

Steven Evans

SRE220000

BUAN 6335

Snowflake Lab 4

4.1.4 Creating the Region Table

The screenshot shows the Snowflake UI interface. On the left, the sidebar displays various databases and schemas, with the 'REGION' table under the 'CHIPMUNK_DB.PUBLIC' schema selected. The main area shows a query editor with the following SQL code:

```
CREATE WAREHOUSE IF NOT EXISTS CHIPMUNK_WH;
USE WAREHOUSE CHIPMUNK_WH;
CREATE DATABASE IF NOT EXISTS CHIPMUNK_DB;
USE CHIPMUNK_DB.PUBLIC;

-- 4.1.4 Create a REGION table. This table will be loaded from a source file:
CREATE OR REPLACE TABLE REGION (
    R_REGIONKEY NUMBER(38,0) NOT NULL,
    R_NAME     VARCHAR(25) NOT NULL,
    R_COMMENT  VARCHAR(152)
);

-- 4.1.5 Create a file format called MYPIPEFORMAT, that will read the pipe-
-- delimited region.tbl file:
CREATE OR REPLACE FILE FORMAT MYPIPEFORMAT
    TYPE = CSV
    COMPRESSION = NONE
    FIELD_DELIMITER = '|'
    FILE_EXTENSION = 'tbl'
    ERROR_ON_COLUMN_COUNT_MISMATCH = FALSE;
```

The results pane shows a single row of data: "Table REGION successfully created." Below the results, the "Query Details" section indicates a query duration of 123ms.

4.1.5-6 Creating the pipe formats to read the GZIP and pipe delimited file

The screenshot shows the Snowflake UI interface. The sidebar shows the same database structure as the previous screenshot. The main area shows a query editor with the following SQL code:

```
-- 4.1.5 Create a file format called MYPIPEFORMAT, that will read the pipe-
-- delimited region.tbl file:
CREATE OR REPLACE FILE FORMAT MYPIPEFORMAT
    TYPE = CSV
    COMPRESSION = NONE
    FIELD_DELIMITER = '|'
    FILE_EXTENSION = 'tbl'
    ERROR_ON_COLUMN_COUNT_MISMATCH = FALSE;

-- 4.1.6 Create a file format called MYGZIPPIPEFORMAT that will read the
-- compressed version of the region.tbl file. It should be identical to
-- the MYPIPEFORMAT, except you will set COMPRESSION = GZIP:
CREATE OR REPLACE FILE FORMAT MYGZIPPIPEFORMAT
    TYPE = CSV
    COMPRESSION = GZIP
    FIELD_DELIMITER = '|'
    FILE_EXTENSION = 'tbl'
    ERROR_ON_COLUMN_COUNT_MISMATCH = FALSE;

-- 4.2.0 Load the region.tbl file
-- The files for this task have been pre-loaded into a location on AWS.
```

The results pane shows a single row of data: "File format MYGZIPPIPEFORMAT successfully created." Below the results, the "Query Details" section indicates a query duration of 127ms.

4.2.2 Reviewing the stage properties

The screenshot shows the AWS Glue Data Catalog interface. On the left, the sidebar lists various databases and tables. In the main area, a query is running in a worksheet titled 'CHIPMUNK_DB.PUBLIC'. The code is as follows:

```

82 -- 4.2.1 Review the properties of the stage:
83 -- Take note of the field delimiter for the file format in the stage.
84
85
86 | DESCRIBE STAGE TRAINING_DB.TRAININGLAB.ED_STAGE;
87
88
89 -- 4.2.2 Review the properties of the file:
90 -- Take note of the delimiter used in the file.
91
92 SELECT n.$1
93 FROM @training_db.traininglab.ed_stage/load/lab_files/region.tbl n;
94
95 -- NOTE: As you may have noticed, the field delimiter for the file
96 -- format defined in the stage is set to a comma, not a pipe. However,
97 -- the file data is in pipe format. You have two options to load the
98 -- file despite this discrepancy. First, you can override the stage's
99 -- file format by specifying a different file format to be used when the
100 -- COPY command executes. Second, you can load the data in pipe format
101 -- even though the stage uses a comma as a field delimiter. In this
102 -- exercise, you will override the stage's file format by configuring
103 -- the COPY statement to use the MYPIPEFORMAT file format instead when
104 -- it executes.
105
106 -- 4.2.3 Confirm the region.tbl file is in the external stage with the list

