TeamSQL*

Import Data into Redshift Using the COPY Command

https://blog.teamsql.io/import-data-into-redshift-using-the-copy-command-db89531d3426

Importing a large amount of data into Redshift is easy using the COPY command. To demonstrate this, we'll import the publicly available dataset "Twitter Data for Sentiment Analysis" (see Sentiment140 for additional information).

NOTE: You can connect to AWS Redshift with TeamSQL, a multi-platform DB client that works with Redshift, PostgreSQL, MySQL & Microsoft SQL Server and runs on Mac, Linux and Windows.

You can download TeamSQL for free: https://teamsql.io/

Download the ZIP file containing the training data here.

The Redshift Cluster

For the purposes of this example, the Redshift Cluster's configuration specifications are as follows:

- **Cluster Type**: Single Node
- **Node Type**: dc1.large
- **Zone**: us-east-1a

Create a Database in Redshift

Run the following command to create a new database in your cluster:

```
CREATE DATABASE sentiment;
```

Create a Schema in the sentiment Database

Run the following command to create a scheme within your newly-created database:

```
CREATE SCHEMA tweets;
```

The Schema (Structure) of the Training Data

The CSV file contains the Twitter data with all emoticons removed. There are six columns:

- The polarity of the tweet (key: 0 = negative, 2 = neutral, 4 = positive)
- The id of the tweet (ex. 2087)
- The date of the tweet (ex. Sat May 16 23:58:44 UTC 2009)
- The query (ex. lyx). If there is no query, then this value is NO_QUERY.
- The user that tweeted (ex. robotickilldozr)
- The text of the tweet (ex. Lyx is cool)

Create a table for training data

Begin by creating a table in your database to hold the training data. You an use the following command:

```
CREATE TABLE tweets.training (
    polarity int,
    id BIGINT,
    date_of_tweet varchar,
    query varchar,
    user_id varchar,
    tweet varchar(max)
)
```

Uploading CSV file to S3

To use Redshift's COPY command, your must upload your data source (if it's a file) to S3.

To upload the CSV file to S3:

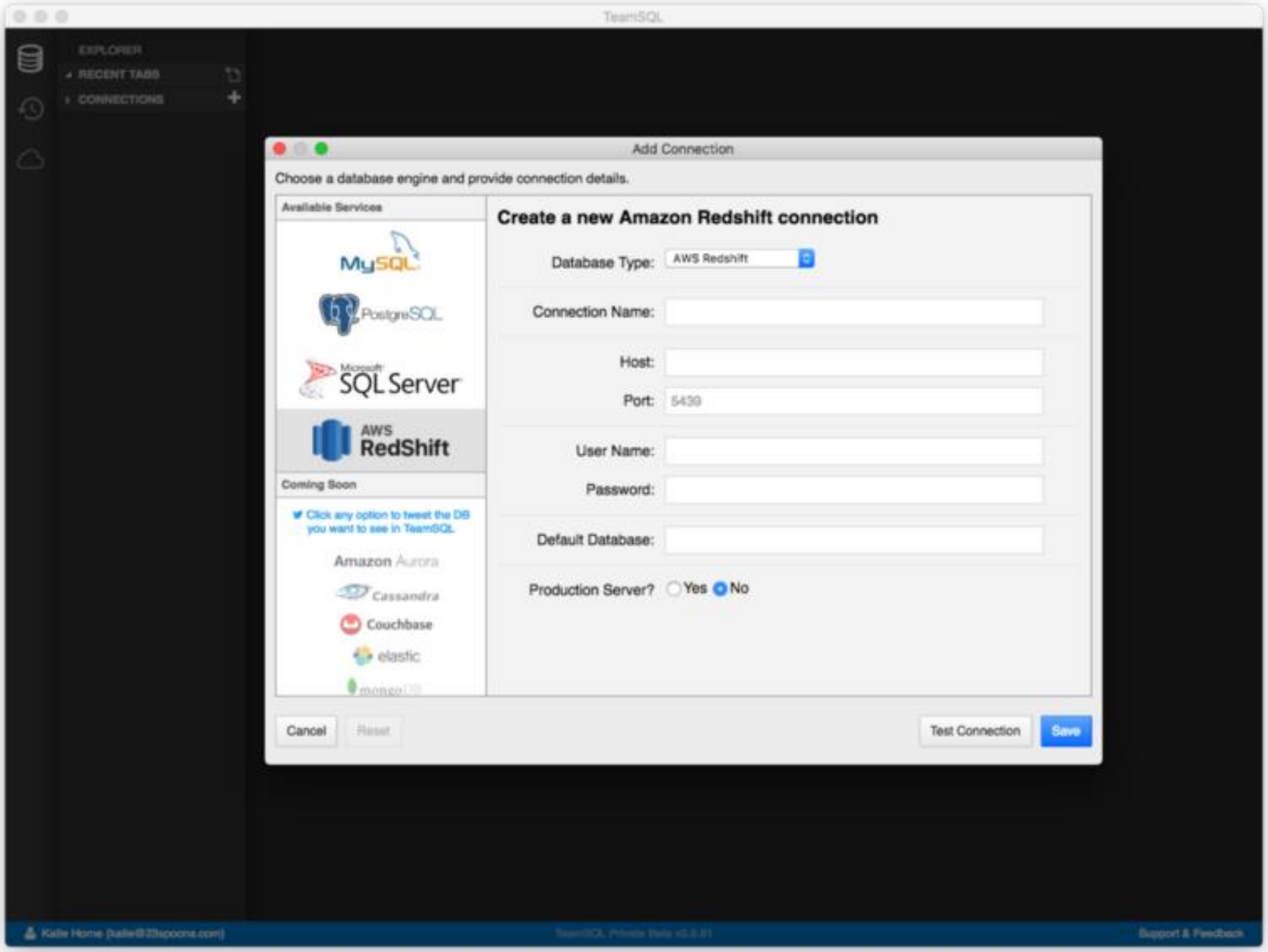
1. Unzip the file you downloaded. You'll see 2 CSV files: one is test data (used to show structure of original dataset), and the other (file name: training.1600000.processed.noemoticon) contains the original data. We will upload and use the latter file.
2. Compress the file. If you're using macOS or Linux, you can compress the file using GZIP by running the following command in Terminal: `gzip training.1600000.processed.noemoticon.csv`
3. Upload your file using the AWS S3 Dashboard.

Alternatively, you can use Terminal/Command Line to upload your file. To do this, you must install AWS CLI and, after installation, configure it (run `aws configure` in your terminal to start the configuration wizard) with your access and secret key.

Connect TeamSQL to the Redshift Cluster and Create the Schema

Open TeamSQL (if you don't have the TeamSQL Client, download it from teamsql.io) and add a new connection.

- Click **Create a Connection** to launch the Add Connection window.



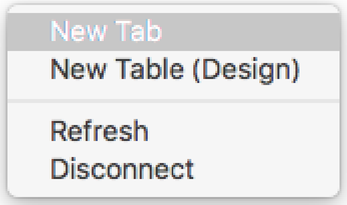
Choose a database engine and provide connection details.

Click any option to tweet the DB you want to see in TeamSQL.

Create a new Amazon Redshift connection

Database Type:	AWS Redshift
Connection Name:	
Host:	
Port:	5439
User Name:	
Password:	
Default Database:	
Production Server?	Yes [✓ No]

- Select Redshift and provide the requested details to set up your new connection.
- Do not forget to enter the **Default Database Name**!
- **Test** the connection, and **save** if the test is successful.
- By default, TeamSQL displays the connections you've added in the left-hand navigation panel. To enable the connection, click on the **socket** icon.
- Right click on default database to open a new tab.

