Snowflake Ingestion, Extraction Techniques and Best Practices

Version 2.0

June 22, 2020

Sravan Reddy Mandapati

Mohan Nutakki

Badri Singh

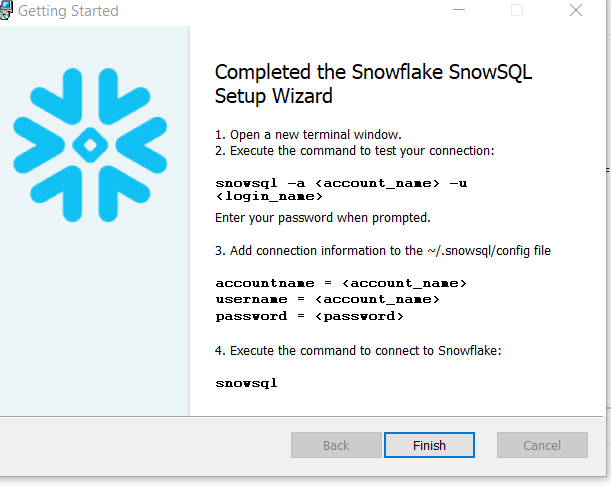
**Objective:** To demonstrate ingestion and extraction process using SnowSQL and DataStage.

**Note:** For this POC, X-Small Datawarehouse with 1 cluster (CDW) was created, stats are collected with 1 million records of 45 column dataset, all commands ran through windows command prompt.

**Connectivity using SnowSQL:**

Download and install SnowSQL from Snowflake

Should see following screen after installation in windows



Test connectivity to snowflake using SnowSQL command

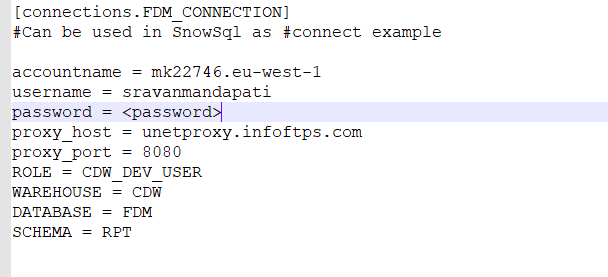
**Command**: snowsql -a <account name>.<region> -u <login> --proxy-host unetproxy.infoftps.com –proxy-port 8080

Provide password when prompted



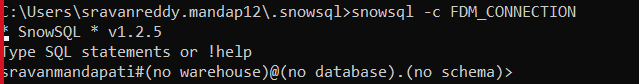
To automate the connection login, go to SnowSQL config file and provide connection settings

Config file:



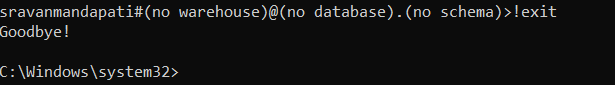
Test connection using following command

**Command**: snowsql -c <connection name>



To exit SnowSQL run following command

**Command**: !exit or !disconnect



**Ingesting Data using SnowSQL:**

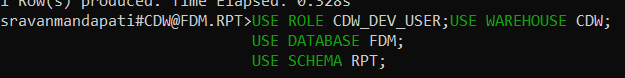
**Step 1**: After login to SnowSQL connect to your data warehouse, database and schema

**Command**: USE ROLE <role>;

USE WAREHOUSE <datawarehouse>;

USE DATABASE <database>;

USE SCHEMA <schema>;

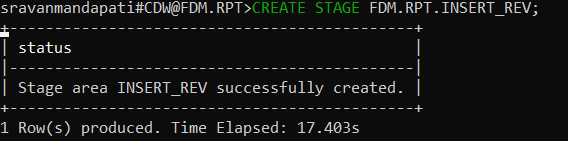


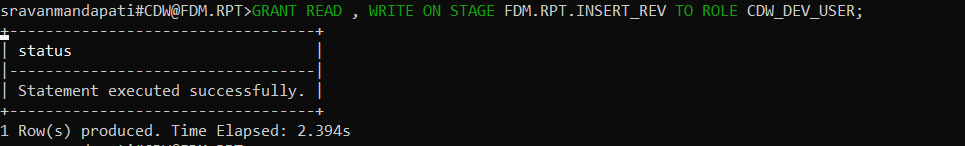
Make sure ROLE has access to warehouse, database and schema

