPGJFJBA	Mr.	Walter	Milligan	1

LANDING_CUSTOMER 20 Rows

CUSTOMER_ID	VARCHAR(16777216)
SALUTATION	VARCHAR(16777216)
FIRST_NAME	VARCHAR(16777216)
LAST_NAME	VARCHAR(16777216)
BIRTH_DAY	VARCHAR(16777216)
BIRTH_MONTH	VARCHAR(16777216)
BIRTH_YEAR	VARCHAR(16777216)
BIRTH_COUNTRY	VARCHAR(16777216)
EMAIL_ADDRESS	VARCHAR(16777216)

```
  from ch19.landing_zone.landi

SHOW TABLES;
```

	created_on
1	2024-04-15 05:05:30.914 -
2	2024-04-15 05:05:38.080 -
3	2024-04-15 05:05:43.478 -

Part –2 Curated Zone

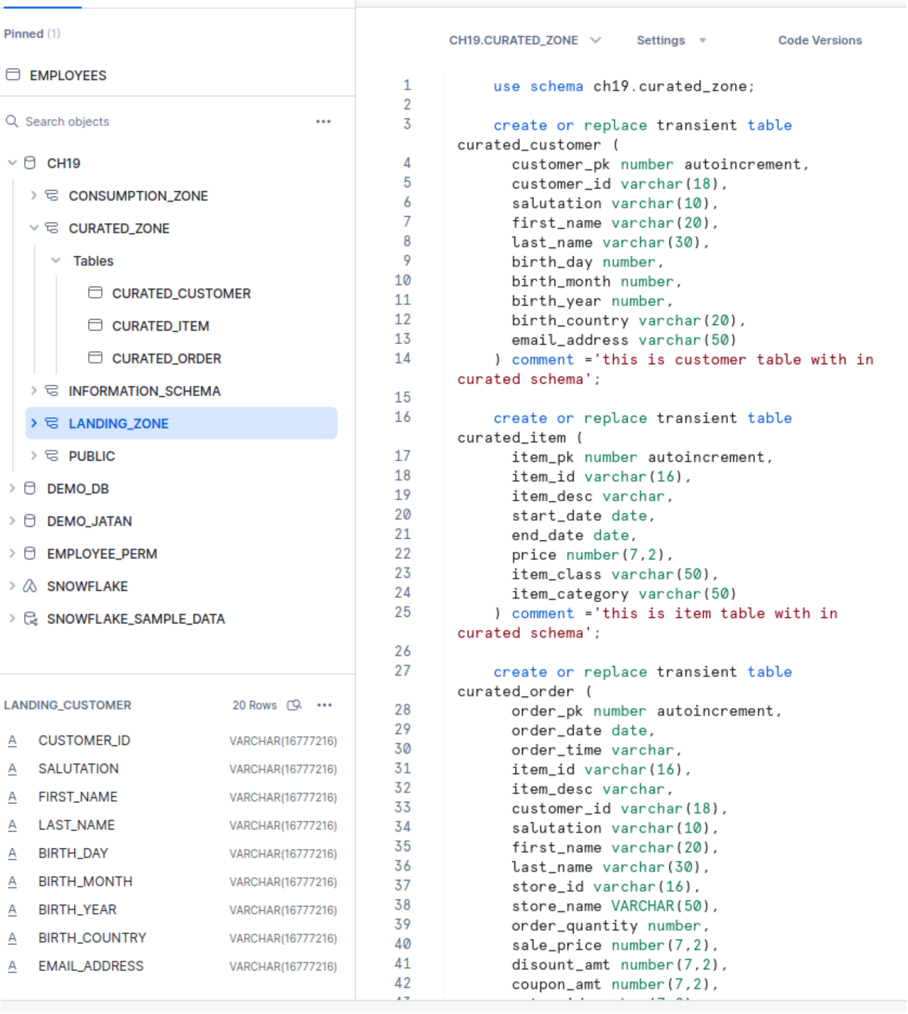
For script check 02-curated-zone-ETL file
Tasks:

1. Create table under Curated Zone Schema.

2. Load data to curated zone as one time load.

3. Verify the tables and table.

1. Create table under Curated Zone Schema.

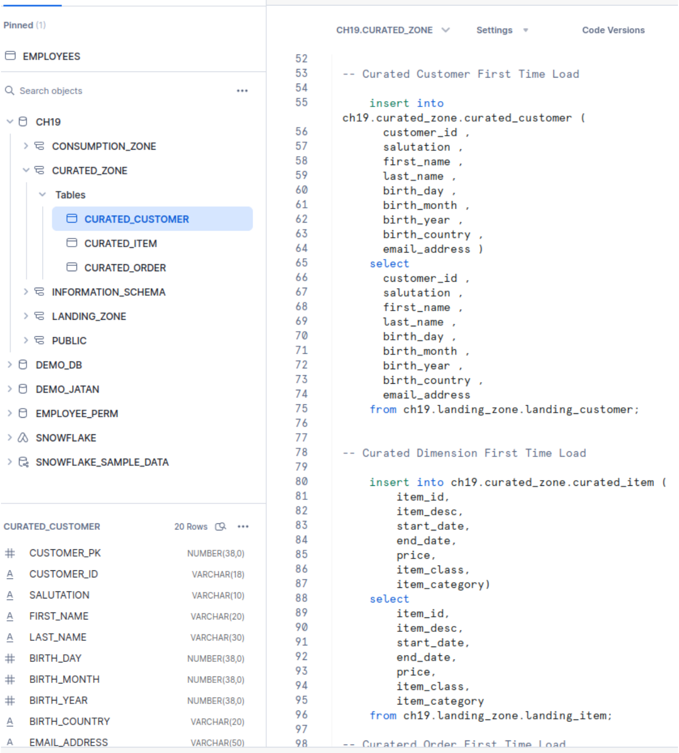


LANDING_CUSTOMER 20 Rows

CUSTOMER_ID	VARCHAR(16777216)
SALUTATION	VARCHAR(16777216)
FIRST_NAME	VARCHAR(16777216)
LAST_NAME	VARCHAR(16777216)
BIRTH_DAY	VARCHAR(16777216)
BIRTH_MONTH	VARCHAR(16777216)
BIRTH_YEAR	VARCHAR(16777216)
BIRTH_COUNTRY	VARCHAR(16777216)
EMAIL_ADDRESS	VARCHAR(16777216)

```
    use schema ch19.curated_zone;

    create or replace transient table curated_customer (
        customer_pk number autoincrement,
        customer_id varchar(18),
        salutation varchar(10),
        first_name varchar(20),
        last_name varchar(30),
        birth_day number,
        birth_month number,
        birth_year number,
        birth_country varchar(20),
        email_address varchar(50)
    ) comment ='this is customer table with in curated schema';

    create or replace transient table curated_item (
        item_pk number autoincrement,
        item_id varchar(16),
        item_desc varchar,
        start_date date,
        end_date date,
        price number(7,2),
        item_class varchar(50),
        item_category varchar(50)
    ) comment ='this is item table with in curated schema';

    create or replace transient table curated_order (
        order_pk number autoincrement,
        order_date date,
        order_time varchar,
        item_id varchar(16),
        item_desc varchar,
        customer_id varchar(18),
        salutation varchar(10),
        first_name varchar(20),
        last_name varchar(30),
        store_id varchar(16),
        store_name VARCHAR(50),
        order_quantity number,
        sale_price number(7,2),
        disount_amt number(7,2),
        coupon_amt number(7,2),
```

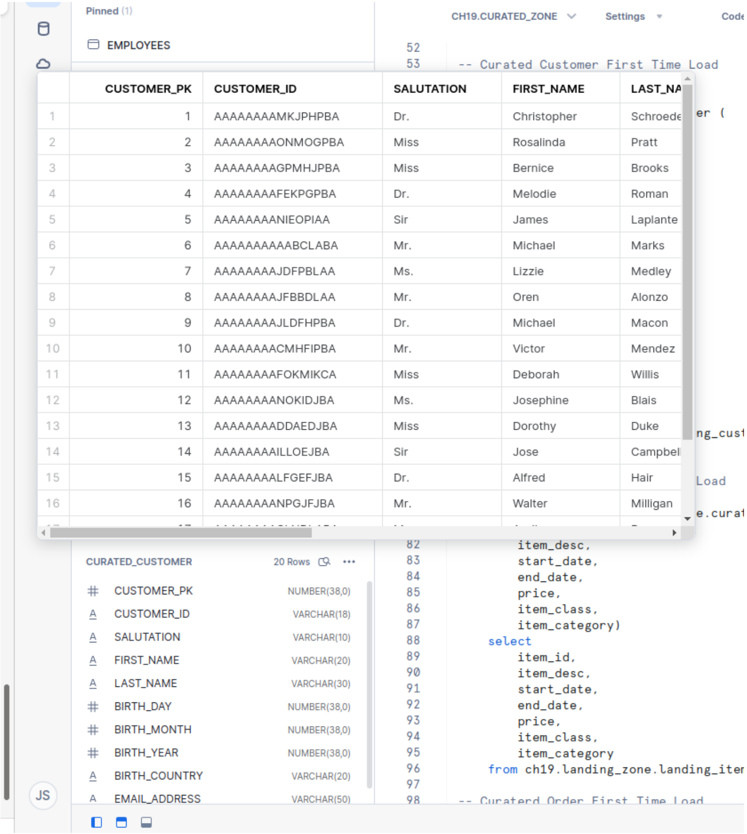
2. Load data to curated zone as one time load.



