

Essentials Lab 8 - Part 2

Exercise 1

- 1) Download files from:

[https://www.snowflakeuniversity.com/E_L8/downloads/Essentials Lesson 8 Part 2_files.zip](https://www.snowflakeuniversity.com/E_L8/downloads/Essentials%20Lesson%208%20Part%202_files.zip)

- 2) Unzip the files and find the file named **create_WEIGHT_INGEST_table.txt**

- 3) Load the file into an empty worksheet.

- 4) Click **[Run]**.

NOTE: Remember to make sure your worksheet context is set to your USDA Database and the PUBLIC schema, otherwise the table might be created in another database or schema.

The screenshot shows a Snowflake SQL worksheet interface. At the top, there is a blue 'Run' button, a 'All Queries' dropdown, and a timestamp 'Saved a minute ago'. Below this, the SQL code for creating the 'WEIGHT_INGEST' table is displayed, numbered from 1 to 9. The table has seven columns: NDB_NO (VARCHAR(7)), SEQ (VARCHAR(4)), AMOUNT (NUMBER(6,3)), MSRE_DESC (VARCHAR(86)), GM_WGT (NUMBER(7,1)), NUM_DATA PTS (NUMBER(4,0)), and STD_DEV (NUMBER(7,3)).

The results section shows a single row indicating the table was successfully created:

Row	status
1	Table WEIGHT_INGEST successfully created.

Below the results, there are buttons for 'Filter result...', 'Download', and 'Copy'.

Essentials Lab 8 - Part 2

Exercise 2

1) Check to ensure you still have your **USDA_FILE_FORMAT** file format. You would have created this file format in the lab for Lesson 5.

a) If your file format exists:
i) Go to the next screen
And begin **Exercise 3**

b) If the file format is not there:

- i) Recreate it using the File Format wizard,
OR,
- ii) Run a script from the file named **create_USDA_FILE_FORMAT_file_format.txt**

Databases > **USDA_NUTRIENT_STDREF**

File Format	Schema	Type	Creation Time
USDA_FILE_FORMAT	PUBLIC	CSV	1/31/19 3:53:43 PM

Essentials Lab 8 - Part 2

Exercise 3

- 1) Type a **COPY INTO** command into an open worksheet. Use the example shown here.
- 2) Click **[Run]**.

The screenshot shows a Snowflake worksheet interface. At the top, there is a blue button labeled "Run" and a status bar indicating "All Queries | Saved a few seconds ago". On the right, there is a "Context" icon. The main area contains a code editor with the following SQL command:

```
1 COPY INTO WEIGHT_INGEST
2 FROM @MY_S3_BUCKET/load/
3 FILES = ('WEIGHT.txt')
4 FILE_FORMAT = (FORMAT_NAME = USDA_FILE_FORMAT);
5
6
```

Below the code editor, there are two tabs: "Results" (which is selected) and "Data Preview". Under "Results", there is a summary: "Query ID" (with a link), "SQL" (with a link), "1.36s" (execution time), and "1 rows" (number of rows). There is also a "Filter result..." input field and "Copy" and "Download" buttons. A data preview table is shown below:

Row	file ↓	status	rows_parsed	rows_loaded	error_limit
1	s3://on-demand-files/load/WEIGHT.txt	LOADED	14449	14449	1

Essentials Lab 8 - Part 2

Exercise 4

Based on the **Results**

Pane shown here, the file has been loaded into the table. When you have a message like this:

- 1) Try clicking the **[Run]** button again and notice the message that appears in the **Results Pane**.

Notice that Snowflake tracks whether a file has been loaded and doesn't let you load it twice by accident.

The screenshot shows the Snowflake interface with a query editor and results pane. The query editor contains the following SQL code:

```
1 COPY INTO WEIGHT_INGEST
2 FROM @MY_S3_BUCKET/load/
3 FILES = ('WEIGHT.txt')
4 FILE_FORMAT = (FORMAT_NAME = USDA_FILE_FORMAT);
5
6
```

The results pane shows the output of the query. It includes a summary bar at the top indicating a successful run with a duration of 1.36s and 1 row processed. Below this is a table with columns: Row, file, status, rows_parsed, rows_loaded, and error_limit. The first row shows the file 's3://on-demand-files/load/WEIGHT.txt' with status 'LOADED', rows_parsed '14449', rows_loaded '14449', and error_limit '1'. This row is highlighted with a red box. Below this table is another table with columns: Row, file, status, and first_error. The single row shows the same file with status 'LOAD_SKIPPED' and the message 'File was loaded before.'