**Step 2:** create stage using following command

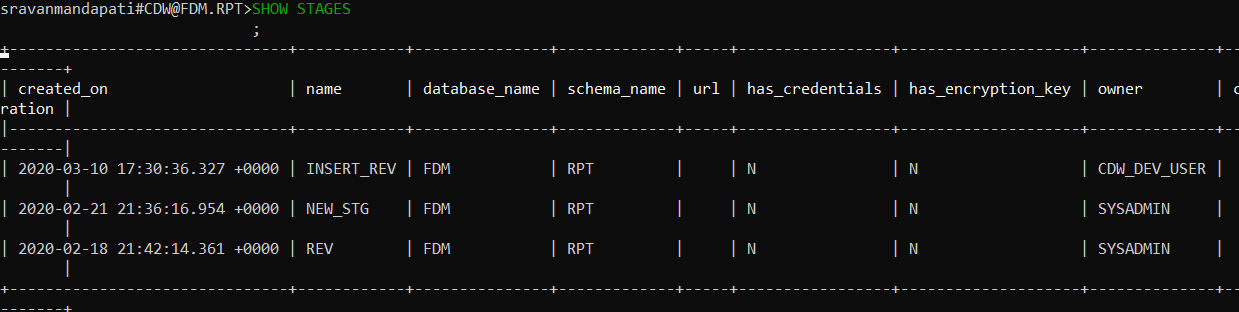
**Command**: CREATE STAGE <DATABASE>.<SCHAMA>.<STAGE NAME>;

GRANT READ, WRITE ON STAGE <DATABASE>.<SCHAMA>.<STAGE NAME> TO ROLE <ROLE>;





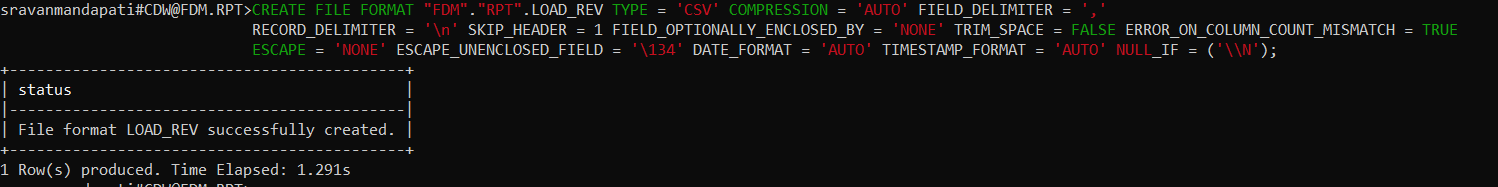
Should see after running show stage command

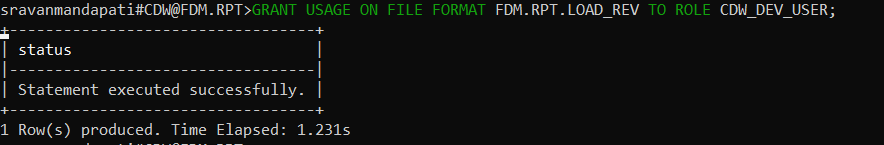


**Step 3**: create file format using following command

**Command**: CREATE FILE FORMAT <DATABASE>.<SCHAMA>.<FILE FORMAT NAME>; SET COMPRESSION = 'AUTO' FIELD\_DELIMITER = '<DELIMETER>' RECORD\_DELIMITER = '<RECORD DELIMETER>' SKIP\_HEADER = <LINES TO SKIP> FIELD\_OPTIONALLY\_ENCLOSED\_BY = 'NONE' TRIM\_SPACE = FALSE ERROR\_ON\_COLUMN\_COUNT\_MISMATCH = TRUE ESCAPE = 'NONE' ESCAPE\_UNENCLOSED\_FIELD = '\134' DATE\_FORMAT = '<DATE FORMAT>' TIMESTAMP\_FORMAT = 'TIMESTAMP FORMAT' NULL\_IF = ('\\N');