```

The results pane shows a table of stage properties:

parent_property	property	property_type	property_value	property_default
1	STAGE_FILE_FORMAT	TYPE	String	CSV
2	STAGE_FILE_FORMAT	RECORD_DELIMITER	String	\n
3	STAGE_FILE_FORMAT	FIELD_DELIMITER	String	,
4	STAGE_FILE_FORMAT	FILE_EXTENSION	String	
5	STAGE_FILE_FORMAT	SKIP_HEADER	Integer	0
6	STAGE_FILE_FORMAT	PARSE_HEADER	Boolean	false
7	STAGE_FILE_FORMAT	DATE_FORMAT	String	AUTO
8	STAGE_FILE_FORMAT	TIME_FORMAT	String	AUTO
9	STAGE_FILE_FORMAT	TIMESTAMP_FORMAT	String	AUTO

On the right, the 'Query Details' panel shows the following information:

- Query duration: 46ms
- Rows: 33
- Query ID: 01b89ed5-0004-3f59-0...

4.2.2 Reviewing the file properties

The screenshot shows the AWS Glue Data Catalog interface. On the left, the sidebar lists various databases and tables. In the main area, a query is running in a worksheet titled 'CHIPMUNK_DB.PUBLIC'. The code is as follows:

```

90 -- Take note of the delimiter used in the file.
91
92
93 SELECT n.$1
94 FROM @training_db.traininglab.ed_stage/load/lab_files/region.tbl n;
95
96 -- NOTE: As you may have noticed, the field delimiter for the file
97 -- format defined in the stage is set to a comma, not a pipe. However,
98 -- the file data is in pipe format. You have two options to load the
99 -- file despite this discrepancy. First, you can override the stage's
100 -- file format by specifying a different file format to be used when the
101 -- COPY command executes. Second, you can load the data in pipe format
102 -- even though the stage uses a comma as a field delimiter. In this
103 -- exercise, you will override the stage's file format by configuring
104 -- the COPY statement to use the MYPIPEFORMAT file format instead when
105 -- it executes.
106 -- 4.2.3 Confirm the region.tbl file is in the external stage with the list

```

The results pane shows a table of file properties:

\$1
0 AFRICA ar deposits. blithely final packages cajole. regular waters are final requests. regular accounts are acc
1 AMERICA hs use ironic
2 ASIA ges. thinly even pinto beans ca
3 EUROPE ly final courts cajole furiously final excuse
4 MIDDLE EAST quickly special accounts cajole carefully blithely close requests. carefully final asymptotes hagg
6 AFRICA ar deposits. blithely final packages cajole. regular waters are final requests. regular accounts are acc
7 AMERICA hs use ironic
8 ASIA ges. thinly even pinto beans ca
9 EUROPE lv final courts caiole furiously final excuse

On the right, the 'Query Details' panel shows the following information:

- Query duration: 502ms
- Rows: 10
- Query ID: 01b89ed6-0004-3f59-0...

Important Noticing: The file is in pipe format with pipes separating the column values.

4.2.4 Loading the data from the external stage into the region table

The screenshot shows the Snowflake interface with the following details:

- Databases:** CHIPMUNK_DB.PUBLIC
- Tables:** PUBLIC.REGION
- Code:**

```

108 LIST @training_db.traininglab.ed_stage/load/lab_files/ pattern='.*region.*';
109
110
111
112 -- 4.2.4 Load the data from the external stage to the REGION table using the
113 -- file format you created in the previous task:
114
115 COPY INTO REGION
116 FROM @training_db.traininglab.ed_stage/load/lab_files/
117 FILES = ('region.tbl')
118 FILE_FORMAT = (FORMAT_NAME = MYPIPEFORMAT);
119
120
121 -- 4.2.5 Select and review the data in the REGION table, either by executing
122 -- the following command in your worksheet or by using Preview Data in
123 -- the sidebar:
124
125 SELECT * FROM REGION;

```
- Results:** A table showing the results of the COPY command.

file	status	rows_parsed	rows_loaded	error_limit	errors
s3://snowflakeed/load/lab_files/region.tbl	LOADED	5	5	1	

- Query Details:**
 - Query duration: 779ms
 - Rows: 1
 - Query ID: 01b89eda-0004-3f5c-0...
- File Progress:** 100% filled
- Ask Copilot:** A button to interact with AI.

4.2.5 SReviewing the data loaded into the region table

The screenshot shows the Snowflake interface with the following details:

- Databases:** CHIPMUNK_DB.PUBLIC
- Tables:** PUBLIC.REGION
- Code:**

```

118 FILE_FORMAT = (FORMAT_NAME = MYPIPEFORMAT);
119
120
121 -- 4.2.5 Select and review the data in the REGION table, either by executing
122 -- the following command in your worksheet or by using Preview Data in
123 -- the sidebar:
124
125 SELECT * FROM REGION;

```
- Results:** A table showing the data in the REGION table.

