Essentials Lab 11
Exercise 1

1) Download and unzip the files for this lab.
   https://www.snowflakeuniversity.com/EL11/downloads/Essentials_Lesson_11_Files.zip
2) Load the script named **Lesson_11_Book_Author_Nested_SQL.txt** into a new worksheet.
3) Run the statements on lines 1 through 7. This will create new table into which you will load nested Book and Author data.

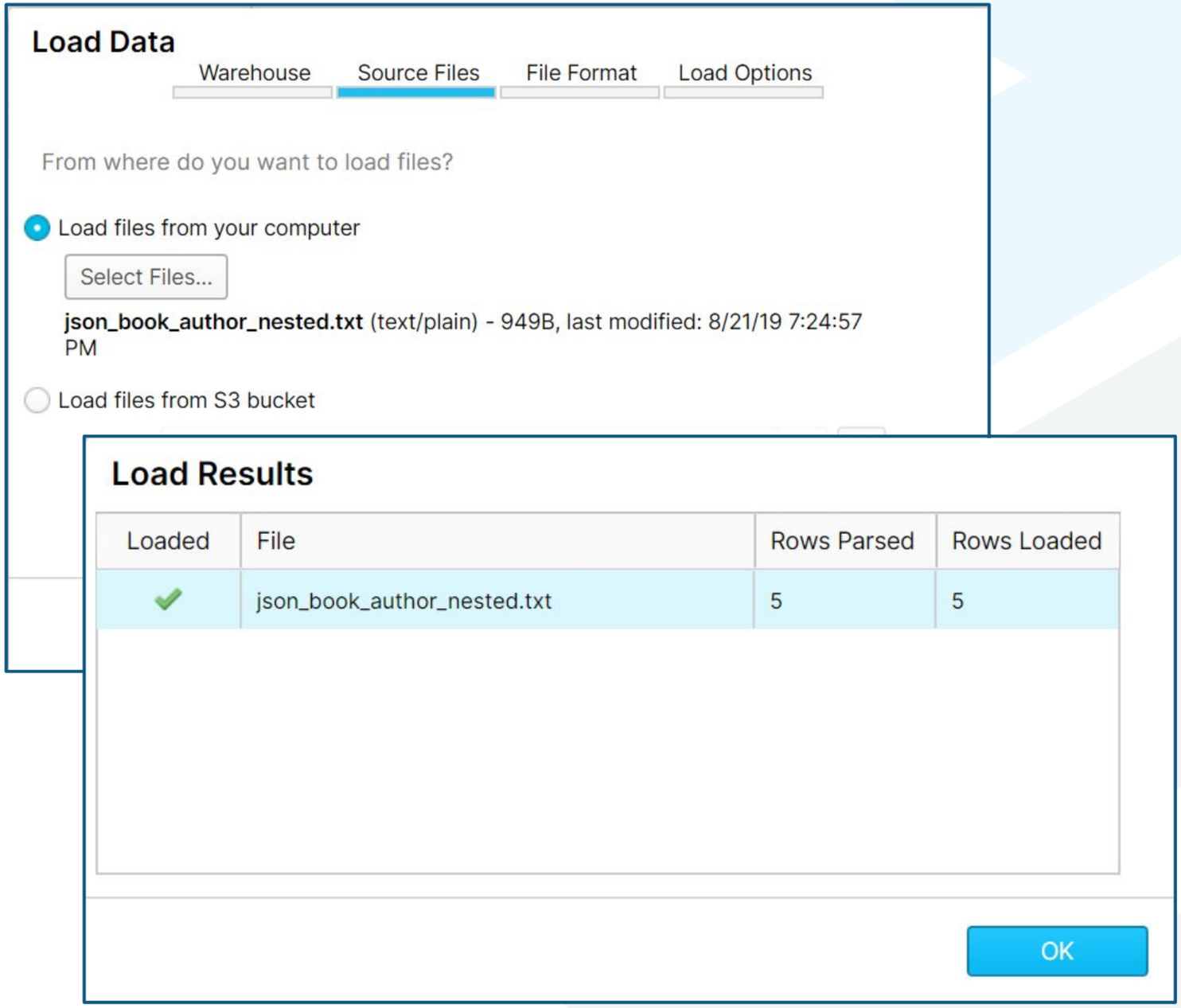
```
1  USE DATABASE LIBRARY_CARD_CATALOG;
2
3  // Create an Ingestion Table for the NESTED JSON Data
4  CREATE OR REPLACE TABLE "LIBRARY_CARD_CATALOG"."PUBLIC"."NESTED_INGEST_JSON"
5  (
6    "RAW_NESTED_BOOK" VARIANT
7  );
8
9  // No need for another File Format, the JSON_FILE_FORMAT from the Lesson 10 lab will work fine.
10
11 //Navigate to the Database Area, locate your new table
12 //load the file json_book_author_nested.txt into the NESTED_INGEST_JSON table
13
```