GRANT USAGE ON FILE FORMAT <DATABASE>.<SCHAMA>.<FILE FORMAT NAME> TO ROLE <ROLE>

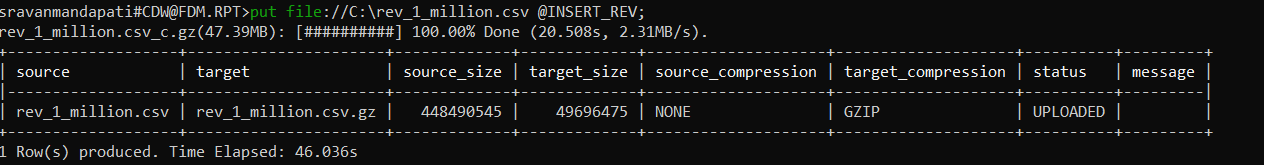




Make sure date and timestamp format matches to dataset format

**Step 4**: Load dataset to stage using following command

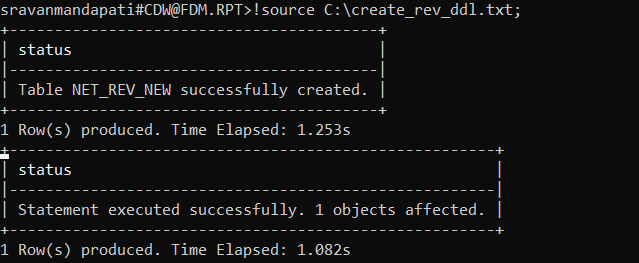
**Command**: PUT FILE:<FILE PATH / FILE NAME. EXTENSION> @<STAGE NAME>



Observations: Compressed file size to 10%,and took 46 seconds to load data into stage

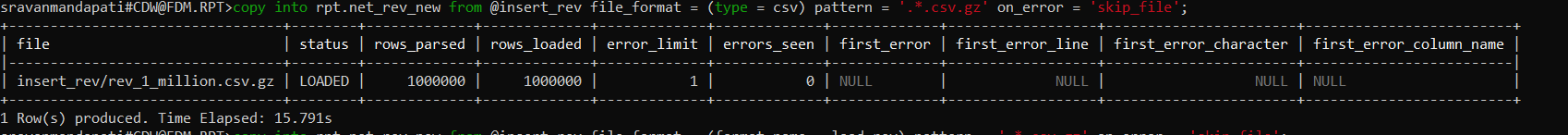
**Step 5**: create table and provide grants to role using following command

**Command**: !SOURCE <DDL FILE>



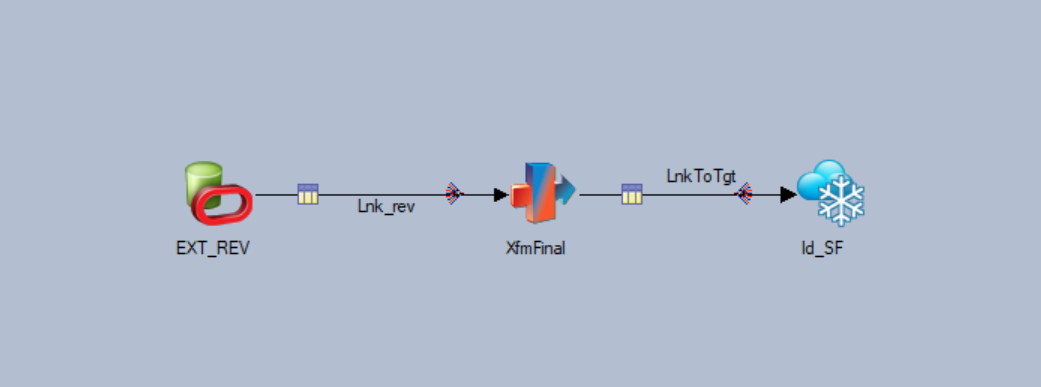
**Step 6**: ingest data into target table using stage and file formats created in above steps

**Command**: COPY INTO <TARGET TABLE> FROM @<STAGE> FILE\_FORMAT = (FORMAT\_NAME = ‘<FORMAT NAME>’) PATTERN = '<FILE PATTERN IN STAGE>' ON\_ERROR = 'SKIP\_FILE';



Observations: loaded 1 million records and took 15 seconds.