Row	file	status	rows_parsed	rows_loaded	error_limit
1	s3://on-demand-files/load/WEIGHT.txt	LOADED	14449	14449	1

Row	file	status	first_error
1	s3://on-demand-files/load/WEIGHT.txt	LOAD_SKIPPED	File was loaded before.

Essentials Lab 8 - Part 2

Exercise 5

We'd like to reload the file into the table, we'll **FORCE** the file to reload.

- 1) Add the phrase

FORCE = TRUE

to the end of the

COPY INTO statement

(as seen in line 7 of the screen shot).

- 2) Click **[Run]**.

The problem with using the **FORCE = TRUE** attribute value is that you have now loaded the table with double the rows. What happens if you **TRUNCATE** the table and set **FORCE=FALSE**. Will the file load?

The screenshot shows a Snowflake query editor interface. At the top, there is a blue 'Run' button, a 'All Queries' link, and a timestamp 'Saved a few seconds ago'. Below the editor area, the SQL code is displayed:

```
1  
2  
3 COPY INTO WEIGHT_INGEST  
4 FROM @MY_S3_BUCKET/load/  
5 FILES = ('WEIGHT.txt')  
6 FILE_FORMAT = (FORMAT_NAME = USDA_FILE_FORMAT)  
7 FORCE = TRUE;  
8  
9
```

The results section is active, showing the following details:

- Query ID: [redacted]
- SQL: [redacted]
- Time: 2.43s
- Rows: 1

Below the results, there is a table with the following data:

Row	file	status	rows_parsed	rows_loaded
1	s3://on-demand...	LOADED	14449	14449

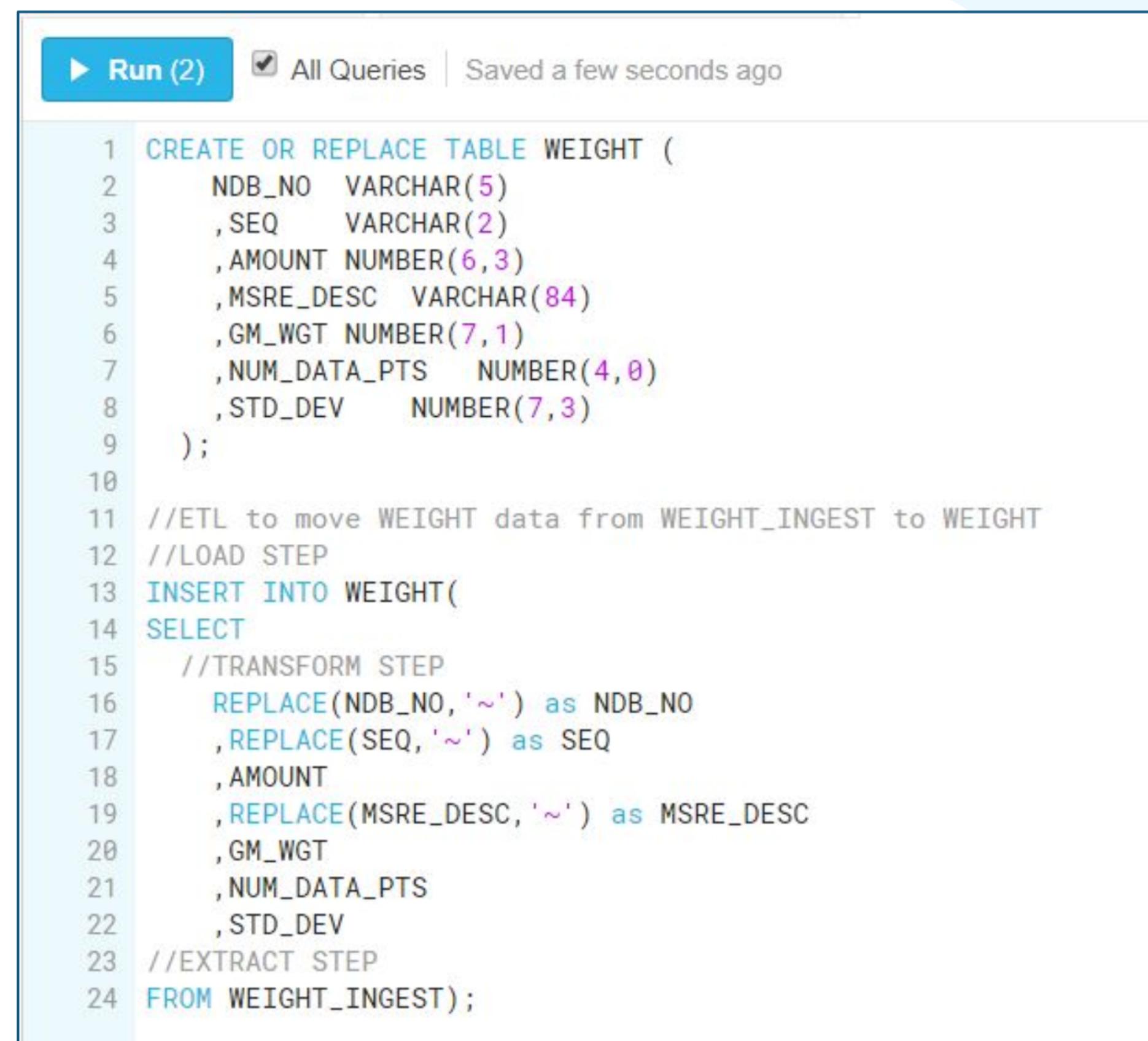
Essentials Lab 8 - Part 2

Exercise 6

Remember that with the previous USDA file we were required to run an ETL to move the data and remove the tildes during the transfer. We'll do that again for our WEIGHT data.

- 1) Open the file named
create_WEIGHT_table_AND_run_ETL.txt
- 2) Check the **[All Queries]** check box.
- 3) Click **[Run (2)]**

NOTE: It would be nice if we were able to remove the tildes while loading the file from the COPY INTO statement. Then, we would not need an ingest table and would not need to run an ETL to transfer the data. In the next lab exercise, you will learn how to remove the tildes during the COPY INTO statement.



```
▶ Run (2)  All Queries | Saved a few seconds ago

1 CREATE OR REPLACE TABLE WEIGHT (
2   NDB_NO  VARCHAR(5)
3   ,SEQ    VARCHAR(2)
4   ,AMOUNT NUMBER(6,3)
5   ,MSRE_DESC VARCHAR(84)
6   ,GM_WGT NUMBER(7,1)
7   ,NUM_DATA_PTS NUMBER(4,0)
8   ,STD_DEV  NUMBER(7,3)
9 );
10
11 //ETL to move WEIGHT data from WEIGHT_INGEST to WEIGHT
12 //LOAD STEP
13 INSERT INTO WEIGHT(
14   SELECT
15     //TRANSFORM STEP
16     REPLACE(NDB_NO, '~') as NDB_NO
17     ,REPLACE(SEQ, '~') as SEQ
18     ,AMOUNT
19     ,REPLACE(MSRE_DESC, '~') as MSRE_DESC
20     ,GM_WGT
21     ,NUM_DATA_PTS
22     ,STD_DEV
23   //EXTRACT STEP
24   FROM WEIGHT_INGEST);
```