Essentials Lab 10
Exercise 2

1) Launch the **[Load Table]** Wizard from the ***NESTED_INGEST_JSON*** *table.*

2) Use the following settings:
   a) Choose your smallest warehouse.
   b) Select the ***json_book_author_nested.txt*** file.
   c) Use the ***JSON_FILE_FORMAT*** as the file format.

3) When the load completes, navigate to the table in the **[Worksheets]** area and preview the loaded data.



Essentials Lab 11
Exercise 3

1) Return to the worksheet used in Exercise 1 and explore the **`SELECT`** statements from line 14 to the end of the file.

```
14 //Come back to this worksheet and run the examples shown in the video.
15 SELECT RAW_NESTED_BOOK
16 FROM NESTED_INGEST_JSON;
17
18 SELECT RAW_NESTED_BOOK:year_published
19 FROM NESTED_INGEST_JSON;
20
21 SELECT RAW_NESTED_BOOK:authors
22 FROM NESTED_INGEST_JSON;
23
24 //try changing the number in the bracketsd to return authors from a different row
25 SELECT RAW_NESTED_BOOK:authors[0].first_name
26 FROM NESTED_INGEST_JSON
27
28 //Use these example flatten commands to explore flattening the nested book and author data
29 SELECT value:first_name
30 FROM NESTED_INGEST_JSON
31 ,LATERAL FLATTEN(input => RAW_NESTED_BOOK:authors);
32
33 SELECT value:first_name
34 FROM NESTED_INGEST_JSON
35 ,table(flatten(RAW_NESTED_BOOK:authors));
```

Essentials Lab 11
Exercise 4

1) Load the script named **Lesson_11_Twitter_Nested_SQL.txt** into a new worksheet.

2) Run the statements on lines 1 through 20.

   This will create new database, table and file format. You will use these objects to load and analyze Twitter data.

