Loading Data into PostgreSQL

Loading data into PostgreSQL can be a difficult process, but it is extremely important that you learn how to do it. You have already learned how to learn data using the INSERT command with tuples

```
INSERT INTO <tableName>
(field1, field2, field3) VALUES (item1, 'item2', item3), (item1, 'item2',
item3), (item1, 'item2', item3)…;
```

However, this is not the most efficient way to load data into the system. The \COPY command is actually the command you wan to use when loading data.

The approach you should use would be to create the table and use the **"\COPY"** command

```
CREATE TABLE iowa_liquor_sales
(
  invoice_number text,
  invoice_date date,
  store_number text,
  store_name text,
  address text,
  city text,
  zip_code text,
  store_location text,
  county_number integer,
  county text,
  category text,
  category_name text,
  vendor_number bigint,
  vendor_name text,
  item_number bigint,
  item_description text,
  pack integer,
  bottle_volume_ml double precision,
  state_bottle_cost money,
  state_bottle_retail money,
  bottles_sold bigint,
  sale money,
  volume_sold_liters double precision,
  volume_sold_gallons double precision
);

\COPY iowa_liquor_sales FROM 'C:\git\GIS5577_spring_2018\GIS5577_lab2\iowa_liquor_sales.csv' WITH CSV HEADER;
```

Unfortunately this creates an error. This is a pgAdmin Error. It occurs in both pgAdmin3 and pgAdmin4

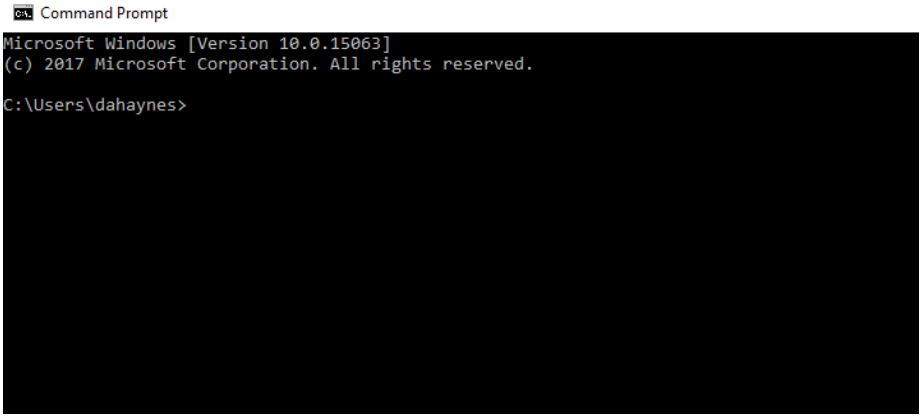
```
ERROR:  syntax error at or near "\"
LINE 1: \COPY iowa_liquor_sales FROM 'C:\git\GIS5577_spring_2018\GIS...
        ^
********** Error **********

ERROR: syntax error at or near "\"
SQL state: 42601
Character: 1
```

Let's simply the error to idea that pgAdmin doesn't care for "\". A solution might be to drop the **"\"** and use **"COPY"** but your accounts (non-superusers) don't have access to that command.

Copying Data

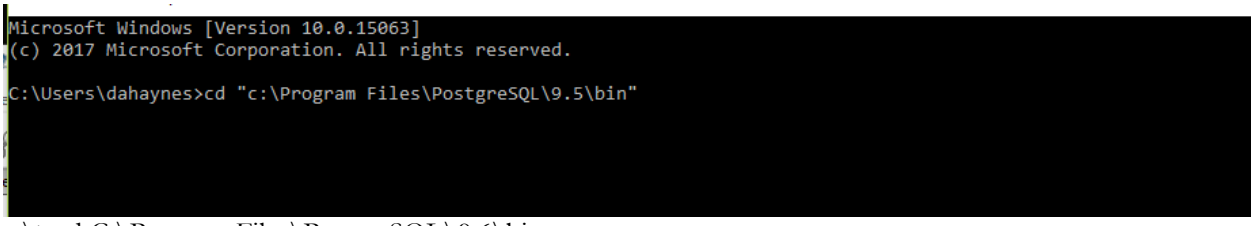
What we need to do is connect to the database and issue the **"\COPY"** command without getting a syntax error. To do that you need to connect to the database. Using the terminal.

