EXPORTING DATA FROM SNOWFLAKE TO OPSVEDA

Setup Guide

12-April-2021

Background & Purpose

OpsVeda's operational intelligence platform provides users with intelligence to prevent operational disruptions and improve efficiency. It leverages transactional and logistics data, to assess the operational situation and prescribe counter measures where needed to the users.

OpsVeda supports data acquisition from a wide range of data sources in several formats. Near real-time acquisition through direct connection to the system of record as well as periodic transfers through files are supported.

Snowflake is a popular cloud-based data warehouse. OpsVeda customers who are storing the transaction data in Snowflake may choose to export the same from there to the OpsVeda platform periodically. This document explains, how customers can set up such data export from Snowflake.

Enterprise data is voluminous. So, it is desirable that only updated, deleted and new records are included in the periodic transfer from Snowflake to keep the volumes at manageable levels. The set up explained in this document is for setting up such incremental data transfer.

Data export set up steps

Snowflake supports the bulk data unloading (i.e. export) from database table into flat, delimited text files (CSV, TSV etc.). Snowflake can directly unload data to Amazon S3, Google cloud storage, and Microsoft Azure in different formats.

The tables with the data required for OpsVeda needs to be identified first. The steps to set up the incremental data export in Snowflake from such tables are outlined in the table below:

Step No.	Description	Relevant Snowflake Command/ Function	Description
1	Specify export file name and format	CREATE OR REPLACE FILE FORMAT	Copies the data from the database table into one or more files in external stage (Amazon S3)
2	**Change Data Capture (STREAM)**: Ensure that only changed, deleted or updated records is sent across.	CREATE STREAM	Records data manipulation language (DML) changes made to a table, including information about inserts, updates, and deletes.
3	**Create "STAGE" for Unloading table data to files**: The Create Stage wizard in the Snowflake web interface may be used. It automatically encloses field values in quotation characters	CREATE STAGE	Automatically encloses field values in quotation characters, as needed.

4	**Unload data:** Exports the data from the database table into one or more files in external stage (Amazon S3)	COPY INTO <location>	Copies the data from the database table into one or more files in external stage (Amazon S3)
5	**Set up job for automated export at specified intervals**:	CREATE TASK	To automate this process and schedule to unloading after specified intervals.

Code snippets & example

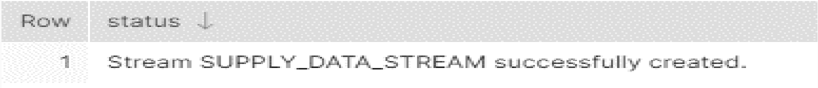
Code snippets for the steps explained above is given below:

Preparing to unload data: File format options specify the type of data contained in a file, as well as other related characteristics about the format of the data. To unload data to the OpsVeda, Inc. S3 bucket file format should be 'csv'. The following example creates a named CSV file format:

- Information: Database name = OV_DEMO_DATA, Schema name = PUBLIC, Table name = SUPPLY_DATA
- Create a stream on the SUPPLY_DATA table

```
CREATE STREAM SUPPLY_DATA_STREAM ON TABLE
"OV_DEMO_DATA"."PUBLIC"."SUPPLY_DATA";
```

Snowflake screenshot

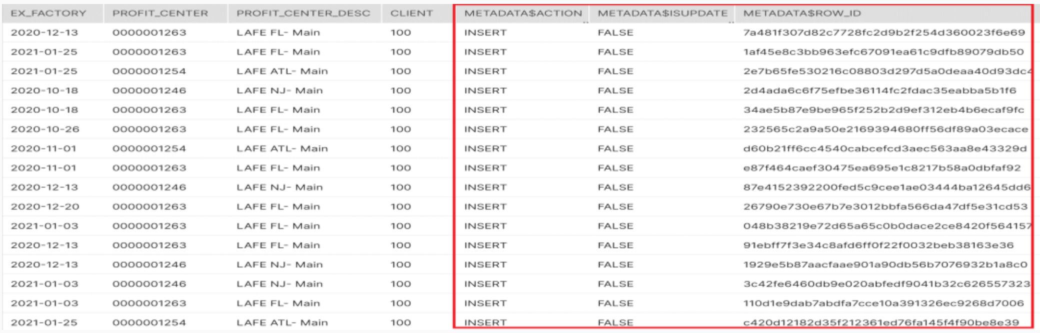


Row	status
1	Stream SUPPLY_DATA_STREAM successfully created.