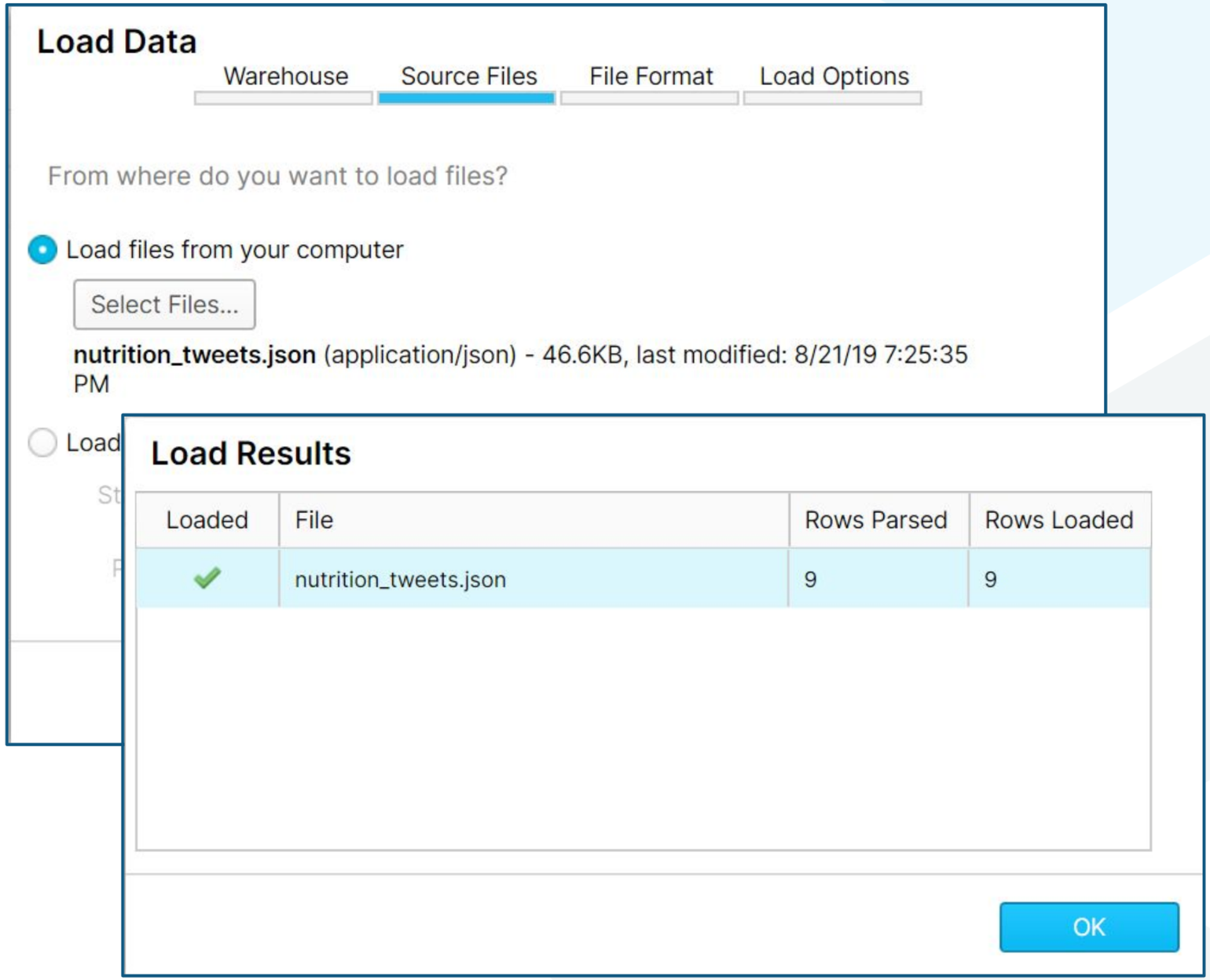
```
1  //Create a new database to hold the Twitter file
2  CREATE DATABASE SOCIAL_MEDIA_FLOODGATES
3  COMMENT = 'There\'s so much data from social media - flood warning';
4
5  USE DATABASE SOCIAL_MEDIA_FLOODGATES;
6
7  //Create a table in the new database
8  CREATE TABLE "SOCIAL_MEDIA_FLOODGATES"."PUBLIC"."TWEET_INGEST"
9  ("RAW_STATUS" VARIANT)
10 COMMENT = 'Bring in tweets, on row per tweet or status entity';
11
12 //Create a JSON file format in the new database
13 CREATE FILE FORMAT "SOCIAL_MEDIA_FLOODGATES"."PUBLIC".JSON_FILE_FORMAT
14 TYPE = 'JSON'
15 COMPRESSION = 'AUTO'
16 ENABLE_OCTAL = FALSE
17 ALLOW_DUPLICATE = FALSE
18 STRIP_OUTER_ARRAY = TRUE
19 STRIP_NULL_VALUES = FALSE
20 IGNORE_UTF8_ERRORS = FALSE;
21
22 //Navigate to the Database Area, locate your new database and table
23 //load the file nutrition_tweets.json into the NESTED_INGEST_JSON table
```

Essentials Lab 10
Exercise 5

1) Launch the **[Load Table]** Wizard from the ***TWEET_INGEST*** table in the ***SOCIAL_MEDIA_FLOODGATES*** database.