**Ingesting Data using DataStage:**



This job loaded 10 million records in 13 min with 45 columns.

Set following option in Snowflake connector

**url**: jdbc:snowflake://worldpay.us-east-1.snowflakecomputing.com/?warehouse=DEV02\_US\_RADAR\_CDM\_WAREHOUSE&role=DEV02\_US\_RADAR\_CDM\_TRANSFORM

**user name**: SERVICE\_OTHER\_US\_RADAR\_CDM

**password**: \*\*\*\*\*\*\*\*

**table name**: NET\_REV

**table action**: Append

**Field Delimiter:** |

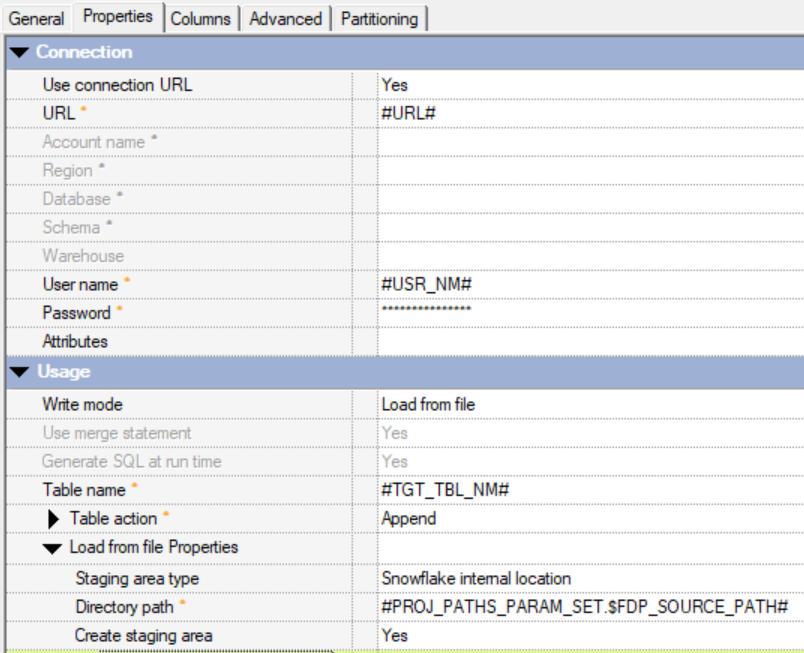
**Record Delimiter:** <NL>

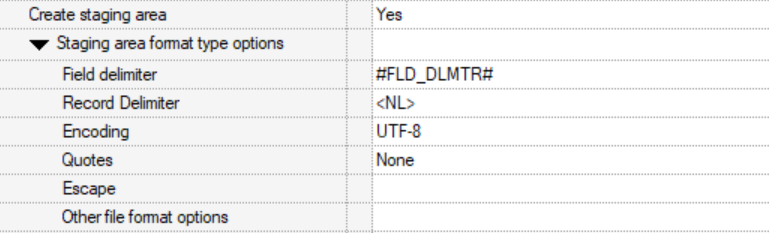
**Encoding:** UTF-8

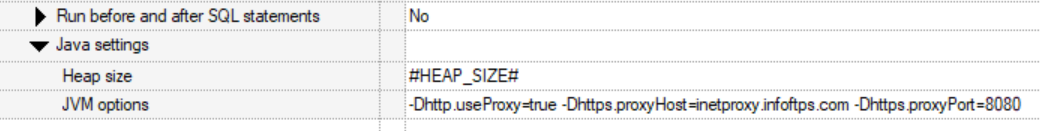
**Quotes:** None

**heap size**: 256

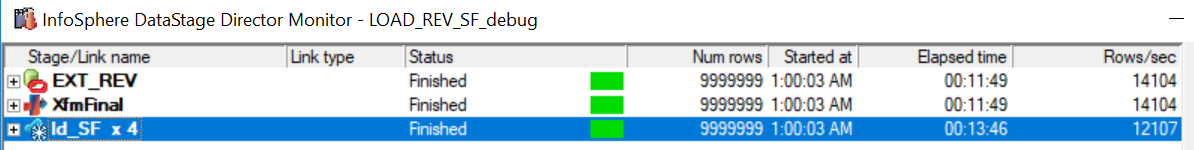
**jvm options**: <Dhttp.useProxy=true -Dhttps.proxyHost=inetproxy.infoftps.com -Dhttps.proxyPort=8080>