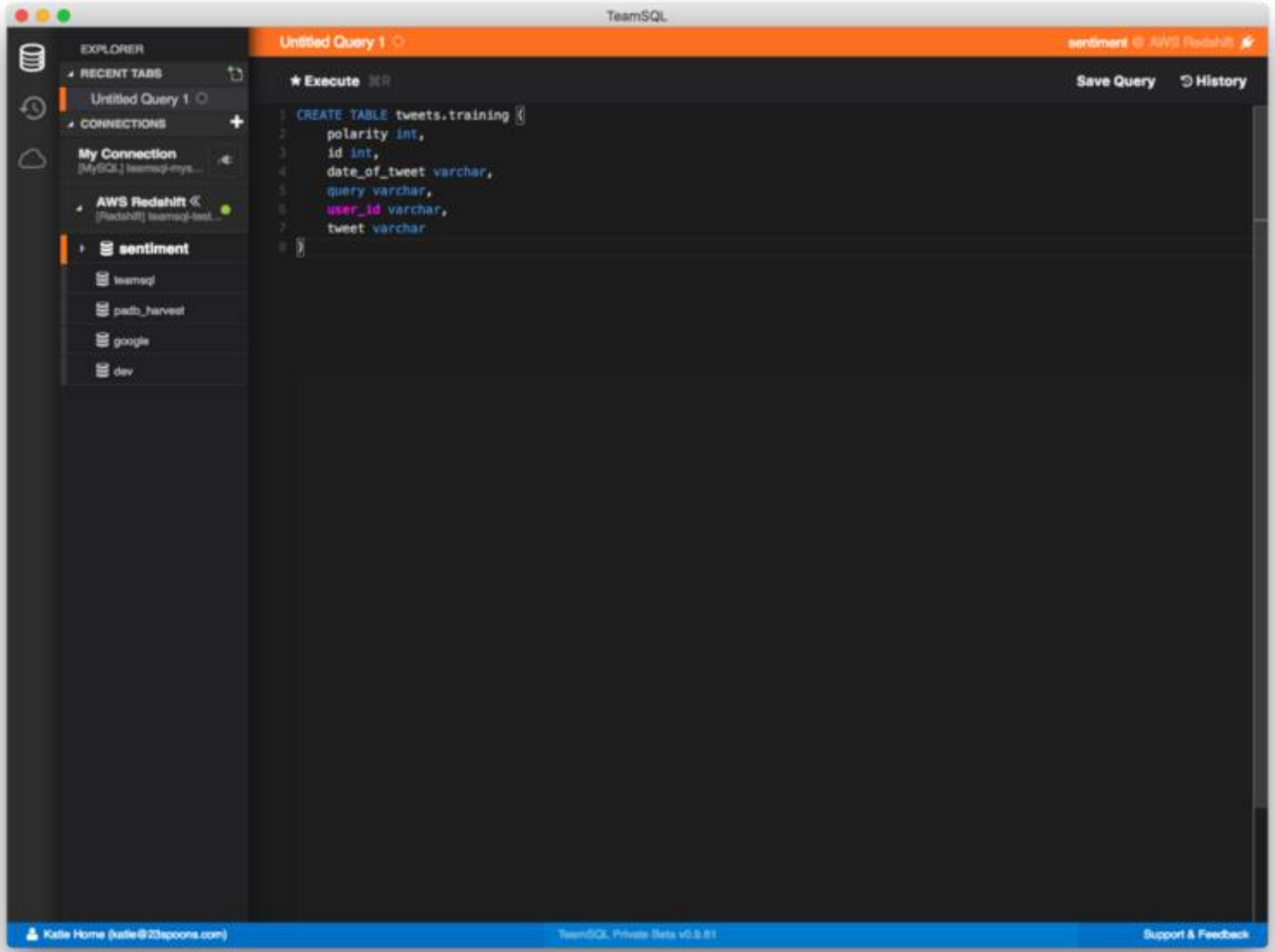


- Run this command to create a new schema in your database.

```
CREATE SCHEMA tweets;
```