2) Use the following settings:
   a) Choose your smallest warehouse.
   b) Select the ***nutrition_tweets.json*** file.
   c) Use the ***JSON_FILE_FORMAT*** as the file format.

3) When the load completes, navigate to the table in the **[Worksheets]** area and preview the loaded data.



Essentials Lab 11
Exercise 6

1) Return to the worksheet used in Exercise 4 and explore the **`SELECT`** statements from line 25 to the end of the file.

```
25 //After loading the file, come back to this worksheet and run the
26 //select statements as seen in the video
27 SELECT RAW_STATUS
28 FROM TWEET_INGEST;
29
30 SELECT RAW_STATUS:entities
31 FROM TWEET_INGEST;
32
33 SELECT RAW_STATUS:entities:hashtags
34 FROM TWEET_INGEST;
35
36 //Explore looking at specific hashtags by adding bracketed numbers
37 //This query returns just he first hashtag in each tweet
38 SELECT RAW_STATUS:entities:hashtags[0].text
39 FROM TWEET_INGEST;
40
41 //This version adds a WHERE clause to get rid of any tweet that
42 //doesn't include any hashtags
43 SELECT RAW_STATUS:entities:hashtags[0].text
44 FROM TWEET_INGEST
45 WHERE RAW_STATUS:entities:hashtags[0].text is not null;
46
47 //Perform a simple CAST on the created_at key
```

Essentials Lab 11
Exercise 7

1) Run the last select statement in the worksheet. It is located on lines 83 to 89 of the worksheet.

2) After running the code to check it and understand the output, highlight the code and use CTRL-C to copy the code.

```
83  //Add the Tweet ID and User ID to the returned table
84  SELECT RAW_STATUS:user:id AS USER_ID
85  ,RAW_STATUS:id AS TWEET_ID
86  ,value:text::VARCHAR AS HASHTAG_TEXT
87  FROM TWEET_INGEST
88  ,LATERAL FLATTEN
89  (input => RAW_STATUS:entities:hashtags);
```

Essentials Lab 11
Exercise 8

1) Navigate to **[Databases]** area.