CUSTOMER_PK	NUMBER(38,0)
CUSTOMER_ID	VARCHAR(18)
SALUTATION	VARCHAR(10)
FIRST_NAME	VARCHAR(20)
LAST_NAME	VARCHAR(30)
BIRTH_DAY	NUMBER(38,0)
BIRTH_MONTH	NUMBER(38,0)
BIRTH_YEAR	NUMBER(38,0)
BIRTH_COUNTRY	VARCHAR(20)
EMAIL_ADDRESS	VARCHAR(50)

```sql

-- Curated Customer First Time Load

    insert into
ch19.curated_zone.curated_customer (
      customer_id ,
      salutation ,
      first_name ,
      last_name ,
      birth_day ,
      birth_month ,
      birth_year ,
      birth_country ,
      email_address )
    select
      customer_id ,
      salutation ,
      first_name ,
      last_name ,
      birth_day ,
      birth_month ,
      birth_year ,
      birth_country ,
      email_address
    from ch19.landing_zone.landing_customer;


-- Curated Dimension First Time Load

    insert into ch19.curated_zone.curated_item (
        item_id,
        item_desc,
        start_date,
        end_date,
        price,
        item_class,
        item_category)
    select
        item_id,
        item_desc,
        start_date,
        end_date,
        price,
        item_class,
        item_category
    from ch19.landing_zone.landing_item;

-- Curaterd Order First Time Load
```

3. Verify data



	CUSTOMER_PK	CUSTOMER_ID	SALUTATION	FIRST_NAME	LAST_NA
1	1	AAAAAAAAMKJPHPBA	Dr.	Christopher	Schroede
2	2	AAAAAAAAONMOGPBA	Miss	Rosalinda	Pratt
3	3	AAAAAAAAGPMHJPBA	Miss	Bernice	Brooks
4	4	AAAAAAAAFEKPGPBA	Dr.	Melodie	Roman
5	5	AAAAAAAANIEOPIAA	Sir	James	Laplante
6	6	AAAAAAAAAABCLABA	Mr.	Michael	Marks
7	7	AAAAAAAAJDFPBLAA	Ms.	Lizzie	Medley
8	8	AAAAAAAAJFBBDLAA	Mr.	Oren	Alonzo
9	9	AAAAAAAAJLDFHPBA	Dr.	Michael	Macon
10	10	AAAAAAAACMHFIPBA	Mr.	Victor	Mendez
11	11	AAAAAAAAFOKMIKCA	Miss	Deborah	Willis
12	12	AAAAAAAANOKIDJBA	Ms.	Josephine	Blais
13	13	AAAAAAAADDAEDJBA	Miss	Dorothy	Duke
14	14	AAAAAAAAILLOEJBA	Sir	Jose	Campbel
15	15	AAAAAAAALFGEFJBA	Dr.	Alfred	Hair
16	16	AAAAAAAANPGJFJBA	Mr.	Walter	Milligan

CUSTOMER_PK	NUMBER(38,0)
CUSTOMER_ID	VARCHAR(18)
SALUTATION	VARCHAR(10)
FIRST_NAME	VARCHAR(20)
LAST_NAME	VARCHAR(30)
BIRTH_DAY	NUMBER(38,0)
BIRTH_MONTH	NUMBER(38,0)
BIRTH_YEAR	NUMBER(38,0)
BIRTH_COUNTRY	VARCHAR(20)
EMAIL_ADDRESS	VARCHAR(50)

```sql

-- Curated Customer First Time Load
                                   er (
                                   ng_cust
                                   Load
                                   e.curat
        item_desc,
        start_date,
        end_date,
        price,
        item_class,
        item_category)
    select
        item_id,
        item_desc,
        start_date,
        end_date,
        price,
        item_class,
        item_category
    from ch19.landing_zone.landing_iter

-- Curaterd Order First Time Load
```

-------------PART 03--------------------

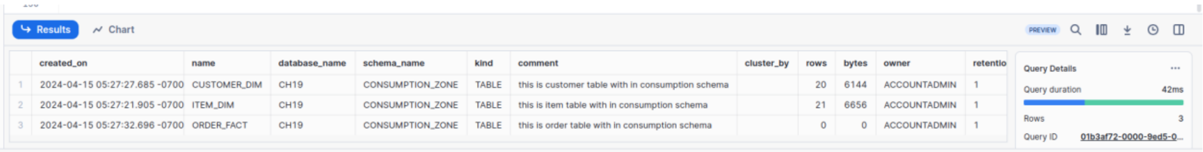
-- 1. Create Tables(DDL) under consumption zone schema.

-- 2. Load data from curated zone to consumption zone as one time load.

---> In order table we have Load order data from curated order to consumtion order fact table

here we have to join the data and data is aggregated at day level.

-- 3. Vertify the tables and data under dimension/fact tables.



	created_on	name	database_name	schema_name	kind	comment	cluster_by	rows	bytes	owner	retentio
1	2024-04-15 05:27:27.685 -0700	CUSTOMER_DIM	CH19	CONSUMPTION_ZONE	TABLE	this is customer table with in consumption schema		20	6144	ACCOUNTADMIN	1
2	2024-04-15 05:27:21.905 -0700	ITEM_DIM	CH19	CONSUMPTION_ZONE	TABLE	this is item table with in consumption schema		21	6656	ACCOUNTADMIN	1
3	2024-04-15 05:27:32.696 -0700	ORDER_FACT	CH19	CONSUMPTION_ZONE	TABLE	this is order table with in consumption schema		0	0	ACCOUNTADMIN	1

Query Details
Query duration 42ms
Rows 3
Query ID 01b3af72-0000-9ed5-0...

As we have less amount of data, we are getting zero rows in order table after joining it

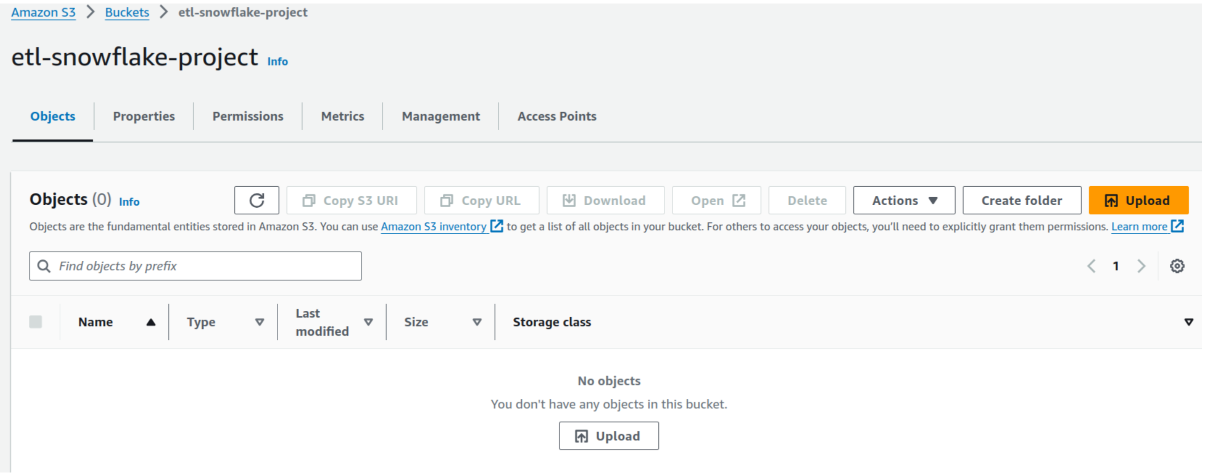
------------------PART 04-------------------

-- TASKS:

-- 1. Create stages and pipes in landing zone schemas.

-- 2. Verify the S3 bucket + notification services.

1. Create stages and pipes in landing zone schemas.
Step 1 : Created S3 bucket

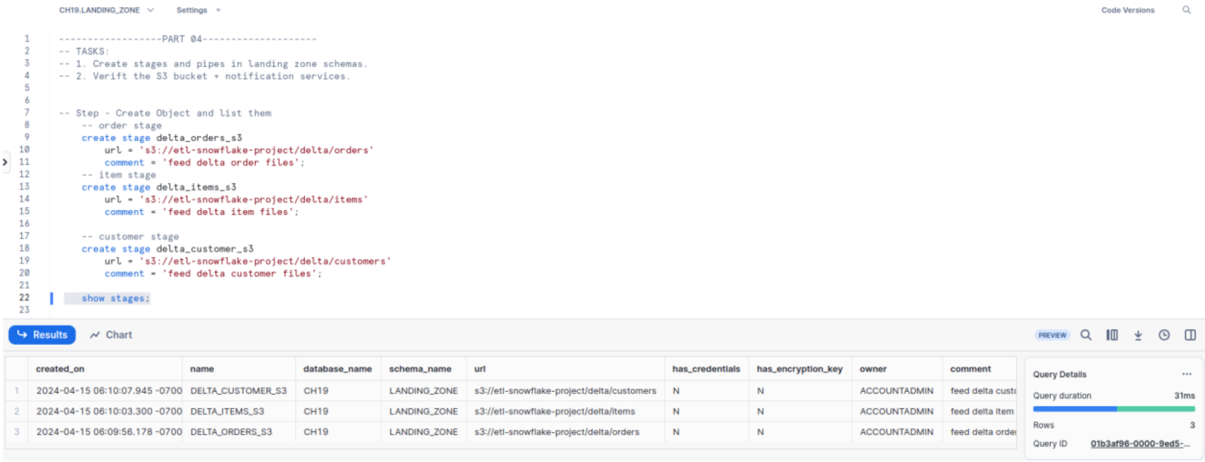


Objects (0) Info

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

No objects
You don't have any objects in this bucket.

Step 2: Created stages

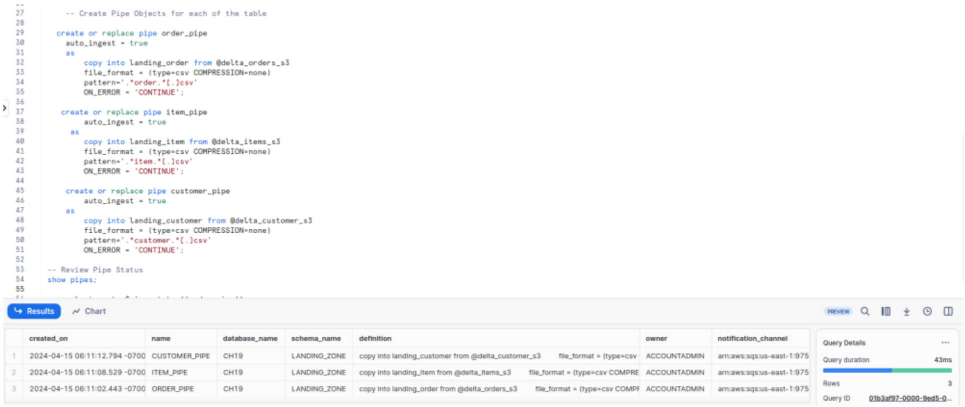


```sql
-------------------PART 04--------------------
-- TASKS:
-- 1. Create stages and pipes in landing zone schemas.
-- 2. Verift the S3 bucket + notification services.


-- Step - Create Object and list them
    -- order stage
    create stage delta_orders_s3
        url = 's3://etl-snowflake-project/delta/orders'
        comment = 'feed delta order files';
    -- item stage
    create stage delta_items_s3
        url = 's3://etl-snowflake-project/delta/items'
        comment = 'feed delta item files';

    -- customer stage
    create stage delta_customer_s3
        url = 's3://etl-snowflake-project/delta/customers'
        comment = 'feed delta customer files';

    show stages;
```

	created_on	name	database_name	schema_name	url	has_credentials	has_encryption_key	owner	comment
1	2024-04-15 06:10:07.945 -0700	DELTA_CUSTOMER_S3	CH19	LANDING_ZONE	s3://etl-snowflake-project/delta/customers	N	N	ACCOUNTADMIN	feed delta custo
2	2024-04-15 06:10:03.300 -0700	DELTA_ITEMS_S3	CH19	LANDING_ZONE	s3://etl-snowflake-project/delta/items	N	N	ACCOUNTADMIN	feed delta item
3	2024-04-15 06:09:56.178 -0700	DELTA_ORDERS_S3	CH19	LANDING_ZONE	s3://etl-snowflake-project/delta/orders	N	N	ACCOUNTADMIN	feed delta orde

Query Details
Query duration 31ms
Rows 3
Query ID 01b3af96-0000-9ed5-...