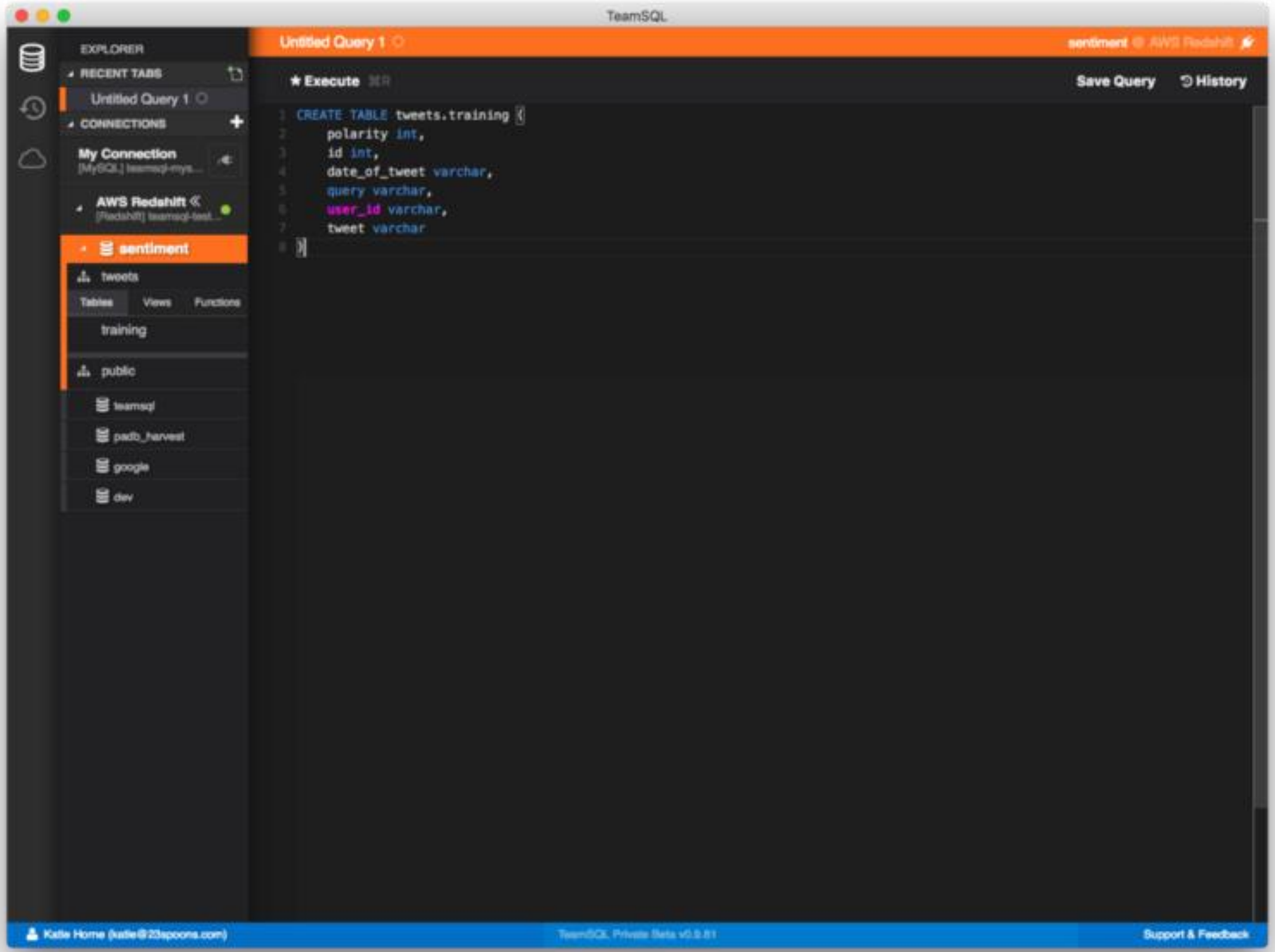
- Refresh the database list in the left-hand navigation panel with right clicking on connection item.
- Create a new table for training data.

```
CREATE TABLE tweets.training (
    polarity int,
    id int,
    date_of_tweet varchar,
    query varchar,
    user_id varchar,
    tweet varchar
)
```



```sql
CREATE TABLE tweets.training (
    polarity int,
    id int,
    date_of_tweet varchar,
    query varchar,
    user_id varchar,
    tweet varchar
)
```

- Refresh the connection and your table should appear in the left-hand list.



