**Instructions**

Follow the instructions and get all the user stories below to pass to finish the project.

You start with several files, one of them is games.csv. It contains a comma-separated list of all games of the final three rounds of the World Cup tournament since 2014; the titles are at the top. It includes the year of each game, the round of the game, the winner, their opponent, and the number of goals each team scored. You need to do three things for this project:

Part 1: Create the database

Log into the psql interactive terminal with psql --username=freecodecamp --dbname=postgres and create your database structure according to the user stories below.

**Don't forget to connect to the database after you create it**.

Part 2: Insert the data

Complete the insert\_data.sh script to correctly insert all the data from games.csv into the database. The file is started for you. Do not modify any of the code you start with. Using the PSQL variable defined, you can make database queries like this: $($PSQL "<query\_here>"). The tests have a 20 second limit, so try to make your script efficient. The less you have to query the database, the faster it will be. You can empty the rows in the tables of your database with TRUNCATE TABLE games, teams;

Part 3: Query the database

Complete the empty echo commands in the queries.sh file to produce output that matches the expected\_output.txt file. The file has some starter code, and the first query is completed for you. Use the PSQL variable defined to complete rest of the queries. Note that you need to have your database filled with the correct data from the script to get the correct results from your queries. Hint: Test your queries in the psql prompt first and then add them to the script file.

**Notes:**  
If you leave your virtual machine, your database may not be saved. You can make a dump of it by entering pg\_dump -cC --inserts -U freecodecamp worldcup > worldcup.sql in a bash terminal (not the psql one). It will save the commands to rebuild your database in worldcup.sql. The file will be located where the command was entered. If it's anything inside the project folder, the file will be saved in the VM. You can rebuild the database by entering psql -U postgres < worldcup.sql in a terminal where the .sql file is.

If you are saving your progress on [freeCodeCamp.org](http://freecodecamp.org/), after getting all the tests to pass, follow the instructions above to save a dump of your database. Save the worldcup.sql file, as well as the final version of your insert\_data.sh and queries.sh files, in a public repository and submit the URL to it on [freeCodeCamp.org](http://freecodecamp.org/).

Complete the tasks below

* You should create a database named worldcup
* You should **connect to your worldcup database** and then create teams and games tables
* Your teams table should have a team\_id column that is a type of SERIAL and is the primary key, and a name column that has to be UNIQUE
* Your games table should have a game\_id column that is a type of SERIAL and is the primary key, a year column of type INT, and a round column of type VARCHAR
* Your games table should have winner\_id and opponent\_id foreign key columns that each reference team\_id from the teams table
* Your games table should have winner\_goals and opponent\_goals columns that are type INT
* All of your columns should have the NOT NULL constraint
* Your two script (.sh) files should have executable permissions. Other tests involving these two files will fail until permissions are correct. When these permissions are enabled, the tests will take significantly longer to run
* When you run your insert\_data.sh script, it should add each unique team to the teams table. There should be 24 rows
* When you run your insert\_data.sh script, it should insert a row for each line in the games.csv file (other than the top line of the file). There should be 32 rows. Each row should have every column filled in with the appropriate info. Make sure to add the correct ID's from the teams table (you cannot hard-code the values)
* You should correctly complete the queries in the queries.sh file. Fill in each empty echo command to get the output of what is suggested with the command above it. Only use a single line like the first query. The output should match what is in the expected\_output.txt file **exactly**, take note of the number of decimal places in some of the query results

**Solutions:**

Part 1: Create the database

connect the serve: psql --username=freecodecamp --dbname=postgres

create a new database called “worldcup”: CREATE DATABASE worldcup;

connect this database:\c worldcup

create table:

CREATE TABLE games();

CREATE TABLE teams();

add columns:

-- teams

ALTER TABLE teams ADD COLUMN team\_id SERIAL PRIMARY KEY;

or:

ALTER TABLE teams ADD COLUMN team\_id SERIAL;

ALTER TABLE teams ADD CONSTRAINT teams\_pkey PRIMARY KEY(team\_id);

(add primary key constraint to the column)

-

ALTER TABLE teams ADD COLUMN name text NOT NULL;

ALTER TABLE teams ADD CONSTRAINT name\_unique UNIQUE(name);

-- games

ALTER TABLE games ADD COLUMN game\_id SERIAL PRIMARY KEY;

ALTER TABLE games ADD COLUMN year INT NOT NULL;

ALTER TABLE games ADD COLUMN round VARCHAR NOT NULL;

add foreign key column:

ALTER TABLE games ADD COLUMN winner\_id INT NOT NULL;

ALTER TABLE games ADD COLUMN opponent\_id INT NOT NULL;

ALTER table games ADD FOREIGN KEY(winner\_id) REFERENCES teams(team\_id);

ALTER table games ADD FOREIGN KEY(opponent\_id) REFERENCES teams(team\_id);

add two more columns:

ALTER TABLE games ADD COLUMN winner\_goals INT NOT NULL;

ALTER TABLE games ADD COLUMN opponent\_goals INT NOT NULL;

Part 2: Insert the data

chmod +x insert\_data.sh

chmod +x queries.sh

Part 3: Query the database

create table:

create table:

create table:

post-notes:

It is fine that a table has two foreign keys that reference the same primary key of another table

how to run sql.file in visual studio code:

\*\* in terminal, inside the folder

1, run sql file:

psql -f worldcup.sql

2, connect to psql server and check if it works out

psql --username=yang --dbname=yang

\c worldcup

\d