Step 3: Created Pipes

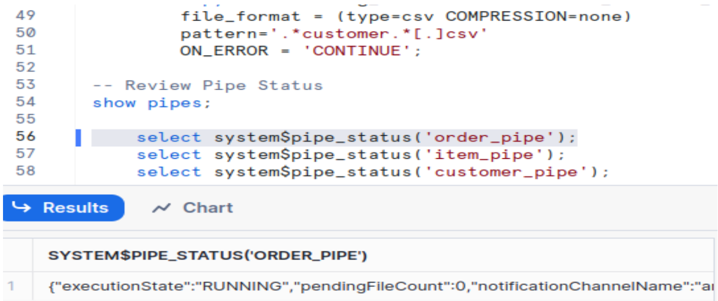


```sql
    -- Create Pipe Objects for each of the table

  create or replace pipe order_pipe
    auto_ingest = true
    as
        copy into landing_order from @delta_orders_s3
        file_format = (type=csv COMPRESSION=none)
        pattern='.*order.*[.]csv'
        ON_ERROR = 'CONTINUE';

   create or replace pipe item_pipe
        auto_ingest = true
     as
        copy into landing_item from @delta_items_s3
        file_format = (type=csv COMPRESSION=none)
        pattern='.*item.*[.]csv'
        ON_ERROR = 'CONTINUE';

    create or replace pipe customer_pipe
        auto_ingest = true
    as
        copy into landing_customer from @delta_customer_s3
        file_format = (type=csv COMPRESSION=none)
        pattern='.*customer.*[.]csv'
        ON_ERROR = 'CONTINUE';

-- Review Pipe Status
show pipes;
```

	created_on	name	database_name	schema_name	definition	owner	notification_channel
1	2024-04-15 06:11:12.794 -0700	CUSTOMER_PIPE	CH19	LANDING_ZONE	copy into landing_customer from @delta_customer_s3 file_format = (type=csv	ACCOUNTADMIN	arn:aws:sqs:us-east-1:975
2	2024-04-15 06:11:08.529 -0700	ITEM_PIPE	CH19	LANDING_ZONE	copy into landing_item from @delta_items_s3 file_format = (type=csv COMPRE	ACCOUNTADMIN	arn:aws:sqs:us-east-1:975
3	2024-04-15 06:11:02.443 -0700	ORDER_PIPE	CH19	LANDING_ZONE	copy into landing_order from @delta_orders_s3 file_format = (type=csv COMPF	ACCOUNTADMIN	arn:aws:sqs:us-east-1:975

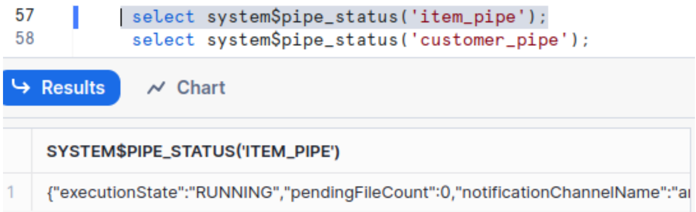
Query Details
Query duration 43ms
Rows 3
Query ID 01b3af97-0000-9ed5-0...

Review pipes running status



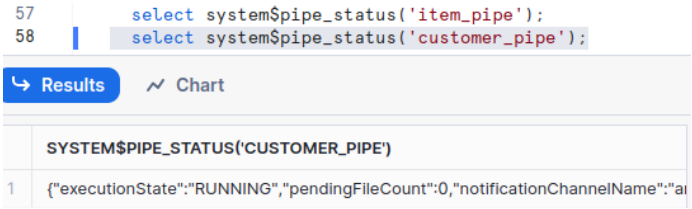
```sql
            file_format = (type=csv COMPRESSION=none)
            pattern='.*customer.*[.]csv'
            ON_ERROR = 'CONTINUE';

    -- Review Pipe Status
    show pipes;

        select system$pipe_status('order_pipe');
        select system$pipe_status('item_pipe');
        select system$pipe_status('customer_pipe');
```

	SYSTEM$PIPE_STATUS('ORDER_PIPE')
1	{"executionState":"RUNNING","pendingFileCount":0,"notificationChannelName":"ar



```sql
        select system$pipe_status('item_pipe');
        select system$pipe_status('customer_pipe');
```

	SYSTEM$PIPE_STATUS('ITEM_PIPE')
1	{"executionState":"RUNNING","pendingFileCount":0,"notificationChannelName":"ar

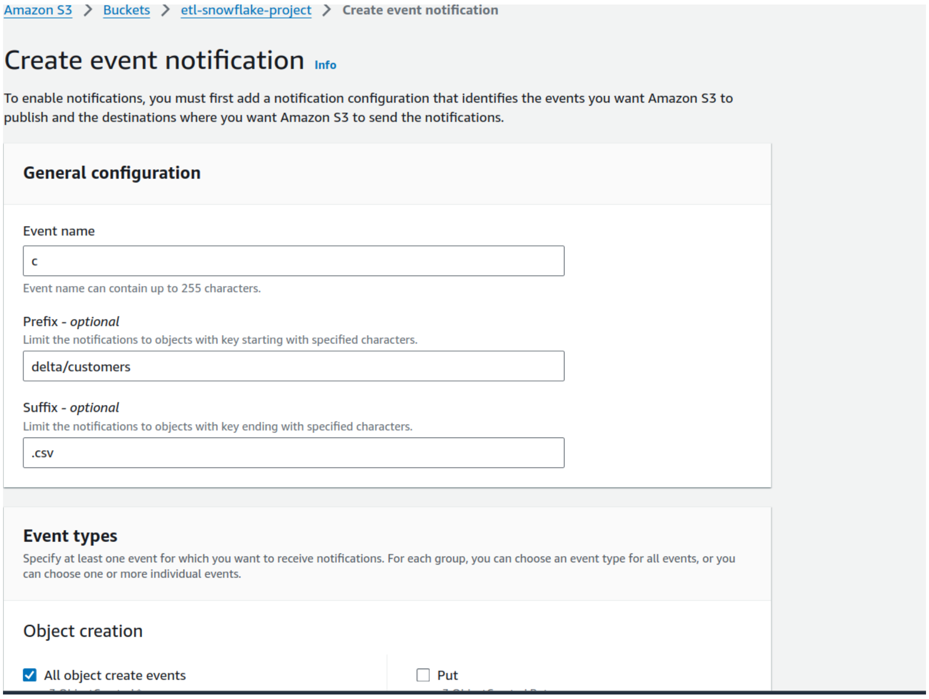


```sql
        select system$pipe_status('item_pipe');
        select system$pipe_status('customer_pipe');
```

	SYSTEM$PIPE_STATUS('CUSTOMER_PIPE')
1	{"executionState":"RUNNING","pendingFileCount":0,"notificationChannelName":"ar

Step 4: Create SQS notification for bucket in S3

Go to bucket > Create event notification



Amazon S3 > Buckets > etl-snowflake-project > Create event notification

Create event notification

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

c

Event name can contain up to 255 characters.

Prefix - *optional*

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

delta/customers

Suffix - *optional*

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

.csv

Event types

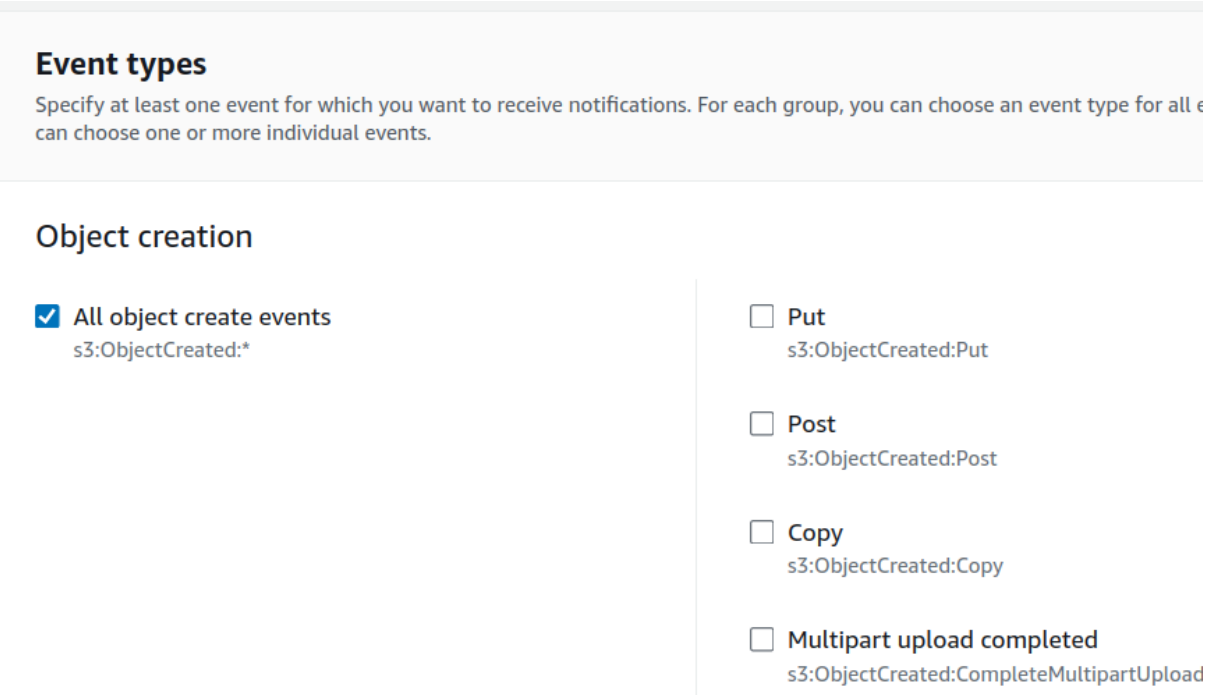
Specify at least one event for which you want to receive notifications. For each group, you can choose an event type for all events, or you can choose one or more individual events.

Object creation

☑ All object create events

□ Put

Select all object create events in event types



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 e can choose one or more individual events.