R_REGIONKEY	R_NAME	R_COMMENT
1	AFRICA	lar deposits. blithely final packages cajole. regular waters are final requests.
2	AMERICA	hs use ironic, even requests. s
3	ASIA	ges. thinly even pinto beans ca
4	EUROPE	ly final courts cajole furiously final excuse
5	MIDDLE EAST	uickly special accounts cajole carefully blithely close requests. carefully fina

- Query Details:**
 - Query duration: 167ms
 - Rows: 5
 - Query ID: 01b89eda-0004-3f59-0...
- Ask Copilot:** A button to interact with AI.

4.3.1 Emptying the region table to reload via the GZIPPIPE format

The screenshot shows the Snowflake UI interface. On the left, the navigation pane displays various databases and schemas, including CHIPMUNK_DB.PUBLIC. In the main workspace, a query is being run:

```

126
127
128 -- 4.3.0 Loading a GZip Compressed File on an External Stage Into a Table
129 -- The scenario for this activity is fundamentally the same as the
130 -- previous activity. The difference is that you will load the REGION
131 -- table from a gzip compressed file that is in the external stage. You
132 -- will use the MYGZIPPIPEFORMAT file format you created in the previous
133 -- part of this lab.
134
135 -- 4.3.1 Empty the REGION table in the PUBLIC schema of CHIPMUNK_DB:
136
137 TRUNCATE TABLE region;
138
139
140 -- 4.3.2 Confirm that the region.tbl.gz file is in the external stage:
141
142 LIST @training_db.trainingLab.ed_stage/load/lab_files/ pattern='.*region.*';

```

The results pane shows a single row:

status
1 Statement executed successfully.

Query Details:

- Query duration: 246ms
- Rows: 1
- Query ID: 01b89edb-0004-3f5c-0...

4.3.2 Confirming the region.tbl.gz file is in the stage

The screenshot shows the Snowflake UI interface. On the left, the navigation pane displays various databases and schemas, including CHIPMUNK_DB.PUBLIC. In the main workspace, a query is being run:

```

140 -- 4.3.2 Confirm that the region.tbl.gz file is in the external stage:
141
142 LIST @training_db.trainingLab.ed_stage/load/lab_files/ pattern='.*region.*';
143
144
145 -- 4.3.3 Reload the REGION table from the region.tbl.gz file. Review the
146 -- syntax of the COPY INTO command used in the previous task. Specify
147 -- the file to COPY as region.tbl.gz.
148
149 COPY INTO region
150   FROM @training_db.trainingLab.ed_stage/load/lab_files/
151   FILES = ('region.tbl.gz')
152   FILE_FORMAT = ( FORMAT_NAME = MYGZIPPIPEFORMAT );
153
154
155 -- 4.3.4 Query the table to confirm the data was successfully loaded:
156

```

The results pane shows a table of file details:

name	size	md5
s3://snowflakeed/load/lab_files/region.json	589	9486cea85693b6b4d05
s3://snowflakeed/load/lab_files/region.parquet	1367	a8a34dbaf7394c2e5803
s3://snowflakeed/load/lab_files/region.tbl	389	c235841b00d29ad4f817
s3://snowflakeed/load/lab_files/region.tbl.gz	267	306f9031adb2140c0f96
s3://snowflakeed/load/lab_files/region/data_0_0_0.tbl	768	1206853fa4c104225725
s3://snowflakeed/load/lab_files/region1	9899309	e1114cced448ee3da3c9
s3://snowflakeed/load/lab_files/region_bad_1.json	617	5818da27506248fb16a
s3://snowflakeed/load/lab_files/region_bad_1.tbl	389	89802771ca308e81eb6t
s3://snowflakeed/load/lab_files/region_bad_2.tbl	389	56a34ebf71f3c904a1d8

Query Details:

- Query duration: 130ms
- Rows: 20
- Query ID: 01b89edc-0004-3f5c-0...