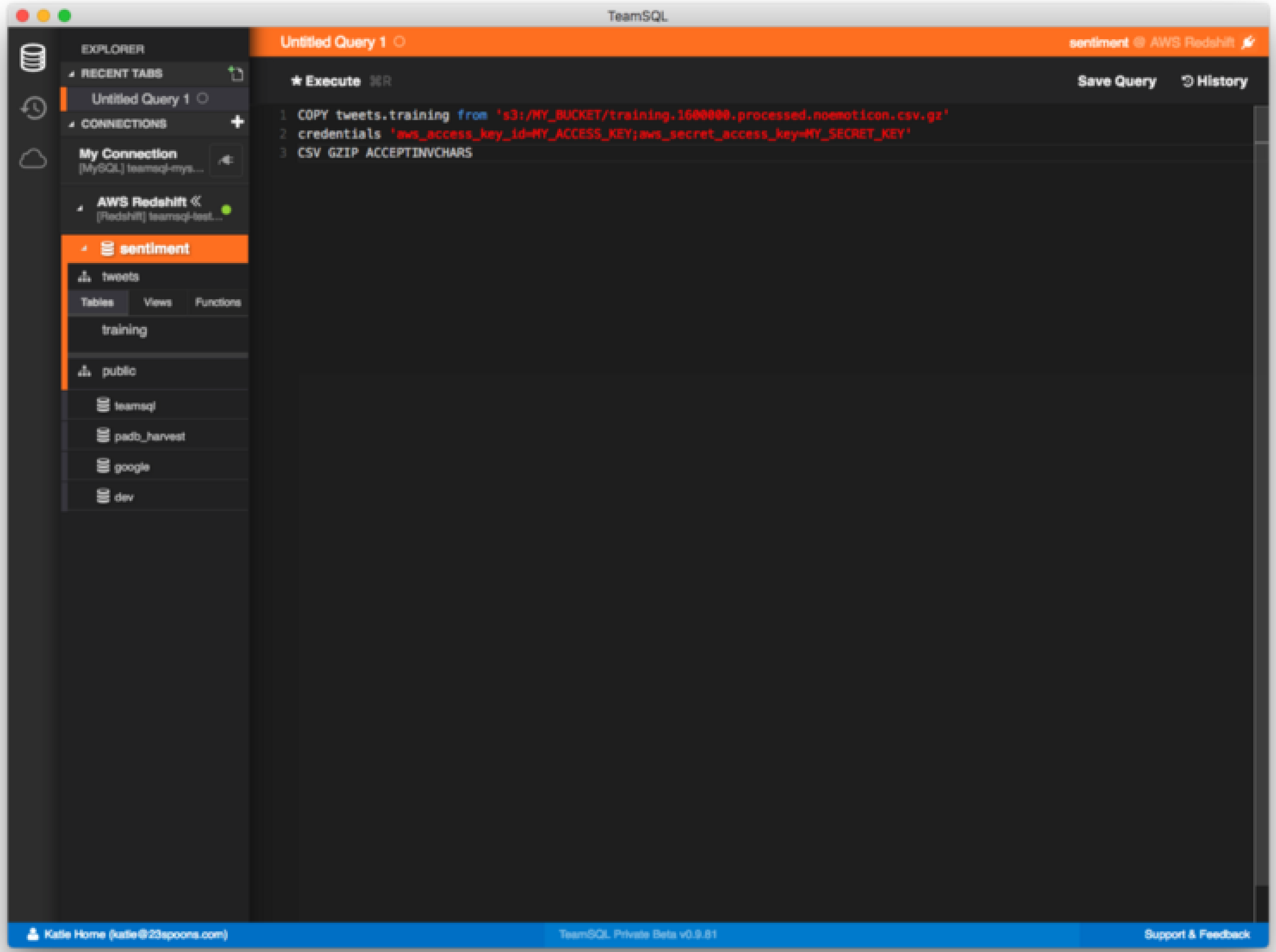
```sql
CREATE TABLE tweets.training (
    polarity int,
    id int,
    date_of_tweet varchar,
    query varchar,
    user_id varchar,
    tweet varchar
)
```