Object creation

☑ All object create events
s3:ObjectCreated:*

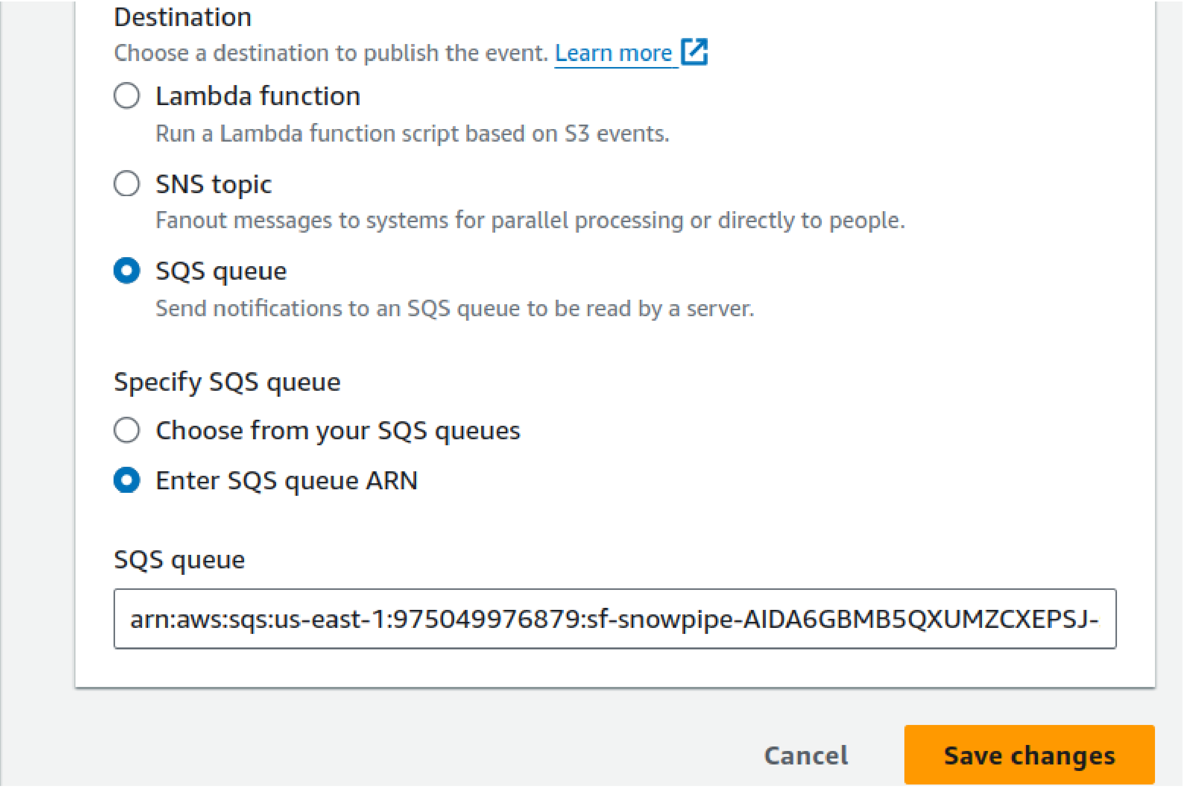
□ Put
s3:ObjectCreated:Put

□ Post
s3:ObjectCreated:Post

□ Copy
s3:ObjectCreated:Copy

□ Multipart upload completed
s3:ObjectCreated:CompleteMultipartUpload

Select SQS queue for sequential notification service. and enter ARN which we can find it in **"SHOW PIPE;"** command in Snowflake (Mention in screenshot)



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.

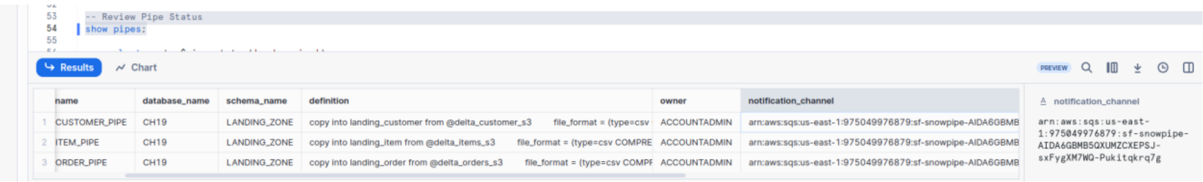
Specify SQS queue

○ Choose from your SQS queues

● Enter SQS queue ARN

SQS queue

arn:aws:sqs:us-east-1:975049976879:sf-snowpipe-AIDA6GBMB5QXUMZCXEPSJ-



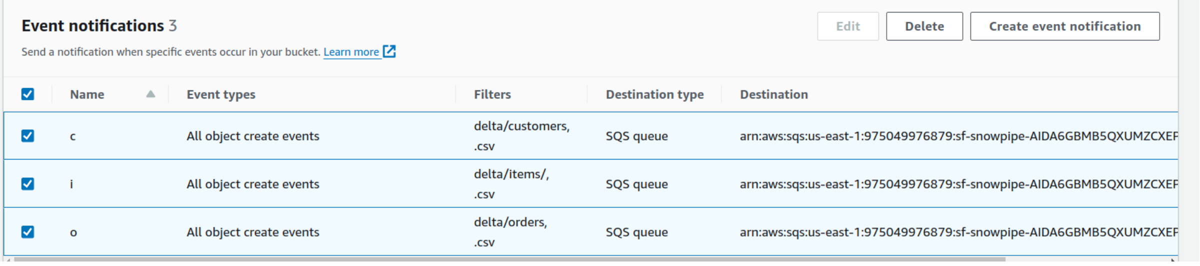
```
-- Review Pipe Status
show pipes;
```

	name	database_name	schema_name	definition	owner	notification_channel
1	CUSTOMER_PIPE	CH19	LANDING_ZONE	copy into landing_customer from @delta_customer_s3 file_format = (type=csv	ACCOUNTADMIN	arn:aws:sqs:us-east-1:975049976879:sf-snowpipe-AIDA6GBMB
2	ITEM_PIPE	CH19	LANDING_ZONE	copy into landing_item from @delta_items_s3 file_format = (type=csv COMPRE	ACCOUNTADMIN	arn:aws:sqs:us-east-1:975049976879:sf-snowpipe-AIDA6GBMB
3	ORDER_PIPE	CH19	LANDING_ZONE	copy into landing_order from @delta_orders_s3 file_format = (type=csv COMPF	ACCOUNTADMIN	arn:aws:sqs:us-east-1:975049976879:sf-snowpipe-AIDA6GBMB

notification_channel

arn:aws:sqs:us-east-1:975049976879:sf-snowpipe-AIDA6GBMB5QXUMZCXEPSJ-sxFygXM7WQ-Pukitqkrq7g

Created 3 SQS



Event notifications 3

Send a notification when specific events occur in your bucket. Learn more

☑	Name	Event types	Filters	Destination type	Destination
☑	c	All object create events	delta/customers, .csv	SQS queue	arn:aws:sqs:us-east-1:975049976879:sf-snowpipe-AIDA6GBMB5QXUMZCXEF
☑	i	All object create events	delta/items/, .csv	SQS queue	arn:aws:sqs:us-east-1:975049976879:sf-snowpipe-AIDA6GBMB5QXUMZCXEF
☑	o	All object create events	delta/orders, .csv	SQS queue	arn:aws:sqs:us-east-1:975049976879:sf-snowpipe-AIDA6GBMB5QXUMZCXEF

-----------PART-05--------------------

-- TASKS:

-- 1. Create streams under landing zone schema.

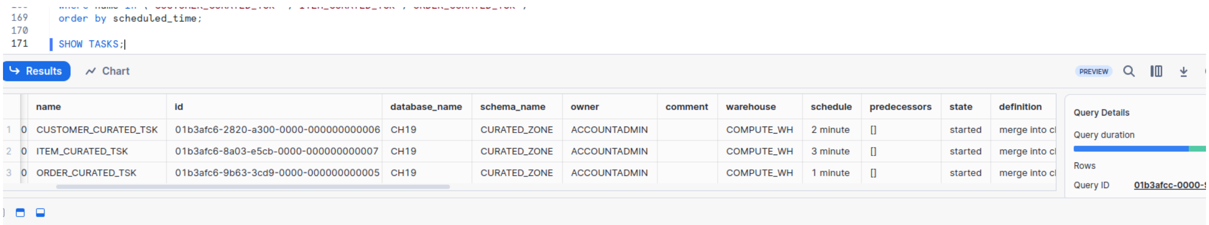
-- 2. Create task under curated zone.

-- 3. Resume task and validate that they are running under curated zone.

-- (Remember - cross schema task linking is not possible - task from one schema cannot call task from other schema)

- 2. Create task under curated zone.

Show task



```
order by scheduled_time;

SHOW TASKS;
```

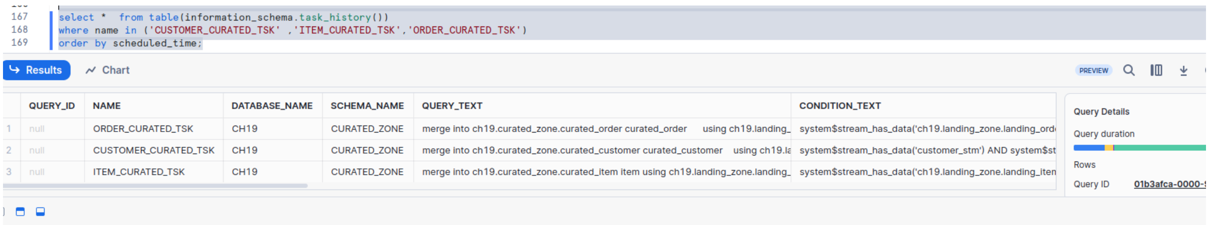
		name	id	database_name	schema_name	owner	comment	warehouse	schedule	predecessors	state	definition
1	0	CUSTOMER_CURATED_TSK	01b3afc6-2820-a300-0000-000000000006	CH19	CURATED_ZONE	ACCOUNTADMIN		COMPUTE_WH	2 minute	[]	started	merge into cı
2	0	ITEM_CURATED_TSK	01b3afc6-8a03-e5cb-0000-000000000007	CH19	CURATED_ZONE	ACCOUNTADMIN		COMPUTE_WH	3 minute	[]	started	merge into cı
3	0	ORDER_CURATED_TSK	01b3afc6-9b63-3cd9-0000-000000000005	CH19	CURATED_ZONE	ACCOUNTADMIN		COMPUTE_WH	1 minute	[]	started	merge into cı

Query Details

Query duration

Rows

Query ID 01b3afcc-0000-



```
select *  from table(information_schema.task_history())
where name in ('CUSTOMER_CURATED_TSK' ,'ITEM_CURATED_TSK','ORDER_CURATED_TSK')
order by scheduled_time;
```