4.3.3 Loading the region data into the region table from the external stage via GZIP

```

139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
-- 4.3.2 Confirm that the region.tbl.gz file is in the external stage:
LIST @training_db.traininglab.ed_stage/load/lab_files/ pattern='.*region.*';

-- 4.3.3 Reload the REGION table from the region.tbl.gz file. Review the
-- syntax of the COPY INTO command used in the previous task. Specify
-- the file to COPY as region.tbl.gz.

COPY INTO region
FROM @training_db.traininglab.ed_stage/load/lab_files/
FILEFORMAT = ('region.tbl.gz')
FILE_FORMAT = ( FORMAT_NAME = MYZIPPIEFORMAT);

-- 4.3.4 Query the table to confirm the data was successfully loaded:

```

file	status	rows_parsed	rows_loaded	error_limit	erro
s3://snowflakeed/load/lab_files/region.tbl.gz	LOADED	5	5	1	

Query Details

- Query duration: 659ms
- Rows: 1
- Query ID: 01b89edc-0004-3f59-0...

Show more ▾

file
100% filled

Ask Copilot

4.3.4 Querying the table to ensure data was loaded correctly

```

152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
FILE_FORMAT = ( FORMAT_NAME = MYZIPPIEFORMAT);

-- 4.3.4 Query the table to confirm the data was successfully loaded:
SELECT * FROM region;

-- 4.3.5 Resize and suspend the virtual warehouse.
ALTER WAREHOUSE CHIPMUNK_WH SET WAREHOUSE_SIZE=XSmall;
ALTER WAREHOUSE CHIPMUNK_WH SUSPEND;

-- 4.4.0 Validating Data Prior to Load
-- Now we're going to practice using the VALIDATION_MODE parameter of
-- the COPY INTO statement to check for problems with the file prior to

```

R_REGIONKEY	R_NAME	R_COMMENT
1	AFRICA	lar deposits. blithely final packages cajole. regular waters are final requests.
2	AMERICA	hs use ironic, even requests. s
3	ASIA	ges. thinly even pinto beans ca
4	EUROPE	ly final courts cajole furiously final excuse
5	MIDDLE EAST	uickly special accounts cajole carefully blithely close requests. carefully fina

Query Details

- Query duration: 123ms
- Rows: 5
- Query ID: 01b89edd-0004-3f59-0...

Show more ▾

R_REGIONKEY
0

Ask Copilot

4.4.1 Modifying the NATION table adding in the designated columns

The screenshot shows a database interface with a sidebar containing a tree view of databases like ADMIN, BADGER_DB, BEETLE_DB, etc., and a schema tree under CHIPMUNK_DB showing PUBLIC, INFORMATION_SCHEMA, and MY_SCHEMA. The main area displays a SQL worksheet titled 'CHIPMUNK_DB.PUBLIC' with the following code:

```
169 -- Loading it.
170 -- NOTE: Some of the steps will intentionally produce errors. Read and
171 -- follow the instructions closely, so you understand what is happening
172 -- in each step.
173
174 -- 4.4.1 Modify the NATION table by running the statement below.
175
176
177 CREATE OR REPLACE TABLE nation (
178     NATION_KEY INTEGER
179     , NATION VARCHAR
180     , REGION_KEY INTEGER
181     , COMMENTS VARCHAR
182 );
183
184
185
```

Below the code, there is a 'Results' tab showing a table named 'status' with one row: 'Table NATION successfully created.' To the right, the 'Query Details' pane shows a duration of 218ms, 1 row, and a query ID of 01b89ede-0004-3f59-0... There is also an 'Ask Copilot' button.