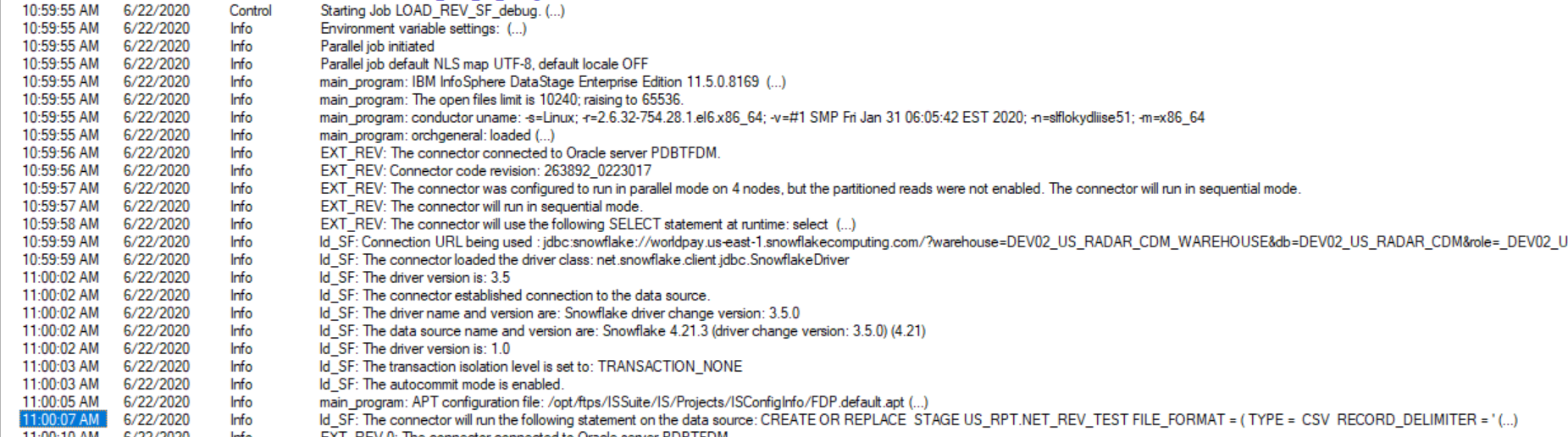


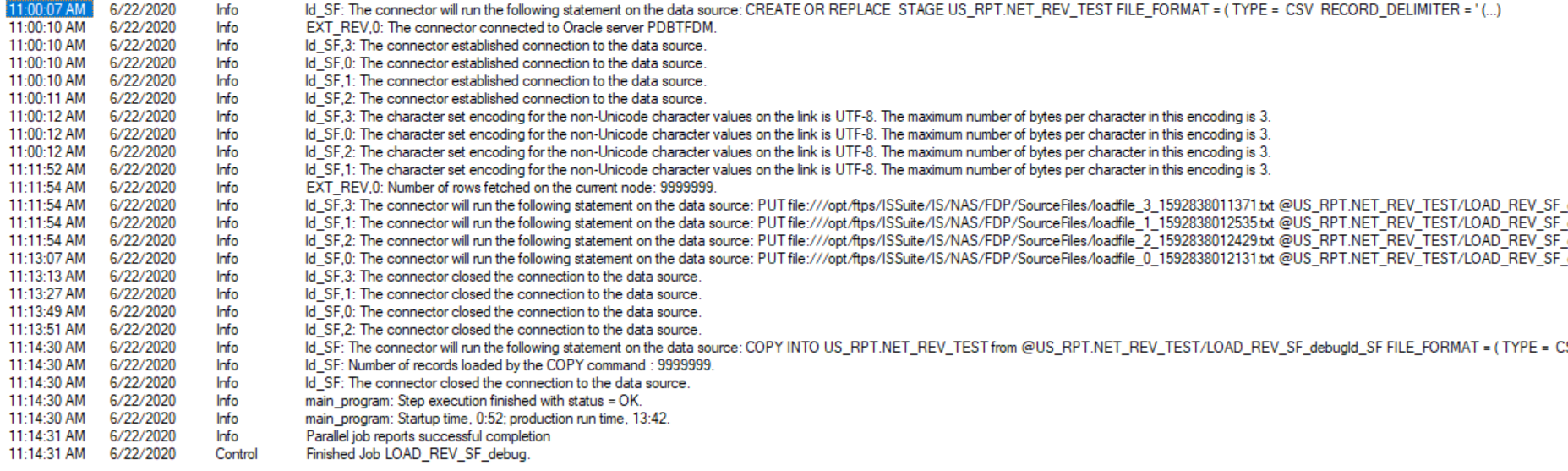




After compiling and running the job, below screenshot shows how much time it took to load.







In the above log, if you see the whole process takes 13 min. That includes below.