	QUERY_ID	NAME	DATABASE_NAME	SCHEMA_NAME	QUERY_TEXT	CONDITION_TEXT
1	null	ORDER_CURATED_TSK	CH19	CURATED_ZONE	merge into ch19.curated_zone.curated_order curated_order using ch19.landing_	system$stream_has_data('ch19.landing_zone.landing_ord
2	null	CUSTOMER_CURATED_TSK	CH19	CURATED_ZONE	merge into ch19.curated_zone.curated_customer curated_customer using ch19.la	system$stream_has_data('customer_stm') AND system$st
3	null	ITEM_CURATED_TSK	CH19	CURATED_ZONE	merge into ch19.curated_zone.curated_item item using ch19.landing_zone.landing_	system$stream_has_data('ch19.landing_zone.landing_iten

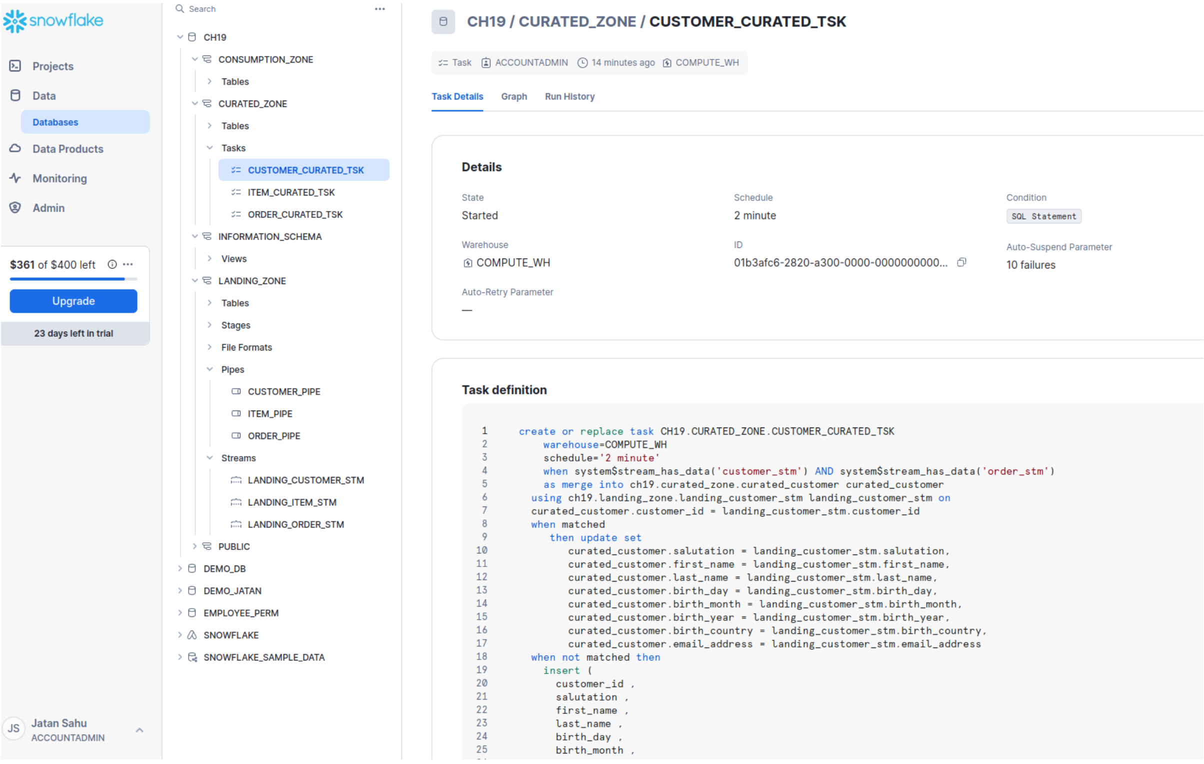
Query Details

Query duration

Rows

Query ID 01b3afca-0000-

All tasks



CH19 / CURATED_ZONE / CUSTOMER_CURATED_TSK

Task ACCOUNTADMIN 14 minutes ago COMPUTE_WH

Details

State: Started

Schedule: 2 minute

Condition: SQL Statement

Warehouse: COMPUTE_WH

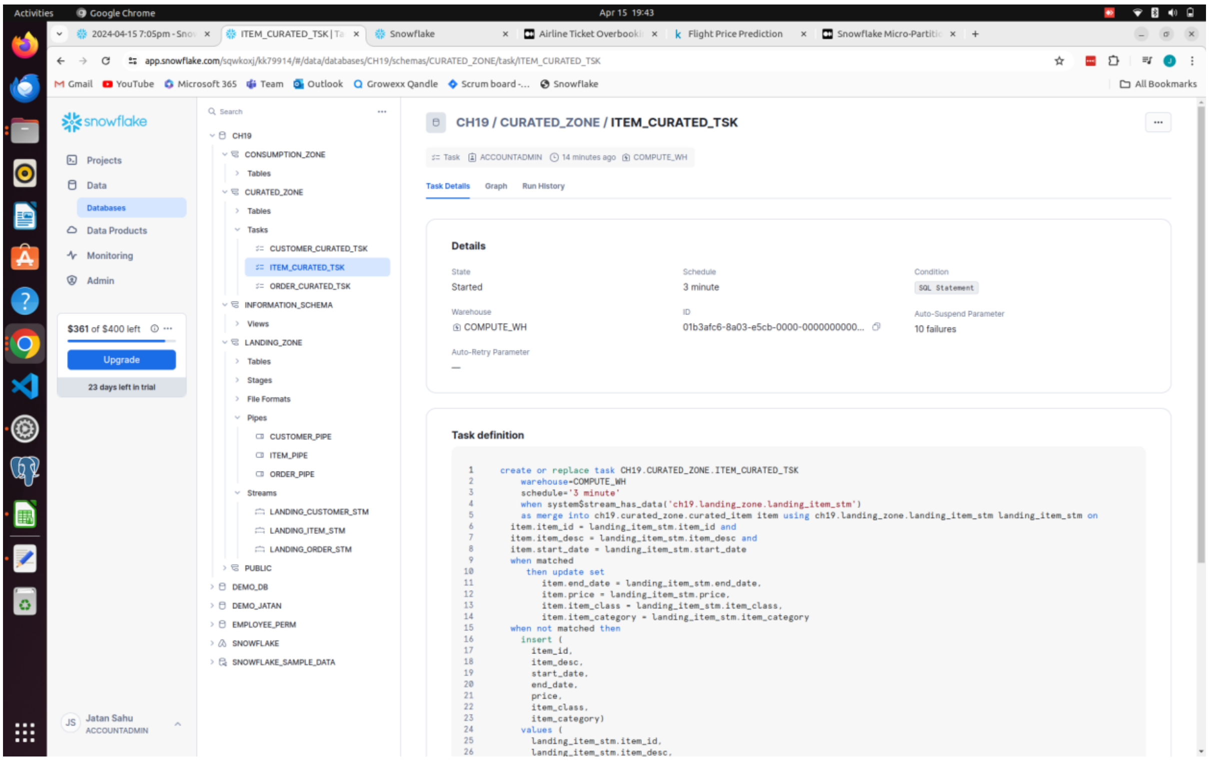
ID: 01b3afc6-2820-a300-0000-0000000000...

Auto-Suspend Parameter: 10 failures

Auto-Retry Parameter: —

Task definition

```
create or replace task CH19.CURATED_ZONE.CUSTOMER_CURATED_TSK
    warehouse=COMPUTE_WH
    schedule='2 minute'
    when system$stream_has_data('customer_stm') AND system$stream_has_data('order_stm')
    as merge into ch19.curated_zone.curated_customer curated_customer
  using ch19.landing_zone.landing_customer_stm landing_customer_stm on
  curated_customer.customer_id = landing_customer_stm.customer_id
  when matched
     then update set
        curated_customer.salutation = landing_customer_stm.salutation,
        curated_customer.first_name = landing_customer_stm.first_name,
        curated_customer.last_name = landing_customer_stm.last_name,
        curated_customer.birth_day = landing_customer_stm.birth_day,
        curated_customer.birth_month = landing_customer_stm.birth_month,
        curated_customer.birth_year = landing_customer_stm.birth_year,
        curated_customer.birth_country = landing_customer_stm.birth_country,
        curated_customer.email_address = landing_customer_stm.email_address
  when not matched then
    insert (
      customer_id ,
      salutation ,
      first_name ,
      last_name ,
      birth_day ,
      birth_month ,
```



CH19 / CURATED_ZONE / ITEM_CURATED_TSK

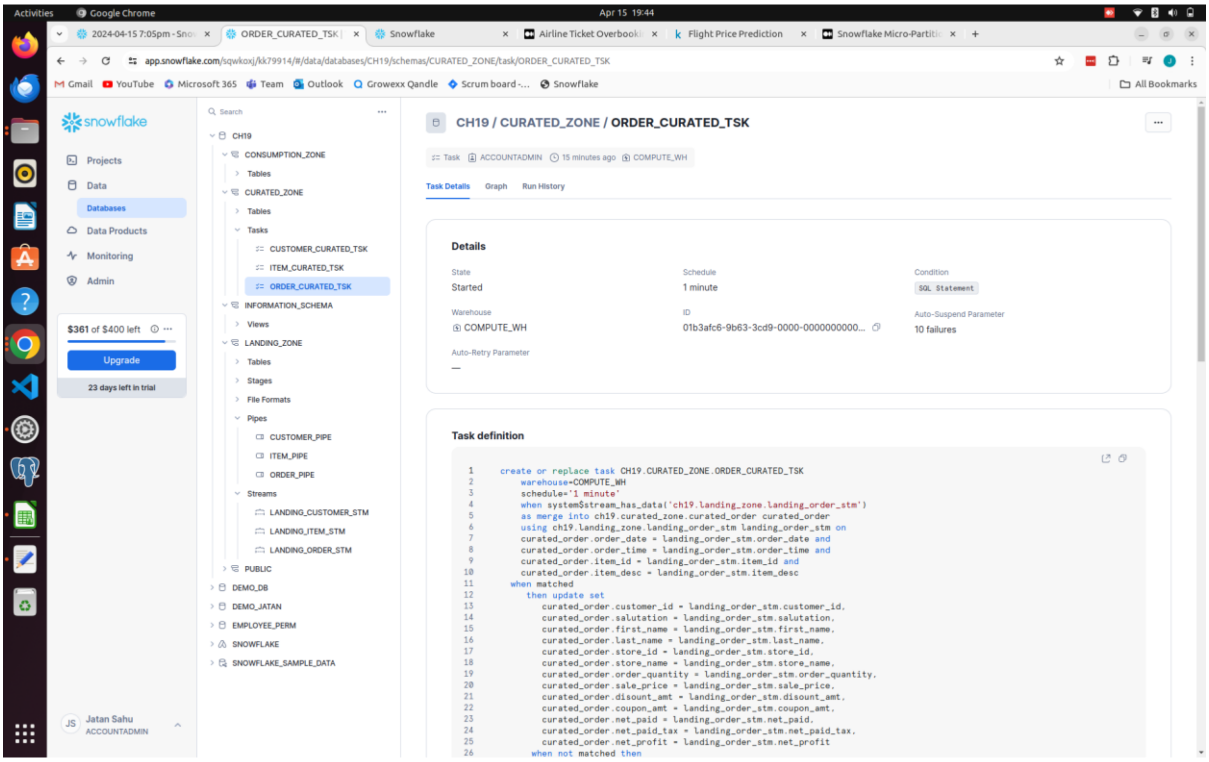
Task · ACCOUNTADMIN · 14 minutes ago · COMPUTE_WH