1. First open a windows prompt, by going to the start menu and typing `"cmd"`



2. Next open find the psql.exe file. For windows users it is usally located in Program Files **"C:\Program Files\PostgreSQL\9.6\bin"** --The version <9.6> depends upon what you installed

In the command prompt change the directory by typing `"cd"`. `"cd"` is a command that means change directory. After telling the computer to change the directory you need to tell it the directory that is should be looking at. In our case it is "C:\Program Files\PostgreSQL\9.6\bin"



c:\>cd C:\Program Files\PostgreSQL\9.6\bin

3. You should be able to type `"psql --help"`

```
C:\Users\dahaynes>cd "c:\Program Files\PostgreSQL\9.5\bin"

c:\Program Files\PostgreSQL\9.5\bin>psql --help
psql is the PostgreSQL interactive terminal.

Usage:
  psql [OPTION]... [DBNAME [USERNAME]]

General options:
  -c, --command=COMMAND    run only single command (SQL or internal) and exit
  -d, --dbname=DBNAME      database name to connect to (default: "dahaynes")
  -f, --file=FILENAME      execute commands from file, then exit
  -l, --list               list available databases, then exit
  -v, --set=, --variable=NAME=VALUE
                           set psql variable NAME to VALUE
                           (e.g., -v ON_ERROR_STOP=1)
  -V, --version            output version information, then exit
  -X, --no-psqlrc          do not read startup file (~/.psqlrc)
  -1 ("one"), --single-transaction
                           execute as a single transaction (if non-interactive)
  -?, --help[=options]     show this help, then exit
      --help=commands      list backslash commands, then exit
      --help=variables     list special variables, then exit
```

Psql is postgresql's actual terminal for communicating with the database. Throughout the course I will show you some of the options contained within the terminal. You can use this as a way of connecting to a database. Similar to connecting with pgAdmin4.

4. To connect to the server you will need to specify the following
    a. host address
    b. username
    c. database

PSQL uses **"flags"** to identify these. **"-h"** is for host **"-U"** is for username and **"-d"** is for database

```
c:\Program Files\PostgreSQL\9.5\bin>psql -h 149.165.170.114 -U david -d classroom
psql (9.5.14, server 9.6.11)
```

5. Now you can issue your "\COPY" command.

\COPY iowa_liquor_sales FROM 'C:\git\GIS5577_spring_2018\GIS5577_lab2\iowa_liquor_sales.csv' WITH CSV HEADER;

```
classroom=# \COPY iowa_liquor_sales FROM 'C:\git\GIS5577_spring_2018\GIS5577_lab2\iowa_liquor_sales.csv' WITH CSV HEADER
;
```

6. It is important to have the ";" so psql knows when the command ends
7. Hit the enter key to run the command. It takes about 4 minutes for the data to load

```
classroom=# \COPY iowa_liquor_sales FROM 'C:\git\GIS5577_spring_2018\GIS5577_lab2\iowa_liquor_sales.csv' WITH CSV HEADER
;
COPY 125817
classroom=#
```

8. When the psql is finished loading the data prompt <database># will be available for you to type
9. You can run other SQL commands here to verify the data is loaded.

```
classroom=# SELECT count(1) from iowa_liquor_sales;
 count
--------
 125817
(1 row)
```

10. Type "\q" to quit psql. You should see the new table in pgAdmin4 and be able to finish you assignment.

One-step Connect&Copy

There is a way to combine steps 4 & 5 into a single step. You do this by issuing a 4th flag. I don't recommend that you try this until you have successfully connected to the database and issued the copy command.

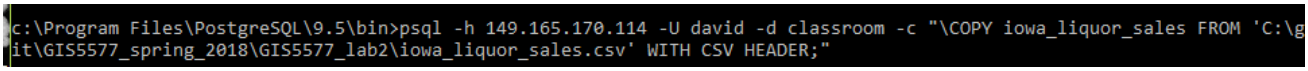
1. To connect to the server you will need to specify the following
    a. "-h" host address
    b. "-U" username
    c. "-d" database
    d. "-c" command

```
>psql -h 149.165.170.114 -U <username> -d <database> -c "\COPY ……"
```

2. To load the data simply add the "-c" command for copying the data into the table. The command for copying the data must be enclosed in double quotes ""

```
"\COPY <tablename> FROM 'csvPath' WITH CSV HEADER;
```

-- The "\" is important as are the double quotes surrounding the statement "", single quotes are used inside for specifying the csvpath. csvPath must be the complete file path including the file extension.




```
C:\Program Files\PostgreSQL\9.6\bin>psql -h 149.165.170.114 -U david -d david
-c "COPY iowa_liquor_sales FROM
'C:\work\GIS5577\GIS5577_lab2\iowa_liquor_sales.csv' WITH CSV HEADER;"
```