Essentials Lab 8 - Part 2

Exercise 7

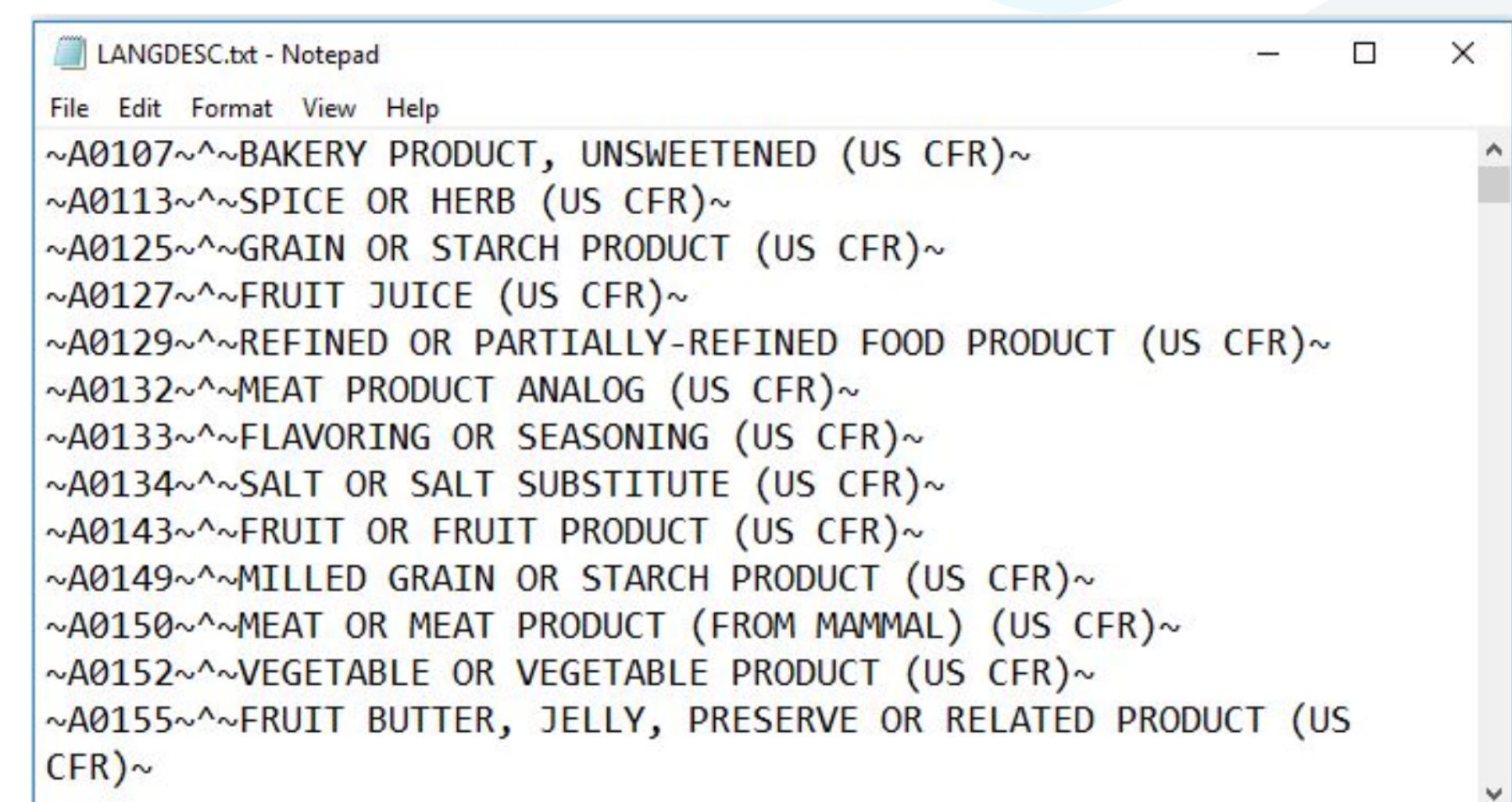
We'll now create yet another USDA data table using a script. We won't make room for the tildes this time because we'll be removing the tildes during the COPY INTO step.

- 1) Review the screenshot at left, taken from the USDA Specification Document.
- 2) Review the screenshot at left that shows the data file opened in a text editor.
- 3) Create a new table in your USDA database using either the **Create Table wizard** or by writing a SQL script. You do not need to leave room for the tildes because we will be removing them prior to the load.

Table 5. LanguaL Factor Description File Format

Field name	Type	Blank	Description
Factor_Code	A 5*	N	LanguaL factor from the LanguaL thesaurus; includes only codes used to assign factor to foods in the LanguaL Factor file
Description	A 140	N	Description of the LanguaL factor code from the LanguaL thesaurus

* Primary key for the LanguaL Factor Description file



The screenshot shows a Windows Notepad window titled "LANGDESC.txt - Notepad". The window contains a list of food product codes and their descriptions, separated by tilde (~). The list includes:

- ~A0107~~BAKERY PRODUCT, UNSWEETENED (US CFR)~
- ~A0113~~SPICE OR HERB (US CFR)~
- ~A0125~~GRAIN OR STARCH PRODUCT (US CFR)~
- ~A0127~~FRUIT JUICE (US CFR)~
- ~A0129~~REFINED OR PARTIALLY-REFINED FOOD PRODUCT (US CFR)~
- ~A0132~~MEAT PRODUCT ANALOG (US CFR)~
- ~A0133~~FLAVORING OR SEASONING (US CFR)~
- ~A0134~~SALT OR SALT SUBSTITUTE (US CFR)~
- ~A0143~~FRUIT OR FRUIT PRODUCT (US CFR)~
- ~A0149~~MILLED GRAIN OR STARCH PRODUCT (US CFR)~
- ~A0150~~MEAT OR MEAT PRODUCT (FROM MAMMAL) (US CFR)~
- ~A0152~~VEGETABLE OR VEGETABLE PRODUCT (US CFR)~
- ~A0155~~FRUIT BUTTER, JELLY, PRESERVE OR RELATED PRODUCT (US CFR)~

Essentials Lab 8 - Part 2

Exercise 8

- 1) Confirm that your table is named and structured like the one shown here.
- 2) If you made any mistakes, you can drop the table and try again. You might want to use code like the second screenshot shown here.

The screenshot shows the Snowflake web interface. At the top, there are navigation icons for Databases, Shares, Warehouses, Worksheets, History, and Account. Below the header, the path 'Databases > USDA_NUTRIENT_STDREF > LANGDESC (PUBLIC)' is displayed. A 'Tables' tab is selected, showing a table structure with two columns: 'FACTOR_CODE' (Ordinal 1, VARCHAR(5), Nullable true, Default NULL) and 'DESCRIPTION' (Ordinal 2, VARCHAR(140), Nullable true, Default NULL). There is also a 'Load Table' button.

```
1 CREATE TABLE LANGDESC (
2   FACTOR_CODE VARCHAR(5)
3   ,DESCRIPTION VARCHAR(140)
4 );
5
```

Table 5. LanguaL Factor Description File Format

Field name	Type	Blank	Description
Factor_Code	A 5*	N	LanguaL factor from the LanguaL thesaurus; includes only codes used to assign factor to foods in the LanguaL Factor file
Description	A 140	N	Description of the LanguaL factor code from the LanguaL thesaurus