4.4.2 Loading the data into the nation table checking any errors when loading

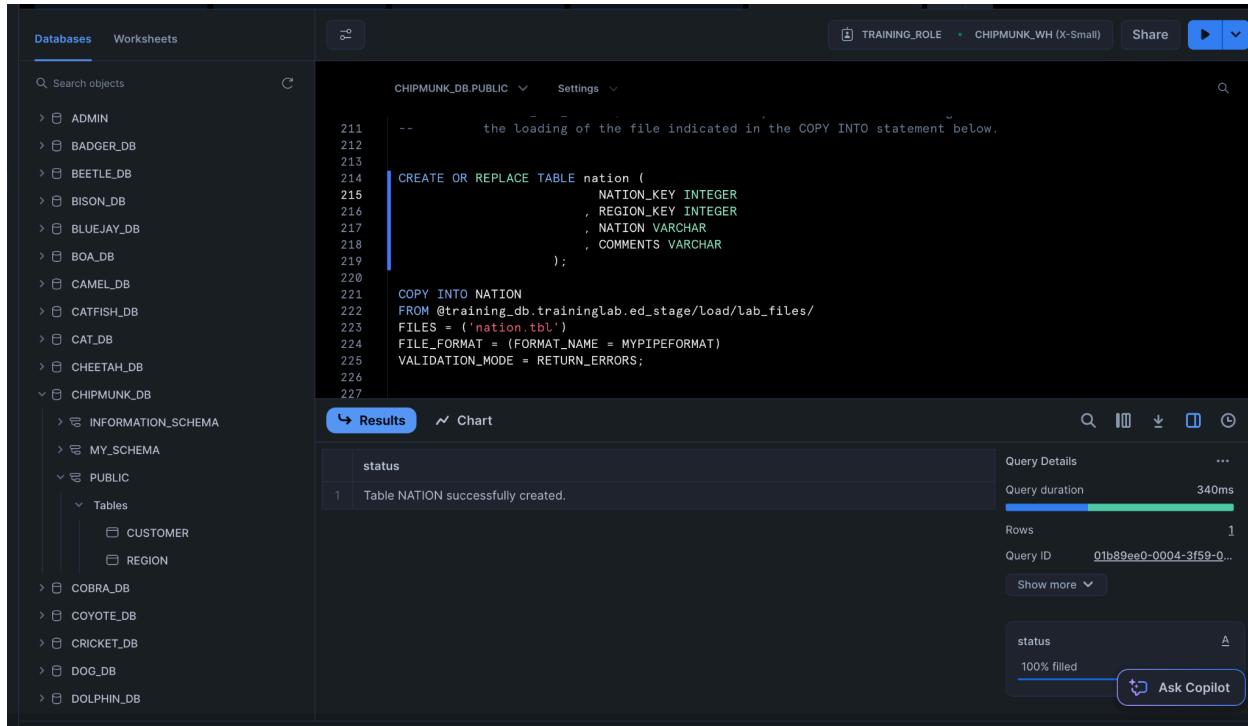
The screenshot shows the same database interface as the previous one. The main area displays a SQL worksheet with the following code:

```
194 COPY INTO NATION
195 FROM @training_db.traininglab.ed_stage/load/lab_files/
196 FILES = ('nation.tbl')
197 FILE_FORMAT = (FORMAT_NAME = MYPipeFormat)
198 VALIDATION_MODE = RETURN_ALL_ERRORS;
199
200
201 -- You should have gotten a message that says, Query produced no
202 -- results. This means there were no errors and that you can load the
203 -- table. But now we're going to recreate the table and switch the order
204 -- of the columns. By making REGION_KEY, which is an integer column, the
205 -- second column in the table, we will have errors because the second
206 -- column in the file is a VARCHAR field.
207
208 -- 4.4.3 Recreate the NATION table and execute the COPY INTO statement.
209 -- Note that in this case, we are using RETURN_ERRORS. Like
210 -- RETURN_ALL_ERRORS, it will return any and all errors stemming from
```

Below the code, there is a 'Results' tab showing a table named 'ERROR' with the message 'Query produced no results'. To the right, the 'Query Details' pane shows a duration of 535ms, 0 rows, and a query ID of 01b89ede-0004-3f59-0... There is also an 'Ask Copilot' button.