Initial extraction of data into intermediate files – 11 min

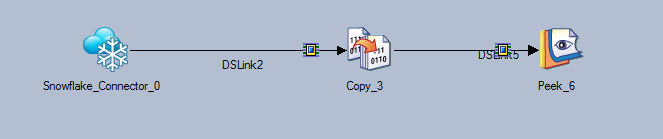
PUT command running on extracted data – around 2 min

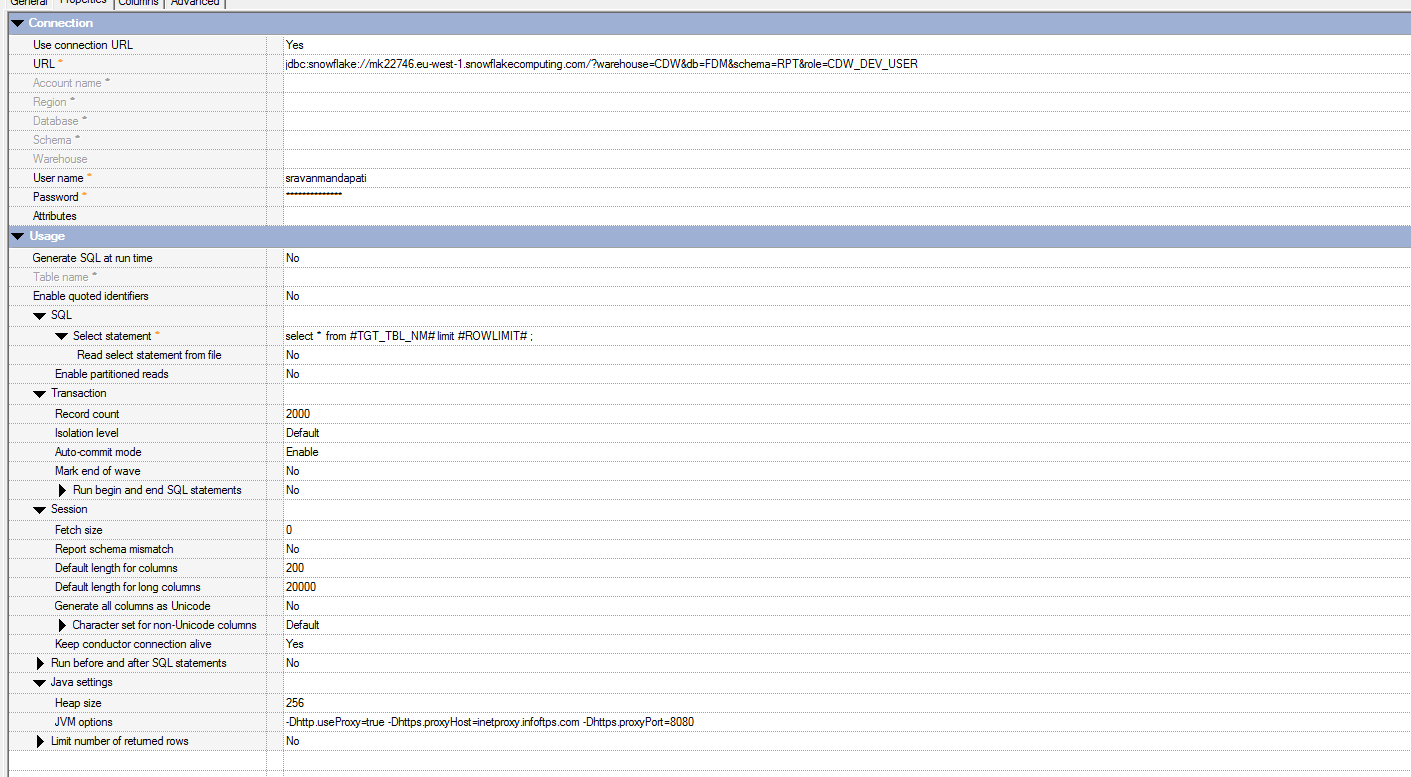
COPY command running on staged data – around 1 min