* Primary key for the LanguaL Factor Description file

Essentials Lab 8 - Part 2

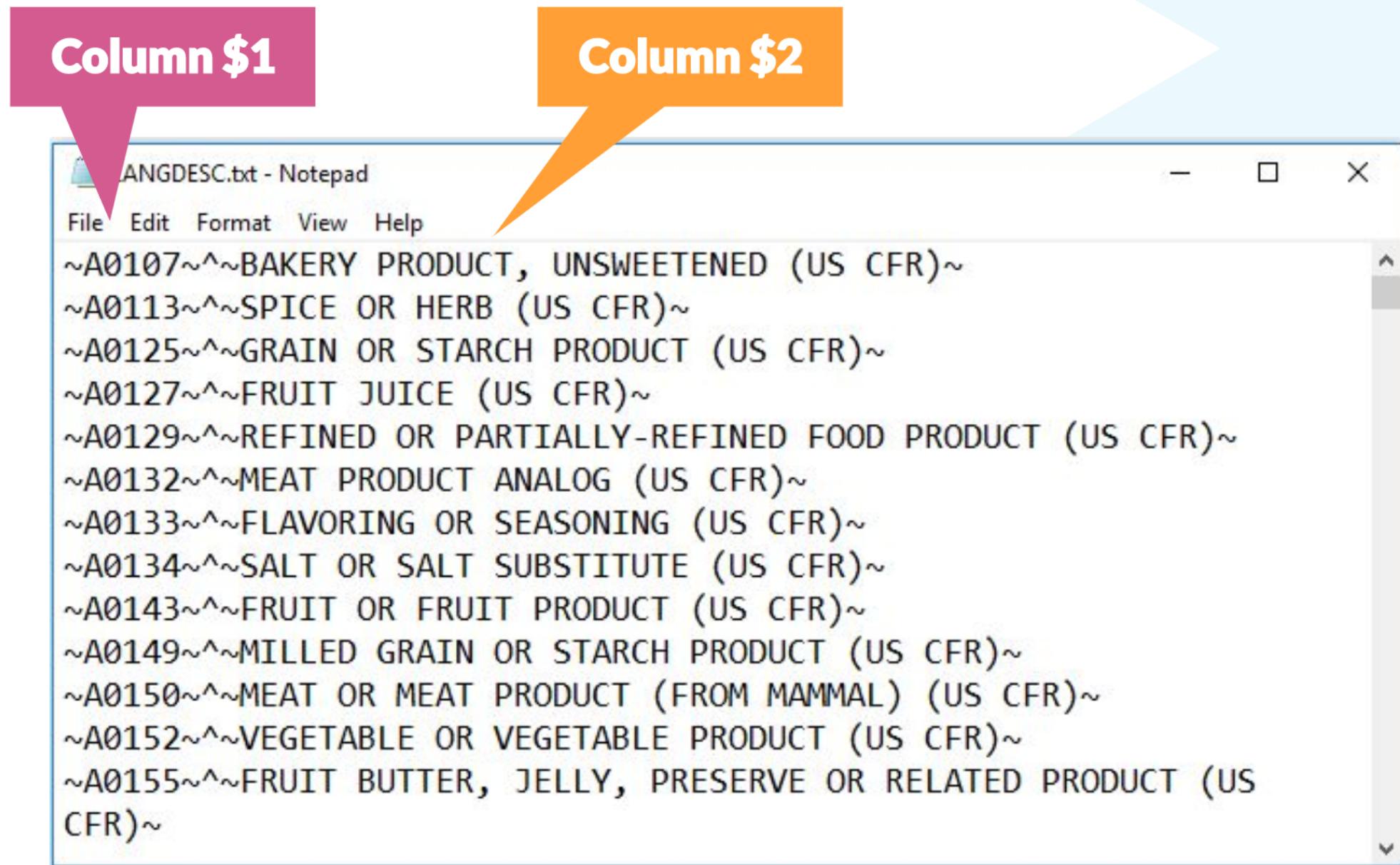
Exercise 9

NOTE: Before loading the data from the file into the Snowflake table, we can actually query the data.

Since the file does not have headers that identify the column names, we have to refer to the columns in sequential order.

We select them using a dollar sign and their sequence number.

- 1) Write and run a select statement like the one shown in the lower screenshot.



ANGDESC.txt - Notepad

File Edit Format View Help

~A0107~~BAKERY PRODUCT, UNSWEETENED (US CFR)~
~A0113~~SPICE OR HERB (US CFR)~
~A0125~~GRAIN OR STARCH PRODUCT (US CFR)~
~A0127~~FRUIT JUICE (US CFR)~
~A0129~~REFINED OR PARTIALLY-REFINED FOOD PRODUCT (US CFR)~
~A0132~~MEAT PRODUCT ANALOG (US CFR)~
~A0133~~FLAVORING OR SEASONING (US CFR)~
~A0134~~SALT OR SALT SUBSTITUTE (US CFR)~
~A0143~~FRUIT OR FRUIT PRODUCT (US CFR)~
~A0149~~MILLED GRAIN OR STARCH PRODUCT (US CFR)~
~A0150~~MEAT OR MEAT PRODUCT (FROM MAMMAL) (US CFR)~
~A0152~~VEGETABLE OR VEGETABLE PRODUCT (US CFR)~
~A0155~~FRUIT BUTTER, JELLY, PRESERVE OR RELATED PRODUCT (US CFR)~

```
1 SELECT $1, $2
2 FROM @MY_S3_BUCKET/load/LANGDESC.txt
3 (FILE_FORMAT => USDA_FILE_FORMAT );
```

Essentials Lab 8 - Part 2

Exercise 10

- 1) Change your select statement so that you return column 2 before column 1 and click **[Run]**.

NOTE: Notice that you can re-order columns while selecting data from the staged file.

- 2) Change your select statement so that it only returns column 1.
- 3) Add a Column Name using the **AS** keyword, and click **[Run]**.

NOTE: Notice that you can filter out columns you don't want to include. You can also rename columns while selecting the data from the staged file.



```
▶ Run All Queries | Saved a few seconds ago  
1 //Re-order columns  
2 SELECT $2, $1  
3 FROM @MY_S3_BUCKET/load/LANGDESC.txt  
4 (FILE_FORMAT => USDA_FILE_FORMAT );  
5  
6 //Restrict columns to a subset, and rename  
7 SELECT $1 as FIRST_COLUMN  
8 FROM @MY_S3_BUCKET/load/LANGDESC.txt  
9 (FILE_FORMAT => USDA_FILE_FORMAT );  
10
```