Important Noticing: There were no errors because the query produced no results

4.4.3 Editing the Nation table so that when we load the data again we can see errors



The screenshot shows the AWS Lambda SQL interface. On the left, the database schema is visible, including the PUBLIC schema which contains the CUSTOMER and REGION tables. A query is being run in the main pane:

```

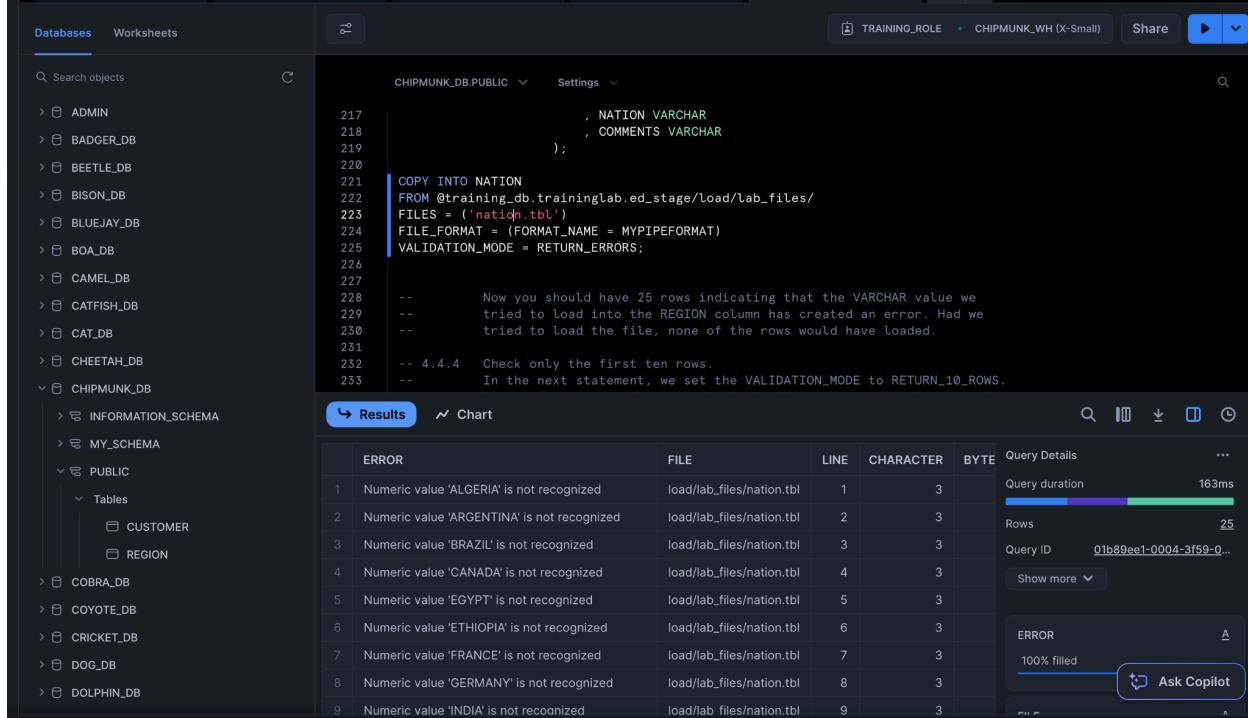
211 --      the loading of the file indicated in the COPY INTO statement below.
212
213
214 CREATE OR REPLACE TABLE nation (
215     NATION_KEY INTEGER
216     , REGION_KEY INTEGER
217     , NATION VARCHAR
218     , COMMENTS VARCHAR
219 );
220
221 COPY INTO NATION
222 FROM @training_db.traininglab.ed_stage/load/lab_files/
223 FILES = ('nation.tbl')
224 FILE_FORMAT = (FORMAT_NAME = MYPipeFormat)
225 VALIDATION_MODE = RETURN_ERRORS;
226
227

```

The results pane shows a single row in the status table:

status	
1	Table NATION successfully created.

Query Details: Query duration 340ms, Rows 1, Query ID 01b89ee0-0004-3f59-0...



The screenshot shows the same setup as above, but with the VALIDATION_MODE set to RETURN_10_ROWS. The results pane now displays 25 error rows:

ERROR				
	FILE	LINE	CHARACTER	BYTE
1	load/lab_files/nation.tbl	1	3	
2	load/lab_files/nation.tbl	2	3	
3	load/lab_files/nation.tbl	3	3	
4	load/lab_files/nation.tbl	4	3	
5	load/lab_files/nation.tbl	5	3	
6	load/lab_files/nation.tbl	6	3	
7	load/lab_files/nation.tbl	7	3	
8	load/lab_files/nation.tbl	8	3	
9	load/lab_files/nation.tbl	9	3	

Query Details: Query duration 163ms, Rows 25, Query ID 01b89ee1-0004-3f59-0...

Important Noticing: The Nation column is getting errors because of the change made to the order of columns.

4.4.4 Checking the first ten rows and returning the first error that occurs

Databases Worksheets CHIPMUNK_DB.PUBLIC Settings

```

232 -- 4.4.4  Check only the first ten rows.
233 -- In the next statement, we set the VALIDATION_MODE to RETURN_10_ROWS.
234 -- So, our statement will only check the first ten rows and return the
235 -- first error it encounters.
236
237
238 COPY INTO NATION
239 FROM @training_db.traininglab.ed_stage/load/lab_files/
240 FILES = ('nation.tbl')
241 FILE_FORMAT = (FORMAT_NAME = MYPipeFormat)
242 VALIDATION_MODE = RETURN_10_ROWS;
243
244
245 -- As you can see, we have a message saying that the numeric value
246 -- ALGERIA is not recognized.
247
248 -- 4.5.0  Error Handling

```

Results Chart

Numeric value 'ALGERIA' is not recognized

File 'Load/Lab_files/nation.tbl', line 1, character 3
Row 1, column "NATION"["REGION_KEY":2]