Create STREAM: Checking data from SUPPLY_DATA_STREAM stream. Stream stores data in the same shape as source table with three additional columns: METADATA$ACTION, METADAATA$ISUPDATE, and METADATA$ROW_ID. Interpretation for the contents of these 3 columns can be found here.

```
SELECT * FROM SUPPLY_DATA_STREAM;
```

Snowflake screenshot

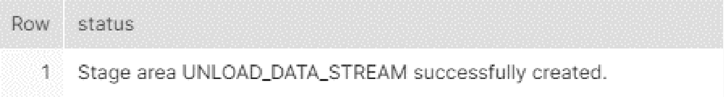


EX_FACTORY	PROFIT_CENTER	PROFIT_CENTER_DESC	CLIENT	METADATA$ACTION	METADATA$ISUPDATE	METADATA$ROW_ID
2020-12-13	0000001263	LAFE FL- Main	100	INSERT	FALSE	7a481f307d82c7728fc2d9b2f254d360023f6e69
2021-01-25	0000001263	LAFE FL- Main	100	INSERT	FALSE	1af45e8c3bb963efc67091ea61c9dfb89079db50
2021-01-25	0000001254	LAFE ATL- Main	100	INSERT	FALSE	2e7b65fe530216c08803d297d5a0deaa40d93dc4
2020-10-18	0000001246	LAFE NJ- Main	100	INSERT	FALSE	2d4ada6c6f75efbe36114fc2fdac35eabba5b1f6
2020-10-18	0000001263	LAFE FL- Main	100	INSERT	FALSE	34ae5b87e9be965f252b2d9ef312eb4b6ecaf9fc
2020-10-26	0000001263	LAFE FL- Main	100	INSERT	FALSE	232565c2a9a50e2169394680ff56df89a03ecace
2020-11-01	0000001254	LAFE ATL- Main	100	INSERT	FALSE	d60b21ff6cc4540cabcefcd3aec563aa8e43329d
2020-11-01	0000001263	LAFE FL- Main	100	INSERT	FALSE	e87f464caef30475ea695e1c8217b58a0dbfaf92
2020-12-13	0000001246	LAFE NJ- Main	100	INSERT	FALSE	87e4152392200fed5c9cee1ae03444ba12645dd6
2020-12-20	0000001263	LAFE FL- Main	100	INSERT	FALSE	26790e730e67b7e3012bbfa566da47df5e31cd53
2021-01-03	0000001263	LAFE FL- Main	100	INSERT	FALSE	048b38219e72d65a65c0b0dace2ce8420f564157
2020-12-13	0000001263	LAFE FL- Main	100	INSERT	FALSE	91ebff7f3e34c8afd6ff0f22f0032beb38163e36
2020-12-13	0000001246	LAFE NJ- Main	100	INSERT	FALSE	1929e5b87aacfaae901a90db56b7076932b1a8c0
2021-01-03	0000001246	LAFE NJ- Main	100	INSERT	FALSE	3c42fe6460db9e020abfedf9041b32c626557323
2021-01-03	0000001263	LAFE FL- Main	100	INSERT	FALSE	110d1e9dab7abdfa7cce10a391326ec9268d7006
2021-01-25	0000001254	LAFE ATL- Main	100	INSERT	FALSE	c420d12182d35f212361ed76fa145f4f90be8e39

Create external STAGE in S3 bucket: The code below creates STAGE in S3 bucket with access management permission for unloading data .

```
CREATE OR REPLACE STAGE UNLOAD_DATA_STREAM
  URL = 's3://------/'
  Credentials =( aws_key_id='------' aws_secret_key='---');
```

Snowflake screenshot



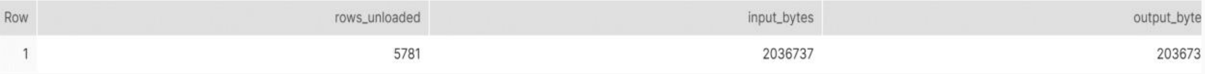
Row	status
1	Stage area UNLOAD_DATA_STREAM successfully created.

(i.e. the stage name is “UNLOAD_DATA_STREAM”)

Unloading data to S3 Bucket:

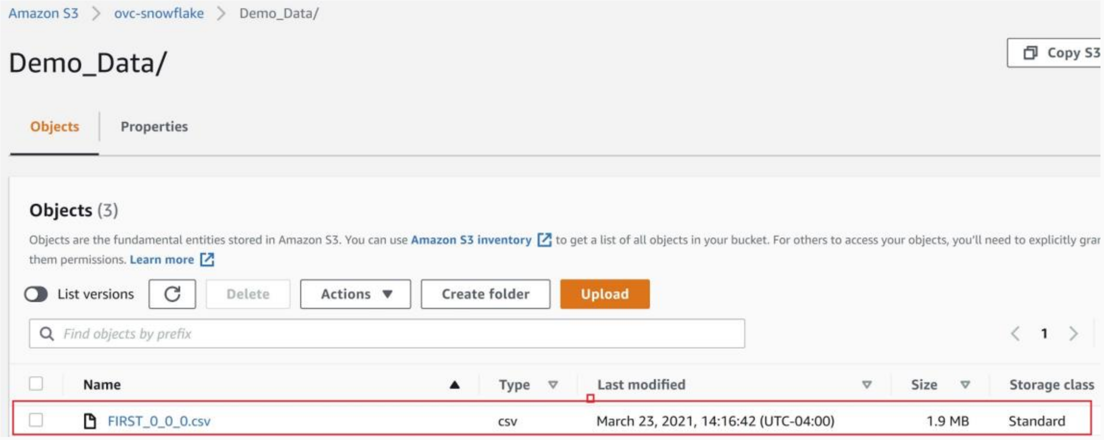
```
COPY INTO @ UNLOAD_DATA_STREAM/FIRST
    FROM SUPPLY_DATA_STREAM
    file_format = (type = CSV COMPRESSION = NONE)
    HEADER = TRUE
    OVERWRITE = TRUE;
```