Using the COPY Command to Import Data

To copy your data from your source file to your data table, run the following command:

```
COPY tweets.training from 's3://MY_BUCKET/training.1600000.processed.noemoticon.csv.gz'
credentials 'aws_access_key_id=MY_ACCESS_KEY;aws_secret_access_key=MY_SECRET_KEY'
CSV GZIP ACCEPTINVCHARS
```

This command loads the CSV file and imports the data to our `tweets.training` table.



```
COPY tweets.training from 's3:/MY_BUCKET/training.1600000.processed.noemoticon.csv.gz'
credentials 'aws_access_key_id=MY_ACCESS_KEY;aws_secret_access_key=MY_SECRET_KEY'
CSV GZIP ACCEPTINVCHARS
```

Command Parameter Definitions

CSV

Enables use of CSV format in the input data.

DELIMITER

Specifies the single ASCII character that is used to separate fields in the input file, such as a pipe character (|), a comma (,), or a tab (\t).

GZIP

A value that specifies that the input file or files are in compressed gzip format (.gz files). The COPY operation reads each compressed file and uncompresses the data as it loads.

ACCEPTINVCHARS

Enables loading of data into VARCHAR columns even if the data contains invalid UTF-8 characters. When ACCEPTINVCHARS is specified, COPY replaces each invalid UTF-8 character with a string of equal length consisting of the character specified by*replacement_char*. For example, if the replacement character is '^', an invalid three-byte character will be replaced with '^^^'.

The replacement character can be any ASCII character except NULL. The default is a question mark (?). For information about invalid UTF-8 characters, see Multibyte Character Load Errors.

COPY returns the number of rows that contained invalid UTF-8 characters, and it adds an entry to the STL_REPLACEMENTS system table for each affected row, up to a maximum of 100 rows for each node slice. Additional invalid UTF-8 characters are also replaced, but those replacement events are not recorded.

If ACCEPTINVCHARS is not specified, COPY returns an error whenever it encounters an invalid UTF-8 character.