Ask Copilot

4.5.2 Noticing the error created by changing the region 1 value to a varchar

Databases Worksheets CHIPMUNK_DB.PUBLIC Settings

```

272
273     n.$1 AS N_KEY
274     , n.$2 AS NATION
275     , CASE
276         WHEN n.$3 = 1 THEN 'AMERICA'
277             ELSE n.$3
278         END AS R_KEY
279     , n.$4 AS COMMENTS
280
281
282
283
284 -- 4.5.3  Attempt to load the data.
285 -- Notice that we've set the ON_ERROR parameter to continue. This means
286 -- that all rows that don't generate an error will get loaded.
287
288

```

Results Chart

	N_KEY	NATION	R_KEY	COMMENTS
17	16	MOZAMBIQUE	0	s. ironic, unusual asymptotes wake blithely r
18	17	PERU	AMERICA	platelets. blithely pending dependencies use fluffy across the ever
19	18	CHINA	2	c dependencies. furiously express notoris sleep slyly regular acco
20	19	ROMANIA	3	ular asymptotes are about the furious multipliers. express depende
21	20	SAUDI ARABIA	4	ts. silent requests haggle. closely express packages sleep across th
22	21	VIETNAM	2	hely enticingly express accounts. even, final
23	22	RUSSIA	3	requests against the platelets use never according to the quickly re
24	23	UNITED KINGDOM	3	eans boost carefully special requests. accounts are. carefull
25	24	UNITED STATES	AMERICA	y final packages. slow foxes cajole quickly. quickly silent platelets b

Query Details

Query duration 152ms

Rows 25

Query ID 01b89ee4-0004-3f59-0...

Show more ▾

N_KEY

100% filled

Ask Copilot

4.5.3 Partially loading the table because of the error created when making the region 1 a varchar

The screenshot shows the Snowflake interface with the database set to 'CHIPMUNK_DB.PUBLIC'. In the left sidebar, under the 'Tables' section of the 'PUBLIC' schema, there are two tables: 'CUSTOMER' and 'REGION'. The 'REGION' table is selected. In the main pane, a query is being run:

```

281
282
283
284 -- 4.5.3 Attempt to Load the data.
285 -- Notice that we've set the ON_ERROR parameter to continue. This means
286 -- that all rows that don't generate an error will get loaded.
287
288 COPY INTO nation
289 FROM (
290     SELECT
291         n.$1 AS N_KEY
292         , n.$2 AS NATION
293         , CASE
294             WHEN n.$3 = 1 THEN 'AMERICA'
295             ELSE n.$3
296         END AS R_KEY
297

```

The results show one row was partially loaded:

file	status	rows_parsed	rows_loaded	error_lim
1 s3://snowflakeed/load/lab_files/nation.tbl	PARTIALLY_LOADED	25	20	25

Query Details: Query duration 870ms, Rows 1, Query ID 01b89ee7-0004-3f5c-0...

4.5.4 Loaded data does not contain the info for region 1 observations

The screenshot shows the Snowflake interface with the database set to 'CHIPMUNK_DB.PUBLIC'. In the left sidebar, under the 'Tables' section of the 'PUBLIC' schema, there are two tables: 'CUSTOMER' and 'REGION'. The 'CUSTOMER' table is selected. In the main pane, a query is being run:

```

304
305
306 -- As you can see from the results, the status is PARTIALLY_LOADED and
307 -- rows_loaded is 20 out of the original 25.
308
309 -- 4.5.4 Run the SELECT statement below to verify the contents of the table,
310 -- then truncate the table.
311
312
313 | SELECT * FROM nation;
314
315 TRUNCATE TABLE nation;
316
317
318
319 -- 4.5.5 Retry the insert with ON_ERROR = ABORT_STATEMENT.
320 -- ABORT_STATEMENT is the default value, and it will cause the entire

```

The results show the 'nation' table has 13 rows:

NATION_KEY	NATION	REGION_KEY	COMMENTS
3	5	ETHIOPIA	ven packages wake quickly, regu
4	6	FRANCE	refully final requests, regular, ironi
5	7	GERMANY	l platelets, regular accounts x-ray: unusual, regular acc
6	8	INDIA	ss excuses cajole slyly across the packages, deposits f
7	9	INDONESIA	slyly express asymptotes, regular deposits haggle sly
8	10	IRAN	efully alongside of the slyly final dependencies.
9	11	IRAQ	nic deposits boost atop the quickly final requests? quic
10	12	JAPAN	ously, final, express gifts cajole a
11	13	JORDAN	ic deposits are blithely about the carefully regular pa

Query Details: Query duration 214ms, Rows 20, Query ID 01b89ee7-0004-3f59-0...