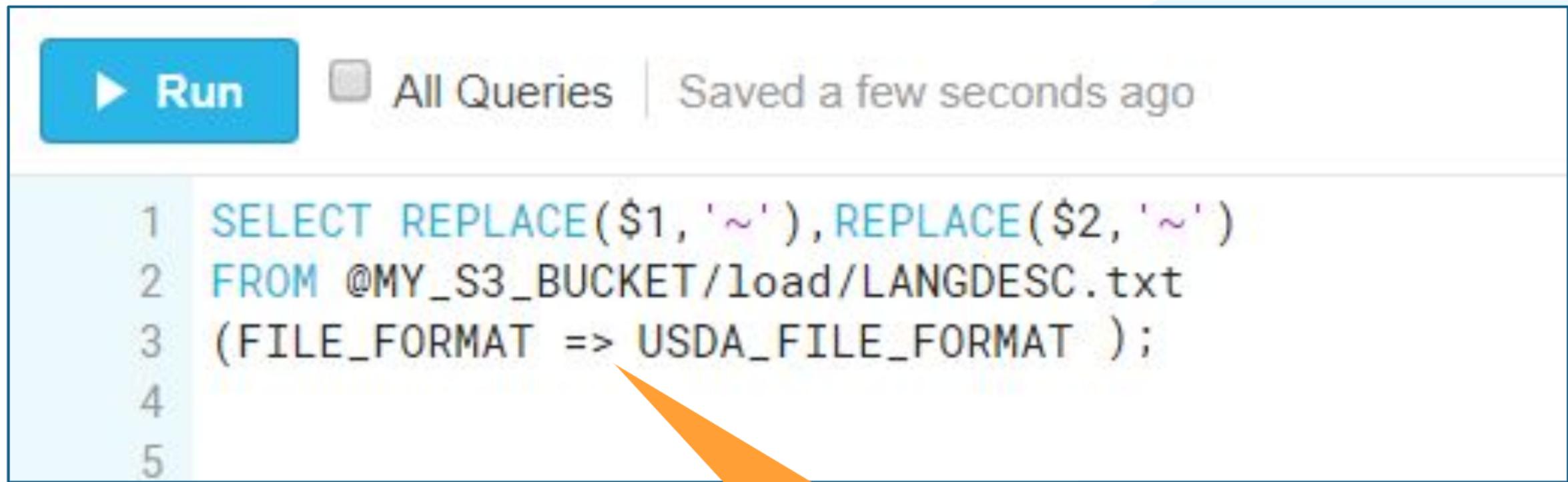


Essentials Lab 8 - Part 2

Exercise 11

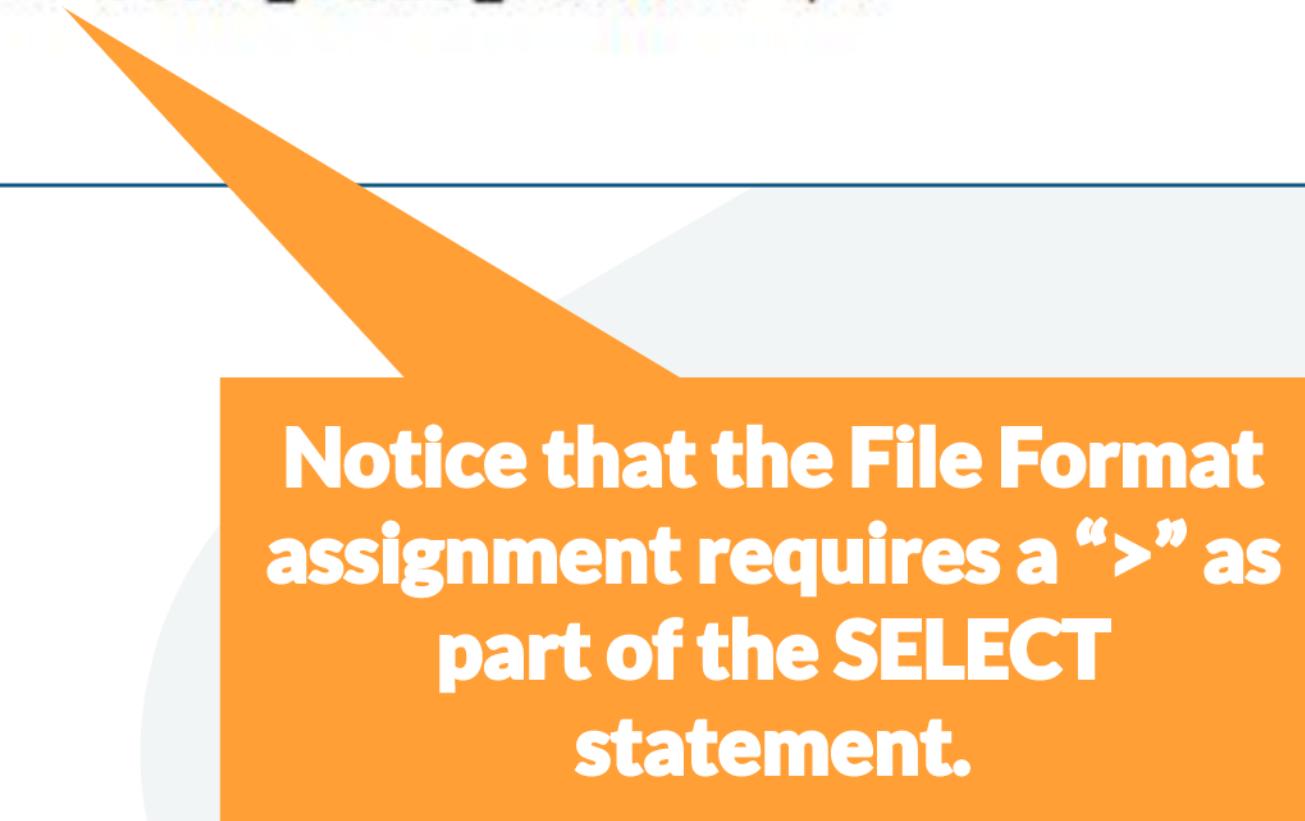
- 1) Use the **REPLACE** function to remove the tildes from the data.
- 2) Click **[Run]**.

NOTE: Now we can combine this Select statement with the COPY INTO statement we need to run in order to load the data from its STAGE file location into the LANGDESC table we created in Exercises 7 & 8 of this Lab.



The screenshot shows a Snowflake query editor interface. At the top, there is a blue button labeled "Run". To the right of the button, it says "All Queries" and "Saved a few seconds ago". Below this, the SQL code is displayed:

```
1 SELECT REPLACE($1, '~'), REPLACE($2, '~')
2 FROM @MY_S3_BUCKET/load/LANGDESC.txt
3 (FILE_FORMAT => USDA_FILE_FORMAT );
```



Notice that the File Format assignment requires a “>” as part of the SELECT statement.

Essentials Lab 8 - Part 2

Exercise 12

- 1) Follow the instructions shown in the callout boxes to convert your **SELECT** statement into a **COPY INTO** statement.