Snowflake screenshot



Row	rows_unloaded	input_bytes	output_byte
1	5781	2036737	203673

S3: Bucket screenshot



Amazon S3 > ovc-snowflake > Demo_Data/

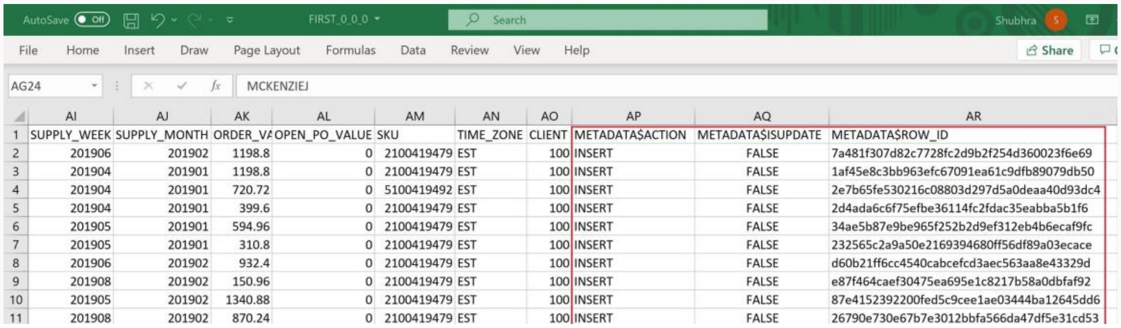
Demo_Data/

Objects (3)

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 grar them permissions. Learn more

□	Name	Type	Last modified	Size	Storage class
□	FIRST_0_0_0.csv	csv	March 23, 2021, 14:16:42 (UTC-04:00)	1.9 MB	Standard

CSV file screenshot

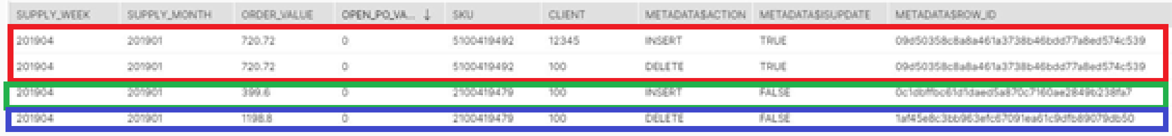


	AI	AJ	AK	AL	AM	AN	AO	AP	AQ	AR
1	SUPPLY_WEEK	SUPPLY_MONTH	ORDER_VA	OPEN_PO_VALUE	SKU	TIME_ZONE	CLIENT	METADATA$ACTION	METADATA$ISUPDATE	METADATA$ROW_ID
2	201906	201902	1198.8	0	2100419479	EST	100	INSERT	FALSE	7a481f307d82c7728fc2d9b2f254d360023f6e69
3	201904	201901	1198.8	0	2100419479	EST	100	INSERT	FALSE	1af45e8c3bb963efc67091ea61c9dfb89079db50
4	201904	201901	720.72	0	5100419492	EST	100	INSERT	FALSE	2e7b65fe530216c08803d297d5a0deaa40d93dc4
5	201904	201901	399.6	0	2100419479	EST	100	INSERT	FALSE	2d4ada6c6f75efbe36114fc2fdac35eabba5b1f6
6	201905	201901	594.96	0	2100419479	EST	100	INSERT	FALSE	34ae5b87e9be965f252b2d9ef312eb4b6ecaf9fc
7	201905	201901	310.8	0	2100419479	EST	100	INSERT	FALSE	232565c2a9a50e2169394680ff56df89a03ecace
8	201906	201902	932.4	0	2100419479	EST	100	INSERT	FALSE	d60b21ff6cc4540cabcefcd3aec563aa8e43329d
9	201908	201902	150.96	0	2100419479	EST	100	INSERT	FALSE	e87f464caef30475ea695e1c8217b58a0dbfaf92
10	201905	201902	1340.88	0	2100419479	EST	100	INSERT	FALSE	87e4152392200fed5c9cee1ae03444ba12645dd6
11	201908	201902	870.24	0	2100419479	EST	100	INSERT	FALSE	26790e730e67b7e3012bbfa566da47df5e31cd53

Verifying that it works: This step is not part of the regular set up. It is just to verify that the setup is working as expected.