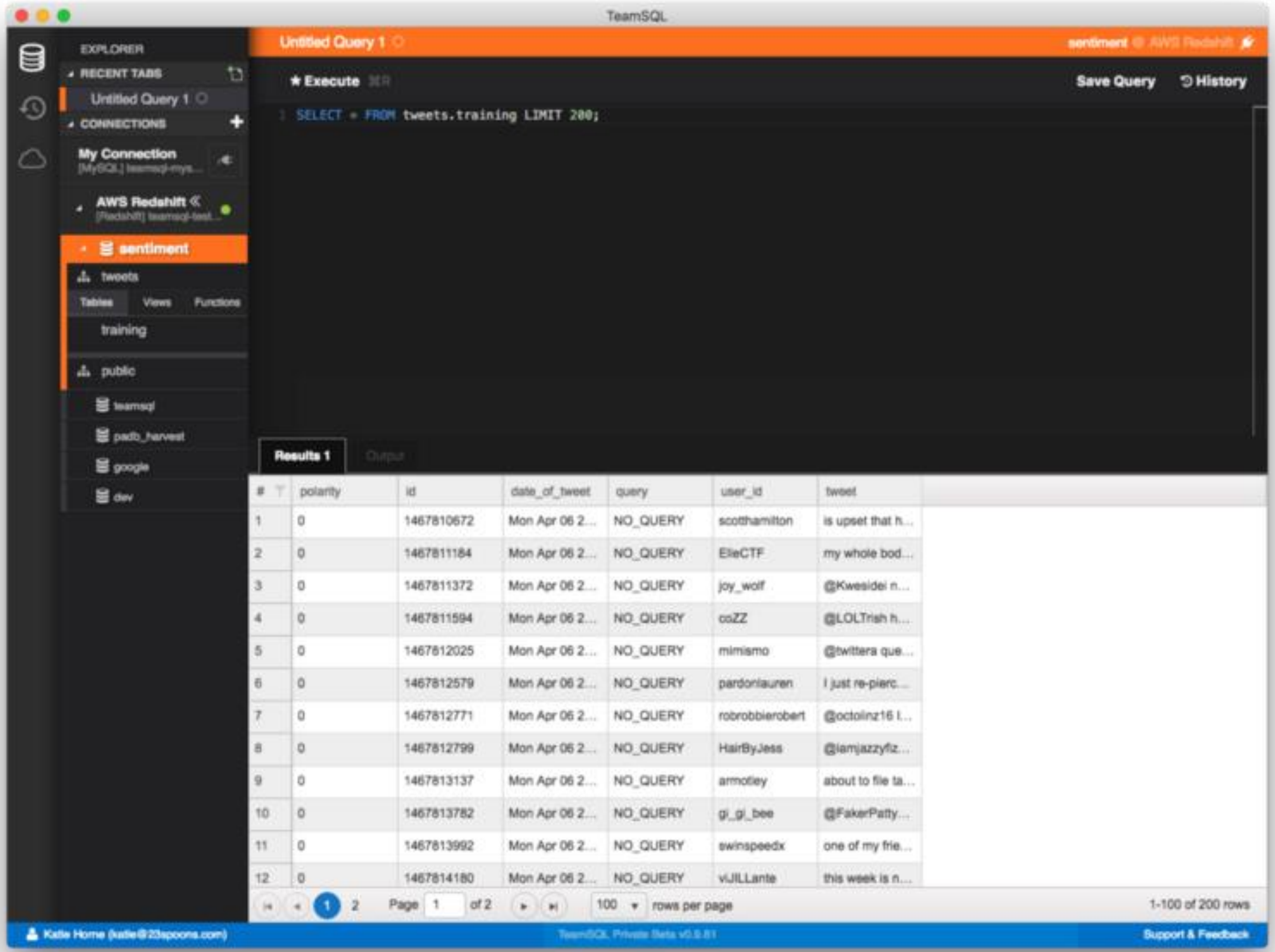
ACCEPTINVCHARS is valid only for VARCHAR columns.

For additional information, please see Redshift Copy Parameters and Data Format.

Accessing Imported Data

After your COPY process has finished, run a SELECT query to see if everything imported properly:

```
SELECT * FROM tweets.training LIMIT 200;
```



```
SELECT * FROM tweets.training LIMIT 200;
```

#	polarity	id	date_of_tweet	query	user_id	tweet
1	0	1467810672	Mon Apr 06 2...	NO_QUERY	scotthamilton	is upset that h...
2	0	1467811184	Mon Apr 06 2...	NO_QUERY	ElleCTF	my whole bod...
3	0	1467811372	Mon Apr 06 2...	NO_QUERY	joy_wolf	@Kwesidei n...
4	0	1467811594	Mon Apr 06 2...	NO_QUERY	coZZ	@LOLTrish h...
5	0	1467812025	Mon Apr 06 2...	NO_QUERY	mimismo	@twittera que...
6	0	1467812579	Mon Apr 06 2...	NO_QUERY	pardonlauren	I just re-pierc...
7	0	1467812771	Mon Apr 06 2...	NO_QUERY	robrobbierobert	@octolinz16 I...
8	0	1467812799	Mon Apr 06 2...	NO_QUERY	HairByJess	@iamjazzyfiz...
9	0	1467813137	Mon Apr 06 2...	NO_QUERY	armotley	about to file ta...
10	0	1467813782	Mon Apr 06 2...	NO_QUERY	gi_gi_bee	@FakerPatty...
11	0	1467813992	Mon Apr 06 2...	NO_QUERY	swinspeedx	one of my frie...
12	0	1467814180	Mon Apr 06 2...	NO_QUERY	viJILLante	this week is n...

1 2 Page 1 of 2 100 rows per page 1-100 of 200 rows

Troubleshooting

If you get an error while executing the COPY command, you can check the Redshift logs by running the following:

```
SELECT * FROM stl_load_errors;
```

You can download TeamSQL for free: https://teamsql.io/

- AWS
- Redshift
- S3