Actual snowflake cost involves only while doing PUT and COPY commands. Before that, it is data prep for data loads.

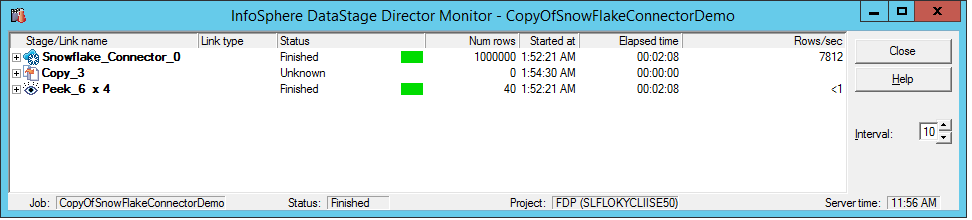
**Extracting Data from DataStage:**

Use Snowflake connector to extract data and use exact setting of ingestion except providing sql statement





Compile and run the job



Observations: extract 1 million records took 110 seconds (1.8 minutes).

**Extracting Data from SnowSQL:**

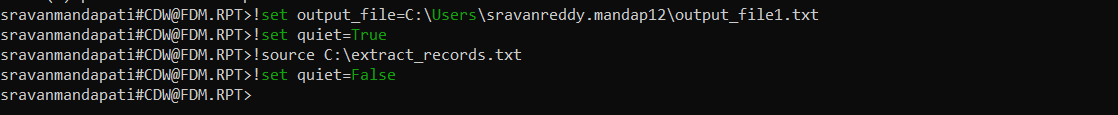
Run following commands to extract and load into target file

**Commands**: !set output\_file=<target file name>

!set quiet=True

!source <sql script>

!set quite=False



Observations: extracted 1 million records in 1,416 seconds (23.6 minutes).

**Pros and Cons:**

|  |  |  |
| --- | --- | --- |
|  | **Pros** | **Cons** |
| **SnowSQL** | * Loading is quick compared to other options * Snowflake native support * Compression size reduced * Reusable stages and file formats | * Requires user training * Extraction to file is slow compared to DataStage * Data needed to load to files first before loading to stages * Manual errors because of scripting * Additional encryption required |
| **DataStage** | * Extracting data fast compared to SnowSQL * Ingestion using datastage take similar time as SnowSQL, but no need of intermediate files creation, which datastage handles itself * No need to maintain files * No user training required * Connects to all sources | * Can not reuse stages |

**Recommendation**

TBD?