Details

State: Started
Schedule: 3 minute
Condition: SQL Statement
Warehouse: COMPUTE_WH
ID: 01b3afc6-8a03-e5cb-0000-0000000000...
Auto-Suspend Parameter: 10 failures
Auto-Retry Parameter: —

Task definition

```
create or replace task CH19.CURATED_ZONE.ITEM_CURATED_TSK
    warehouse=COMPUTE_WH
    schedule='3 minute'
    when system$stream_has_data('ch19.landing_zone.landing_item_stm')
    as merge into ch19.curated_zone.curated_item item using ch19.landing_zone.landing_item_stm landing_item_stm on
  item.item_id = landing_item_stm.item_id and
  item.item_desc = landing_item_stm.item_desc and
  item.start_date = landing_item_stm.start_date
  when matched
     then update set
        item.end_date = landing_item_stm.end_date,
        item.price = landing_item_stm.price,
        item.item_class = landing_item_stm.item_class,
        item.item_category = landing_item_stm.item_category
  when not matched then
    insert (
      item_id,
      item_desc,
      start_date,
      end_date,
      price,
      item_class,
      item_category)
    values (
      landing_item_stm.item_id,
      landing_item_stm.item_desc,
```



CH19 / CURATED_ZONE / ORDER_CURATED_TSK

Task · ACCOUNTADMIN · 15 minutes ago · COMPUTE_WH

Details

State: Started
Schedule: 1 minute
Condition: SQL Statement
Warehouse: COMPUTE_WH
ID: 01b3afc6-9b63-3cd9-0000-0000000000...
Auto-Suspend Parameter: 10 failures
Auto-Retry Parameter: —

Task definition

```
create or replace task CH19.CURATED_ZONE.ORDER_CURATED_TSK
    warehouse=COMPUTE_WH
    schedule='1 minute'
    when system$stream_has_data('ch19.landing_zone.landing_order_stm')
    as merge into ch19.curated_zone.curated_order curated_order
    using ch19.landing_zone.landing_order_stm landing_order_stm on
    curated_order.order_date = landing_order_stm.order_date and
    curated_order.order_time = landing_order_stm.order_time and
    curated_order.item_id = landing_order_stm.item_id and
    curated_order.item_desc = landing_order_stm.item_desc
  when matched
     then update set
        curated_order.customer_id = landing_order_stm.customer_id,
        curated_order.salutation = landing_order_stm.salutation,
        curated_order.first_name = landing_order_stm.first_name,
        curated_order.last_name = landing_order_stm.last_name,
        curated_order.store_id = landing_order_stm.store_id,
        curated_order.store_name = landing_order_stm.store_name,
        curated_order.order_quantity = landing_order_stm.order_quantity,
        curated_order.sale_price = landing_order_stm.sale_price,
        curated_order.disount_amt = landing_order_stm.disount_amt,
        curated_order.coupon_amt = landing_order_stm.coupon_amt,
        curated_order.net_paid = landing_order_stm.net_paid,
        curated_order.net_paid_tax = landing_order_stm.net_paid_tax,
        curated_order.net_profit = landing_order_stm.net_profit
     when not matched then
```

--------------------PART-06--------------------------

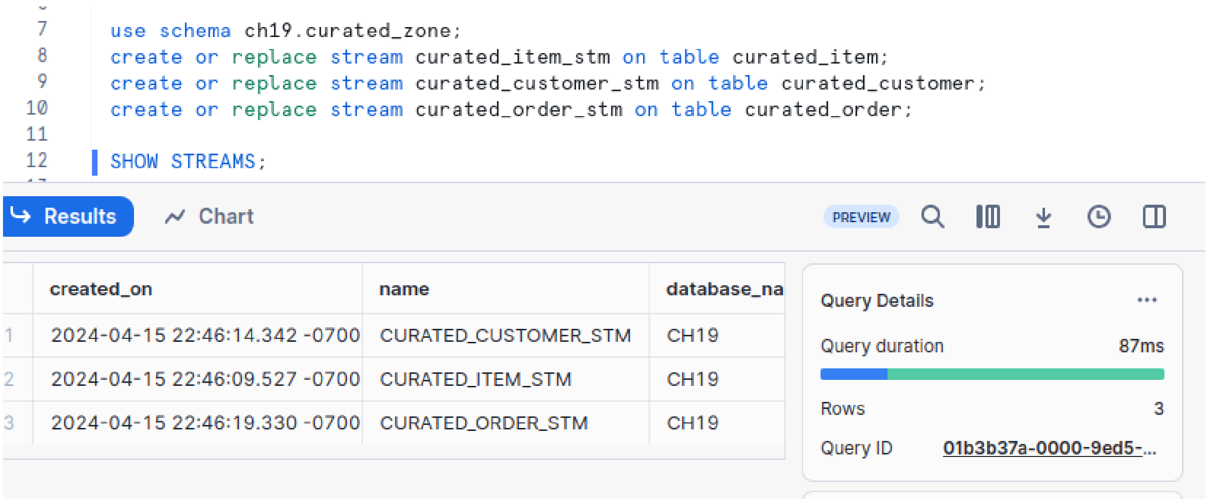
-- TASKS:

-- 1.Create streams under curated zone schema.

-- 2.Create task under consumption zone schema.

-- 3.Resume task and validate that they are running under consumption zone.

-- 1.Create streams under curated zone schema.



```
use schema ch19.curated_zone;
create or replace stream curated_item_stm on table curated_item;
create or replace stream curated_customer_stm on table curated_customer;
create or replace stream curated_order_stm on table curated_order;

SHOW STREAMS;
```

	created_on	name	database_na
1	2024-04-15 22:46:14.342 -0700	CURATED_CUSTOMER_STM	CH19
2	2024-04-15 22:46:09.527 -0700	CURATED_ITEM_STM	CH19
3	2024-04-15 22:46:19.330 -0700	CURATED_ORDER_STM	CH19

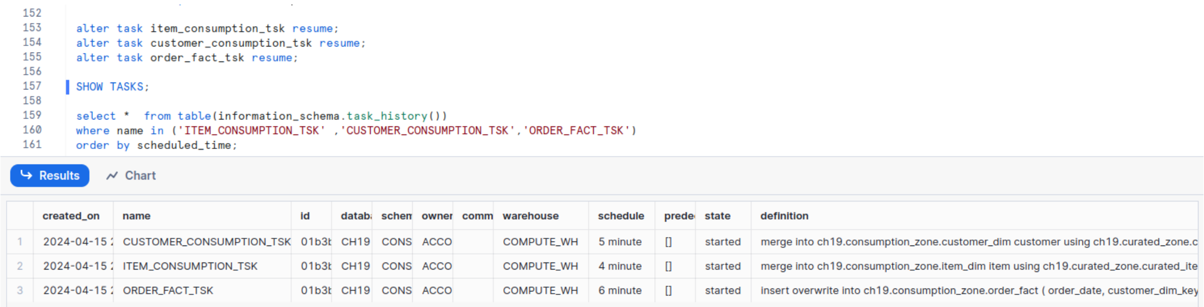
Query Details

Query duration 87ms

Rows 3

Query ID 01b3b37a-0000-9ed5-...

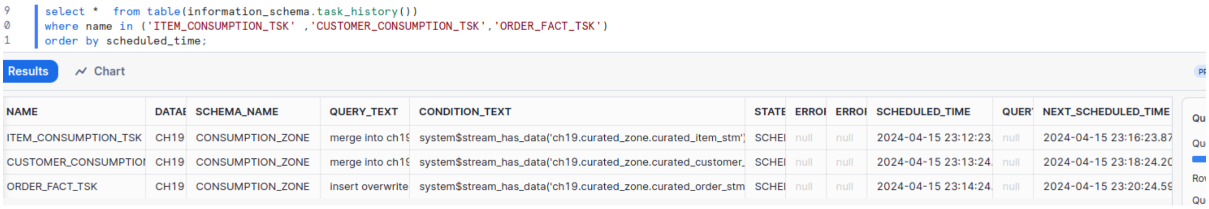
-- 2.Create task under consumption zone schema.



```
alter task item_consumption_tsk resume;
alter task customer_consumption_tsk resume;
alter task order_fact_tsk resume;

SHOW TASKS;

select *  from table(information_schema.task_history())
where name in ('ITEM_CONSUMPTION_TSK' ,'CUSTOMER_CONSUMPTION_TSK','ORDER_FACT_TSK')
order by scheduled_time;
```

	created_on	name	id	datab	schen	owner	comm	warehouse	schedule	prede	state	definition
1	2024-04-15 2	CUSTOMER_CONSUMPTION_TSK	01b3t	CH19	CONS	ACCO		COMPUTE_WH	5 minute	[]	started	merge into ch19.consumption_zone.customer_dim customer using ch19.curated_zone.c
2	2024-04-15 2	ITEM_CONSUMPTION_TSK	01b3t	CH19	CONS	ACCO		COMPUTE_WH	4 minute	[]	started	merge into ch19.consumption_zone.item_dim item using ch19.curated_zone.curated_ite
3	2024-04-15 2	ORDER_FACT_TSK	01b3t	CH19	CONS	ACCO		COMPUTE_WH	6 minute	[]	started	insert overwrite into ch19.consumption_zone.order_fact (order_date, customer_dim_key

- 3.Resume task and validate that they are running under consumption zone.