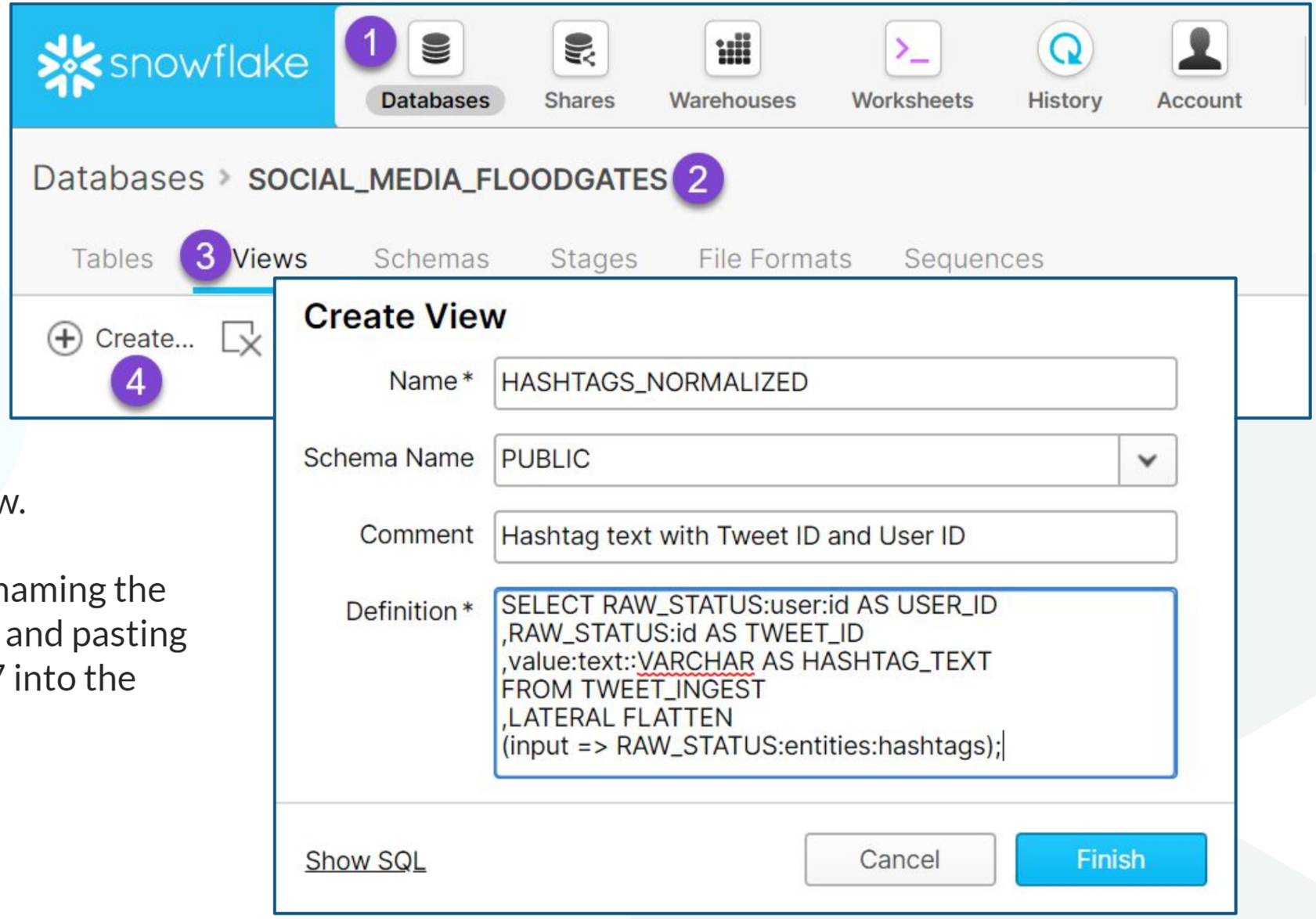
2) Make sure you are in the ***SOCIAL_MEDIA_FLOODGATES*** database.

3) Click the **[Views]** tab.

4) Click **[Create]** to create a new view.

5) Fill out the ***Create View*** dialog by naming the view, filling out the ***Comment*** field and pasting (CTRL-V) the code from Exercise 7 into the ***Definition*** field.

6) Click **[Finish]**.