4.5.6 Attempting the reload but adding an abort call when an error occurs

The screenshot shows the Snowflake interface with the following details:

- Databases:** CHIPMUNK_DB.PUBLIC
- Code:**

```

328      , n.$2 AS NATION
329      , CASE
330          WHEN n.$3 = 1 THEN 'AMERICA'
331          ELSE n.$3
332      END AS R_KEY
333      , n.$4 AS COMMENTS
334
335      FROM @training_db.traininglab.ed_stage/load/lab_files/nation.tbl (file_format => 'MYPIPEFORMAT') n
336
337 FILE_FORMAT = (FORMAT_NAME = MYPIPEFORMAT)
338 ON_ERROR = ABORT_STATEMENT; --ALSO THE DEFAULT OPTION WHEN ON_ERROR ISN'T SPECIFIED
339
340
341 --     As you can see, the error is Numeric value AMERICA is not recognized.
342 --     No data was loaded.
343
344 -- 4.5.6 Retry the insert with ON_ERROR = SKIP_FILE_4.
    
```
- Results:** A warning message: "Numeric value 'AMERICA' is not recognized". Below it, a note: "If you would like to continue loading when an error is encountered, use other values such as 'SKIP_FILE' or 'CONTINUE' for the ON_ERROR option. For more information on loading options, please run 'info loading_data' in a SQL client."
- Query Details:**
 - Query duration: 601ms
 - Rows: 0
 - Query ID: 01b89eeb-0004-3f5c...
- Buttons:** Share, Ask Copilot

Continuing but using the skip_file_4 call to not load if there are 4 or more errors

The screenshot shows the Snowflake interface with the following details:

- Databases:** CHIPMUNK_DB.PUBLIC
- Code:**

```

350 COPY INTO nation
351   FROM (
352       SELECT
353           n.$1 AS N_KEY
354           , n.$2 AS NATION
355           , CASE
356               WHEN n.$3 = 1 THEN 'AMERICA'
357               ELSE n.$3
358           END AS R_KEY
359           , n.$4 AS COMMENTS
360
361           FROM @training_db.traininglab.ed_stage/load/lab_files/nation.tbl (file_format => 'MYPIPEFORMAT') n
362
363 FILE_FORMAT = (FORMAT_NAME = MYPIPEFORMAT)
364 ON_ERROR = SKIP_FILE_4;
365
366
    
```
- Results:** A table showing the load status:

file	status	rows_parsed	rows_loaded	error_limit	er
1 s3://snowflakeed/load/lab_files/nation.tbl	LOAD_FAILED	25	0	4	
- Query Details:**
 - Query duration: 510ms
 - Rows: 1
 - Query ID: 01b89eeb-0004-3f59-0...
- Buttons:** Share, Ask Copilot

4.5.7 Querying the table after the failed load and getting no results

The screenshot shows the Snowflake interface with the following details:

- Databases:** CHIPMUNK_DB.PUBLIC
- ON_ERROR = SKIP_FILE_4;**
- Query:** SELECT * FROM 4.nation;
- Results:** Query produced no results.
- Query Details:**
 - Query duration: 83ms
 - Rows: 0
 - Query ID: 01b89eeb-0004-3f59-0...
- Ask Copilot** button.

4.5.8 Reloading but using skip_file_6 so that the data will load

The screenshot shows the Snowflake interface with the following details:

- Databases:** CHIPMUNK_DB.PUBLIC
- Query:** COPY INTO nation
FROM (
 SELECT
 n.\$1 AS N_KEY
 , n.\$2 AS NATION
 , CASE
 WHEN n.\$3 = 1 THEN 'AMERICA'
 ELSE n.\$3
 END AS R_KEY
 , n.\$4 AS COMMENTS
 FROM @training_db.traininglab.ed_stage/load/lab_files/nation.tbl (file_format => 'MYPIPEFORMAT') n
)
- FILE_FORMAT = (FORMAT_NAME = MYPIPEFORMAT)**
- ON_ERROR = SKIP_FILE_6;**
- Results:** file | status | rows_parsed | rows_loaded | error_lim

file	status	rows_parsed	rows_loaded	error_lim
s3://snowflakeed/load/lab_files/nation.tbl	PARTIALLY_LOADED	25	20	

- Query Details:**
 - Query duration: 637ms
 - Rows: 1
 - Query ID: 01b89eec-0004-3f5c-0...
- Ask Copilot** button.

Takeaways:

Files loaded can be compressed or uncompressed and we can use copy into to load. There are also various commands that can be run to partially load the data if there are errors on certain rows.