Add parentheses around the **SELECT** statement.

```
1 SELECT REPLACE($1, '~'), REPLACE($2, '~')
2 FROM @MY_S3_BUCKET/load/LANGDESC.txt
3
4 (FILE_FORMAT => USDA_FILE_FORMAT );
5
```

Add a blank line.

```
1 (SELECT REPLACE($1, '~'), REPLACE($2, '~')
2 FROM @MY_S3_BUCKET/load/LANGDESC.txt)
3
4 FILE_FORMAT = USDA_FILE_FORMAT ;
5
```

Remove the **>** sign after the **=** sign.

Remove parentheses from around the **FILE_FORMAT** line.

Add the **COPY INTO** statement, the **TABLE NAME**, the column names, and the keyword **FROM**

```
COPY INTO LANGDESC(FACTOR_CODE, DESCRIPTION)
FROM
(SELECT REPLACE($1, '~'), REPLACE($2, '~')
FROM @MY_S3_BUCKET/load/LANGDESC.txt)
FILE_FORMAT = USDA_FILE_FORMAT ;
```



Essentials Lab 8 - Part 2

Exercise 13

- 1) Run your new **COPY INTO** statement to load the **LANGDESC.txt** file into your **LANGDESC** table without ever loading the tildes.

```
COPY INTO LANGDESC(FACTOR_CODE, DESCRIPTION)
FROM
(SELECT REPLACE($1, '~'), REPLACE($2, '~')
FROM @MY_S3_BUCKET/load/LANGDESC.txt)

FILE_FORMAT = USDA_FILE_FORMAT ;
```