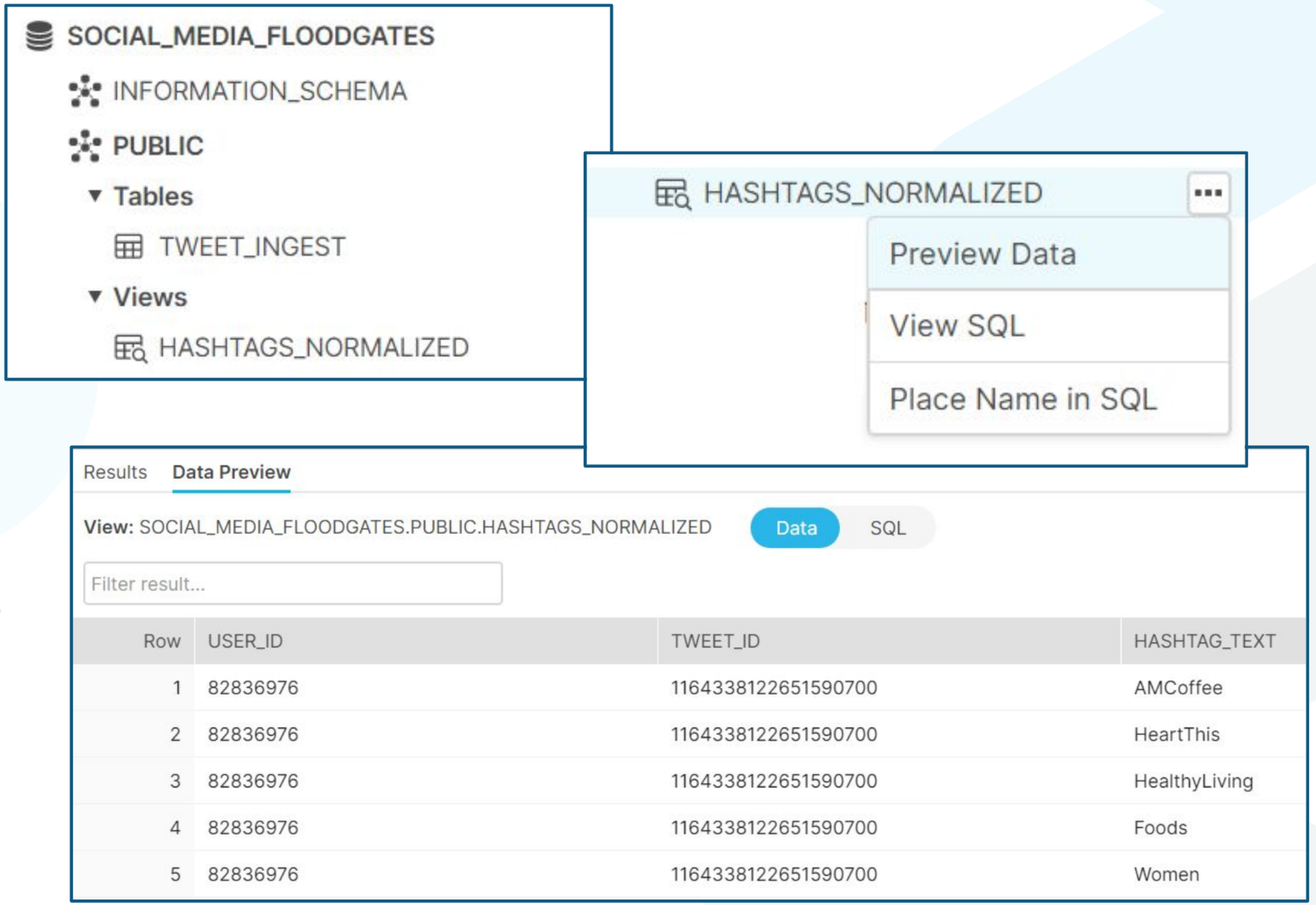
Essentials Lab 11
Exercise 9

1) Navigate to **[Worksheets]** area.

2) Use the ***Navigation Tree*** to locate the view you just created.

3) Use the **[...]** menu on the view's row to select the **[Preview Data]** option.

4) View the data returned in the ***Data Preview*** pane.



Essentials Lab 11
Exercise 10

1) In any worksheet, make sure the database context is set to ***SOCIAL_MEDIA_FLOODGATES***.

2) Type a **`SELECT *`** statement using the newly created view in the **`FROM`** line.

3) Run your **`SELECT`** statement.

NOTE: A view is a saved **`SELECT`** statement that you can use as if it were a table.

```
USE DATABASE SOCIAL_MEDIA_FLOODGATES;

SELECT *
FROM HASHTAGS_NORMALIZED;
```