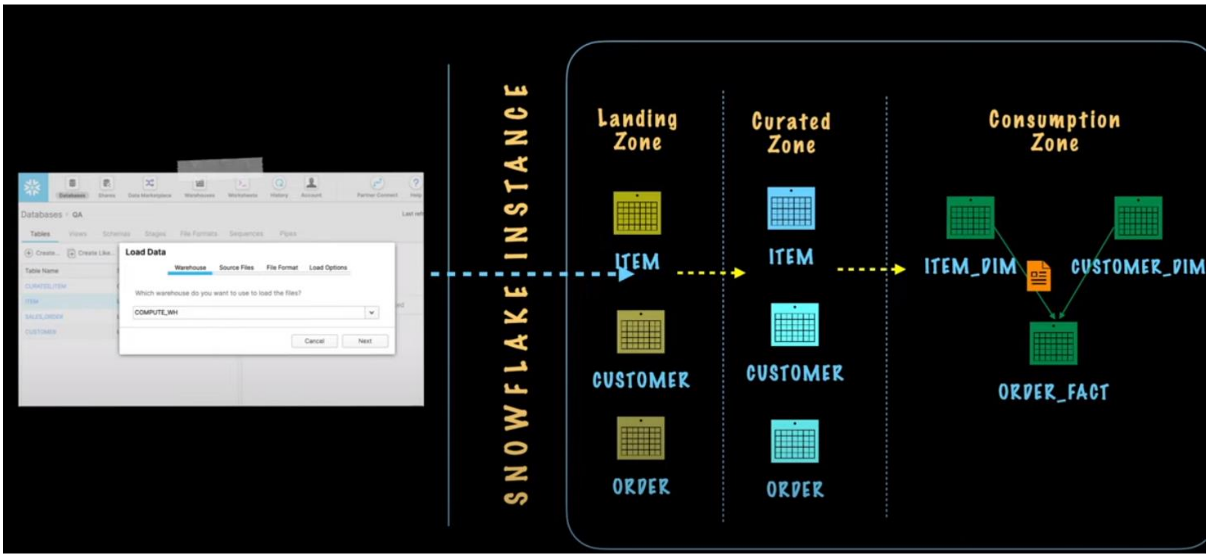


```
select *  from table(information_schema.task_history())
where name in ('ITEM_CONSUMPTION_TSK' ,'CUSTOMER_CONSUMPTION_TSK','ORDER_FACT_TSK')
order by scheduled_time;
```

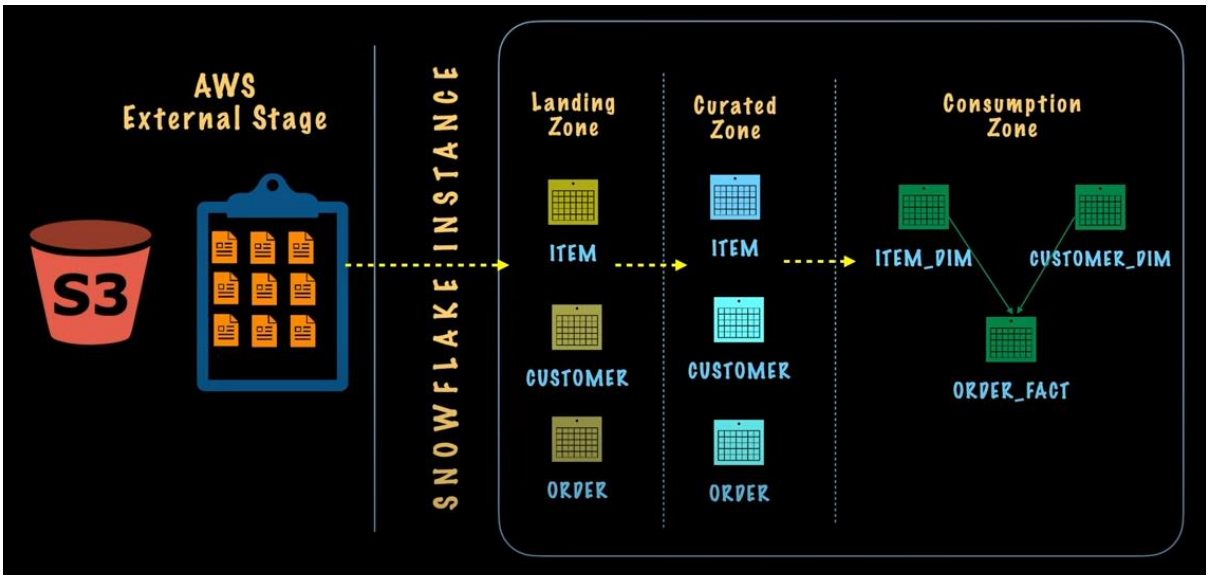
NAME	DATAE	SCHEMA_NAME	QUERY_TEXT	CONDITION_TEXT	STATE	ERROI	ERROI	SCHEDULED_TIME	QUER'	NEXT_SCHEDULED_TIME
ITEM_CONSUMPTION_TSK	CH19	CONSUMPTION_ZONE	merge into ch1¢	system$stream_has_data('ch19.curated_zone.curated_item_stm')	SCHEI	null	null	2024-04-15 23:12:23.	null	2024-04-15 23:16:23.87
CUSTOMER_CONSUMPTIOI	CH19	CONSUMPTION_ZONE	merge into ch1¢	system$stream_has_data('ch19.curated_zone.curated_customer_	SCHEI	null	null	2024-04-15 23:13:24.	null	2024-04-15 23:18:24.2C
ORDER_FACT_TSK	CH19	CONSUMPTION_ZONE	insert overwrite	system$stream_has_data('ch19.curated_zone.curated_order_stm	SCHEI	null	null	2024-04-15 23:14:24.	null	2024-04-15 23:20:24.59

-----------------PART-07---------------------

First check manually by loading data



Then upload using s3 and stream



BEFORE DUMPING DATA :

IN Landing Zone and Curated zone Schema we have:

Customer – 20 rows

Orders – 18rows

Items – 21rows

IN Consumption Zone Schema we have:

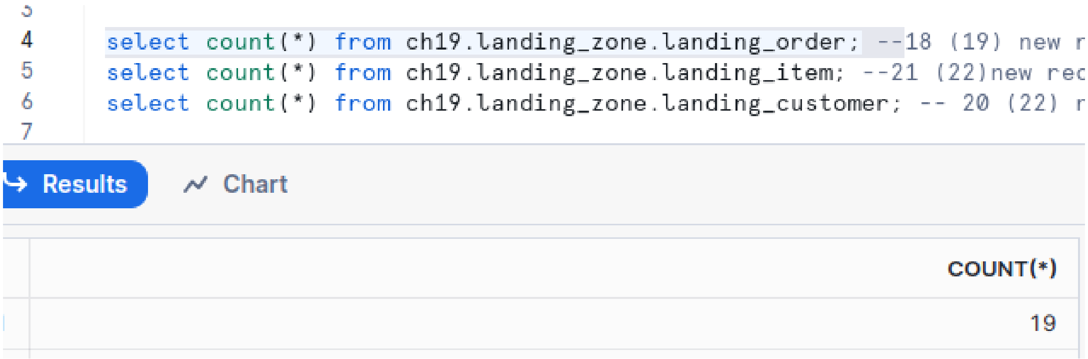
Customer – 20 rows

Orders – 11 rows (filter net_paid >500)

Items – 21 rows

After **INSERTING** one row only:

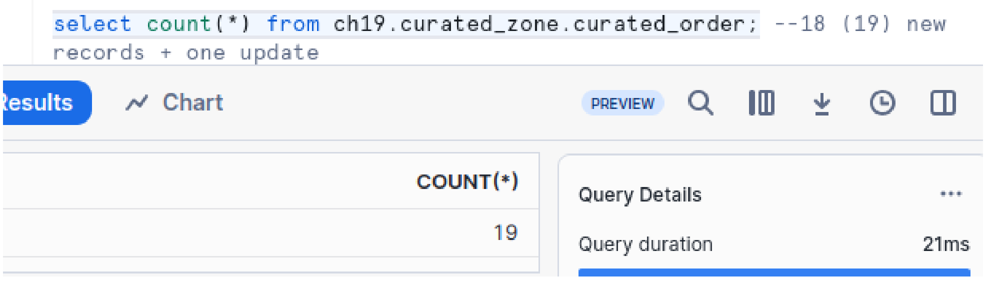
Order table count in landing zone



```
select count(*) from ch19.landing_zone.landing_order; --18 (19) new r
select count(*) from ch19.landing_zone.landing_item; --21 (22)new rec
select count(*) from ch19.landing_zone.landing_customer; -- 20 (22) r
```

```
COUNT(*)
      19
```

Order table count in curated_zone



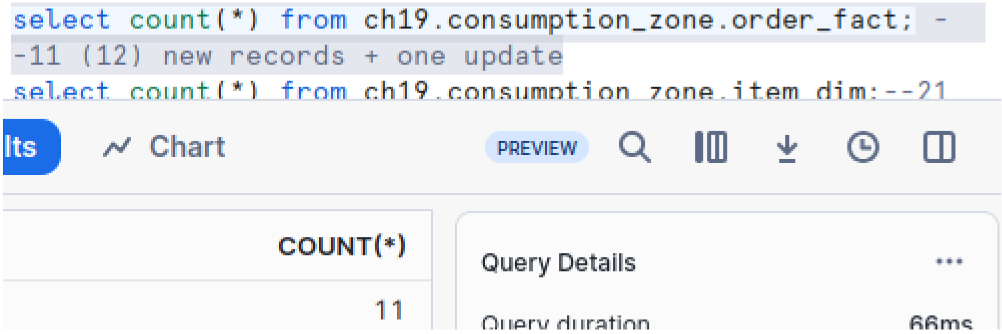
```
select count(*) from ch19.curated_zone.curated_order; --18 (19) new
records + one update
```

```
COUNT(*)
      19
```

Query Details

Query duration 21ms

Order table in Consumption zone



```
select count(*) from ch19.consumption_zone.order_fact; -
-11 (12) new records + one update
select count(*) from ch19.consumption_zone.item_dim;--21
```

```
COUNT(*)
      11
```