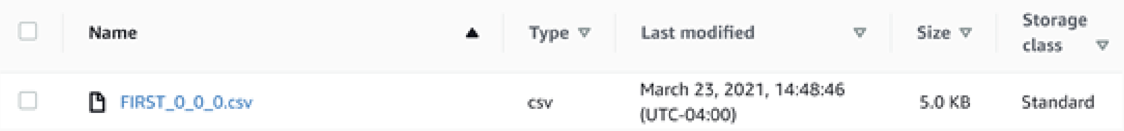
- Update 1 row, insert 1 row, Delete 1 row
- Checking in STREAM and CSV

Snowflake screenshot



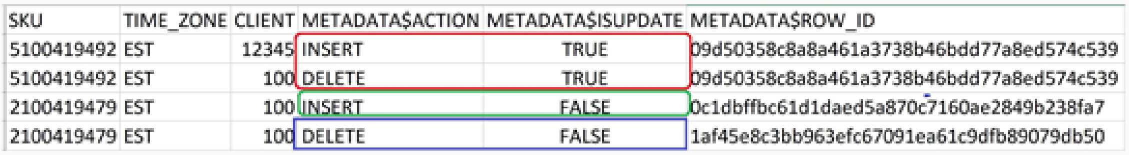
SUPPLY_WEEK	SUPPLY_MONTH	ORDER_VALUE	OPEN_PO_VA... ↓	SKU	CLIENT	METADATA$ACTION	METADATA$ISUPDATE	METADATA$ROW_ID
201904	201901	720.72	0	5100419492	12345	INSERT	TRUE	09d50358c8a8a461a3738b46bdd77a8ed574c539
201904	201901	720.72	0	5100419492	100	DELETE	TRUE	09d50358c8a8a461a3738b46bdd77a8ed574c539
201904	201901	399.6	0	2100419479	100	INSERT	FALSE	0c1dbffbc61d1daed5a870c7160ae2849b238fa7
201904	201901	1198.8	0	2100419479	100	DELETE	FALSE	1af45e8c3bb963efc67091ea61c9dfb89079db50

S3: Bucket screenshot



□	Name ▲	Type ▽	Last modified ▽	Size ▽	Storage class ▽
□	FIRST_0_0_0.csv	csv	March 23, 2021, 14:48:46 (UTC-04:00)	5.0 KB	Standard

CSV file screenshot



SKU	TIME_ZONE	CLIENT	METADATA$ACTION	METADATA$ISUPDATE	METADATA$ROW_ID
5100419492	EST	12345	INSERT	TRUE	09d50358c8a8a461a3738b46bdd77a8ed574c539
5100419492	EST	100	DELETE	TRUE	09d50358c8a8a461a3738b46bdd77a8ed574c539
2100419479	EST	100	INSERT	FALSE	0c1dbffbc61d1daed5a870c7160ae2849b238fa7
2100419479	EST	100	DELETE	FALSE	1af45e8c3bb963efc67091ea61c9dfb89079db50

- UPDATE : METADATA$ACTION = DELETE, METADATA$ISUPDATE = TRUE (OLD DATA)
  METADATA$ACTION = INSERT, METADATA$ISUPDATE = TRUE (NEW DATA)
- INSERT : METADATA$ACTION = INSERT, METADATA$ISUPDATE = FALSE
- DELETE : METADATA$ACTION = DELETE, METADATA$ISUPDATE = FALSE

Set up periodic transfer: This step will ensure that files with incremental data is sent across at the desired cadence (every hour in the example below).

```
           Create a stored procedure that unloads data from a table

CREATE OR REPLACE PROCEDURE UNLOAD_DATA_SP()
    RETURN STRING NOT NULL
    LANGUAGE JAVASCRIPT
    AS
    $$
      var sql_command = “”

      var sql_command = sql_command. concat("COPY INTO
      @UNLOAD_DATA_STREAM/FIRST ","/",Date.now(),"/","
      from SUPPLY_DATA_STREAM overwrite=true;");

      var statement1 = snowflake.createStatement({sqlText: sql_command});
      var result_set1 = statement1.execute()
    return sql_command;
    $$,


           Create a task that calls the stored procedure every hour

CREATE OR REPLACE TASK UNLOAD_DATA_TASK
    WAREHOUSE = -------
    SCHEDULE = ’60 minute’
AS
    CALL UNLOAD_DATA_SP();
```