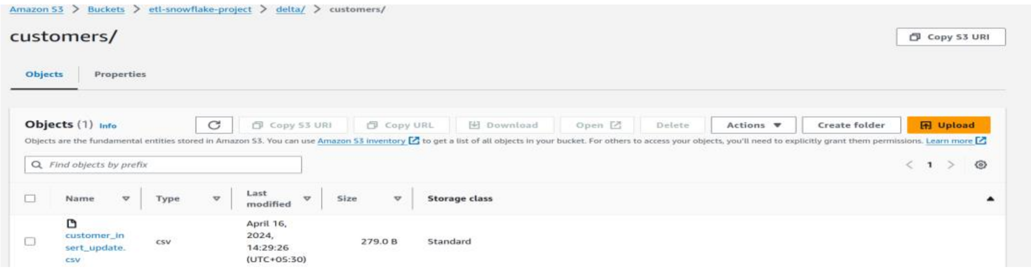
Query Details

Query duration 66ms

Other rable counts are same in all schmas

We have checked manually everthing is working fine

Now we we import data in S3 bucket



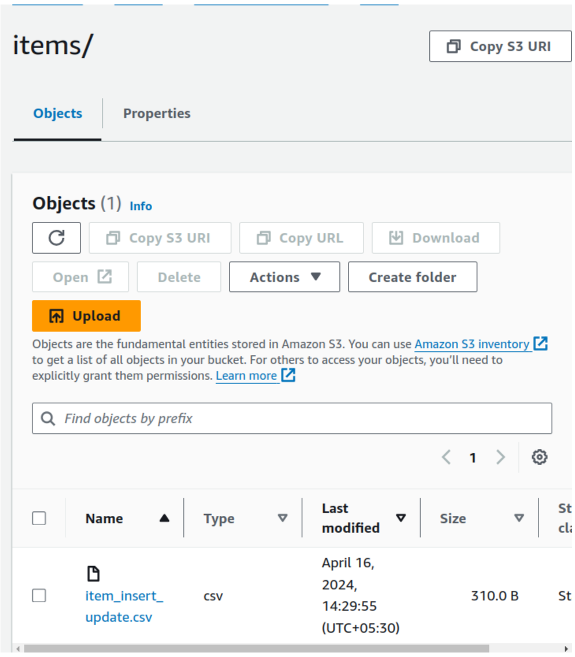
Amazon S3 > Buckets > etl-snowflake-project > delta/ > customers/

customers/

Objects (1)

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

□	Name	Type	Last modified	Size	Storage class
□	customer_insert_update.csv	csv	April 16, 2024, 14:29:26 (UTC+05:30)	279.0 B	Standard



items/

Copy S3 URI

Objects | Properties

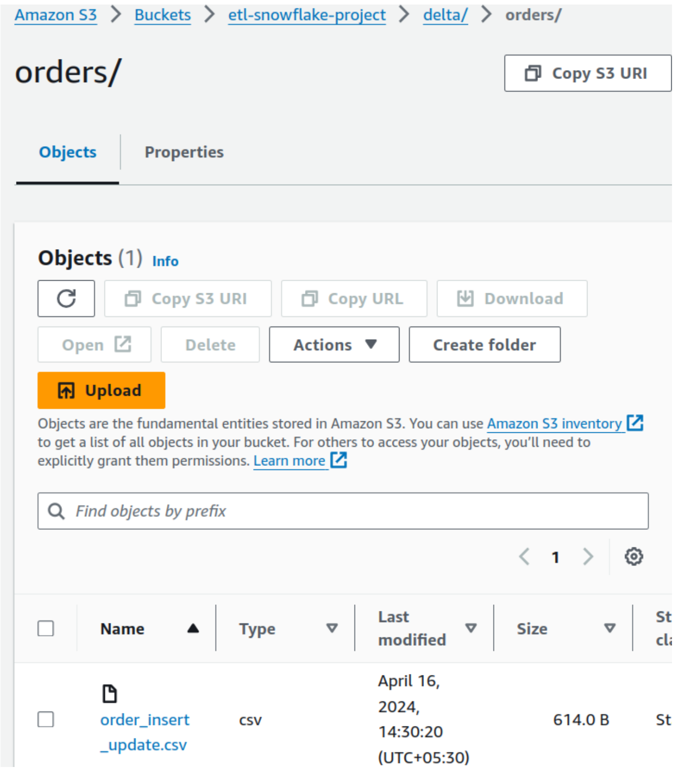
Objects (1) Info

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

Find objects by prefix

< 1 >

□	Name ▲	Type ▽	Last modified ▼	Size ▽	St cla
□	item_insert_update.csv	csv	April 16, 2024, 14:29:55 (UTC+05:30)	310.0 B	St



Amazon S3 > Buckets > etl-snowflake-project > delta/ > orders/

orders/

Copy S3 URI

Objects | Properties

Objects (1) Info

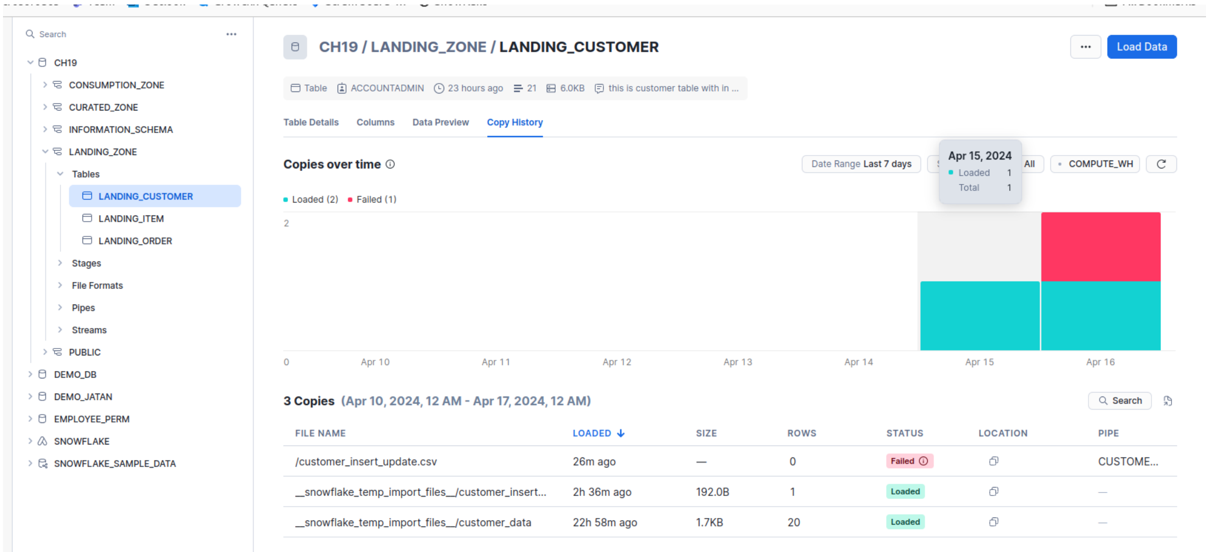
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

Find objects by prefix

< 1 >

□	Name ▲	Type ▽	Last modified ▼	Size ▽	St cla
□	order_insert_update.csv	csv	April 16, 2024, 14:30:20 (UTC+05:30)	614.0 B	St

Getting error in first trial



CH19 / LANDING_ZONE / LANDING_CUSTOMER

Table ACCOUNTADMIN 23 hours ago 21 6.0KB this is customer table with in ...

Copies over time

3 Copies (Apr 10, 2024, 12 AM - Apr 17, 2024, 12 AM)

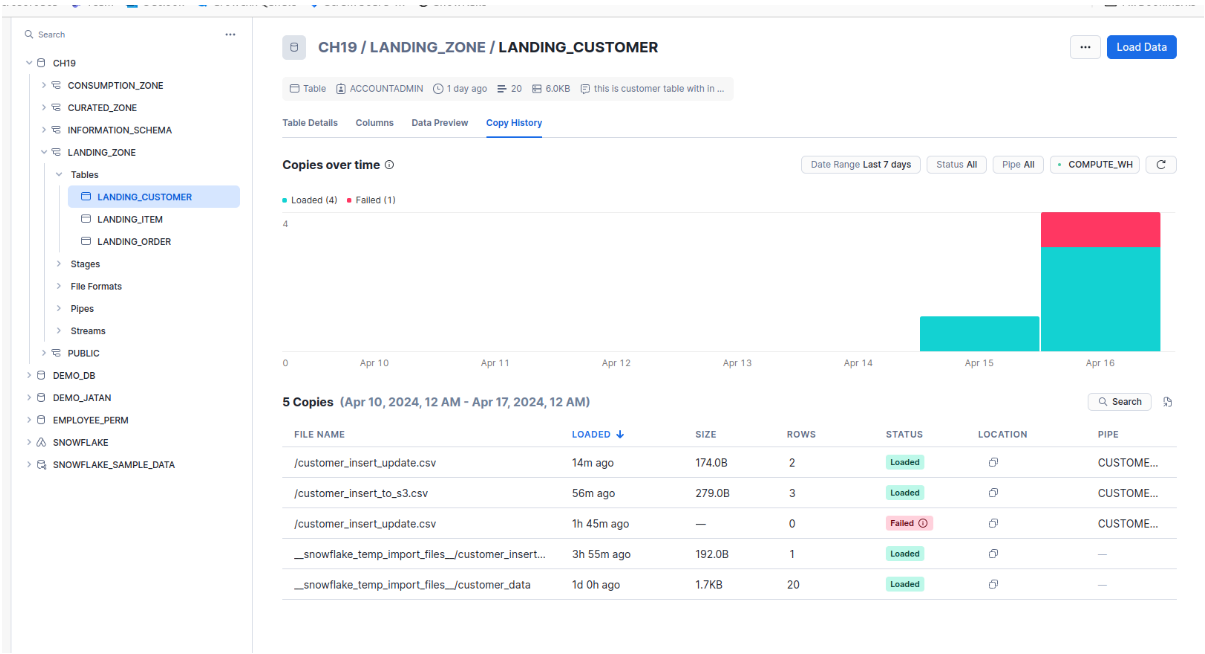
FILE NAME	LOADED ↓	SIZE	ROWS	STATUS	LOCATION	PIPE
/customer_insert_update.csv	26m ago	—	0	Failed ⓘ		CUSTOME...
__snowflake_temp_import_files__/customer_insert...	2h 36m ago	192.0B	1	Loaded		—
__snowflake_temp_import_files__/customer_data	22h 58m ago	1.7KB	20	Loaded		—

Error is related to the credential.

Solution – We have not created the role and given credential using storage integration.

Now do this.

Data is uploaded into S3



CH19 / LANDING_ZONE / LANDING_CUSTOMER

Table ACCOUNTADMIN 1 day ago 20 6.0KB this is customer table with in ...

Copies over time

5 Copies (Apr 10, 2024, 12 AM - Apr 17, 2024, 12 AM)

FILE NAME	LOADED ↓	SIZE	ROWS	STATUS	LOCATION	PIPE
/customer_insert_update.csv	14m ago	174.0B	2	Loaded		CUSTOME...
/customer_insert_to_s3.csv	56m ago	279.0B	3	Loaded		CUSTOME...
/customer_insert_update.csv	1h 45m ago	—	0	Failed ⓘ		CUSTOME...
__snowflake_temp_import_files__/customer_insert...	3h 55m ago	192.0B	1	Loaded		—
__snowflake_temp_import_files__/customer_data	1d 0h ago	1.7KB	20	Loaded		—

Inserted 2 rows in a Landed

-- Problem faced :

-- 1. S3 credential not provided in part 4 (storage integration).

-- 2. Scheduling task query error in part 5(in customer table).

-- 3. Conditional error in part 3 (getting 0 row after filtering in consumption zone order table because of less data, change filtering condition)

-- 4. Not set delimiter in file formating for csv file loading added skip header= 1;