Instructions

For this project, you need to log in to PostgreSQL with psql to create your database. Do that by entering psql --username=freecodecamp --dbname=postgres in the terminal. Make all the tests below pass to complete the project. Be sure to get creative, and have fun!

Don't forget to connect to your database after you create it 😄

Here's some ideas for other column and table names: description, has\_life, is\_spherical, age\_in\_millions\_of\_years, planet\_types, galaxy\_types, distance\_from\_earth.

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 universe > universe.sql in a bash terminal (not the psql one). It will save the commands to rebuild your database in universe.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 < universe.sql in a terminal where the .sql file is.

If you are saving your progress on freeCodeCamp.org, after getting all the tests to pass, follow the instructions above to save a dump of your database. Save the universe.sql file in a public repository and submit the URL to it on freeCodeCamp.org.

Complete the tasks below

* ~~You should create a database named universe~~
* ~~Be sure to connect to your database with \c universe. Then, you should add tables named galaxy, star, planet, and moon~~
* ~~Each table should have a primary key~~
* ~~Each primary key should automatically increment~~
* ~~Each table should have a name column~~
* ~~You should use the INT data type for at least two columns that are not a primary or foreign key~~
* ~~You should use the NUMERIC data type at least once~~
* ~~You should use the TEXT data type at least once~~
* ~~You should use the BOOLEAN data type on at least two columns~~
* ~~Each "star" should have a foreign key that references one of the rows in galaxy~~
* ~~Each "planet" should have a foreign key that references one of the rows in star~~
* ~~Each "moon" should have a foreign key that references one of the rows in planet~~
* ~~Your database should have at least five tables~~
* ~~Each table should have at least three rows~~
* ~~The galaxy and star tables should each have at least six rows~~
* ~~The planet table should have at least 12 rows~~
* ~~The moon table should have at least 20 rows~~
* ~~Each table should have at least three columns~~
* ~~The galaxy, star, planet, and moon tables should each have at least five columns~~
* ~~At least two columns per table should not accept NULL values~~
* ~~At least one column from each table should be required to be UNIQUE~~
* ~~All columns named name should be of type VARCHAR~~
* ~~Each primary key column should follow the naming convention table\_name\_id. For example, the moon table should have a primary key column named moon\_id~~
* ~~Each foreign key column should have the same name as the column it is referencing~~

-----------------------------------------------------------------------------------------------------------------

psql --username=freecodecamp --dbname=postgres

CREATE DATABASE universe;

\c universe;

CREATE TABLE galaxy(

galaxy\_id SERIAL UNIQUE NOT NULL PRIMARY KEY,

name VARCHAR(30) NOT NULL,

shape TEXT,

diameter\_in\_million\_light\_year NUMERIC(5,2),

age\_in\_millions\_of\_year NUMERIC(5,2),

UNIQUE(galaxy\_id,name)

);

CREATE TABLE star(

star\_id SERIAL UNIQUE NOT NULL PRIMARY KEY,

name VARCHAR(30) NOT NULL,

diameter\_in\_million\_km INT,

color TEXT,

galaxy\_id INT REFERENCES galaxy(galaxy\_id),

UNIQUE(star\_id,name)

);

CREATE TABLE planet(

planet\_id SERIAL UNIQUE NOT NULL PRIMARY KEY,

name VARCHAR(30) NOT NULL,

has\_life BOOLEAN,

diameter\_in\_thousand\_km INT,

star\_id INT REFERENCES star(star\_id),

UNIQUE(planet\_id,name)

);

CREATE TABLE moon(

moon\_id SERIAL UNIQUE NOT NULL PRIMARY KEY,

name VARCHAR(30) NOT NULL,

diameter\_in\_km INT,

is\_spherical BOOLEAN,

planet\_id INT REFERENCES planet(planet\_id),

UNIQUE(moon\_id,name)

);

CREATE TABLE black\_hole(

black\_hole\_id SERIAL UNIQUE NOT NULL PRIMARY KEY,

name VARCHAR(30) NOT NULL,

galaxy\_id INT REFERENCES galaxy(galaxy\_id),

UNIQUE(black\_hole\_id,name)

);

INSERT INTO galaxy(name, shape, diameter\_in\_million\_light\_year, age\_in\_millions\_of\_year) VALUES('Galaxy1', 'Spiral',1.0,1.0), ('Galaxy2', 'Elips',2.0,2.0), ('Galaxy3', 'Lenticular',3.0,3.0), ('Galaxy4', 'Spiral',4.0,4.0), ('Galaxy5', 'Irregular',5.0,5.0), ('Galaxy6', 'Spiral',6.0,6.0);

INSERT INTO star(name, diameter\_in\_million\_km, color, galaxy\_id) VALUES('star1',1,'yellow',1), ('star2',2,'blue',2), ('star3',3,'red',3), ('star4',4,'yellow',4), ('star5',5,'yellow',5), ('star6',6,'yellow',6);

INSERT INTO planet(name,has\_life, diameter\_in\_thousand\_km, star\_id) VALUES('planet1',FALSE,1,1), ('planet2',FALSE,2,2), ('planet3',FALSE,3,3), ('planet4',TRUE,4,4), ('planet5',FALSE,5,5), ('planet6',FALSE,6,6), ('planet7',FALSE,1,1), ('planet8',FALSE,1,3), ('planet9',FALSE,1,2), ('planet10',FALSE,1,4), ('planet11',FALSE,1,5), ('planet12',FALSE,1,6);

INSERT INTO moon(name, diameter\_in\_km, is\_spherical, planet\_id) VALUES('moon1',1,TRUE,1),

('moon2',2,TRUE,2), ('moon3',3,TRUE,3), ('moon4',4,TRUE,4), ('moon5',5,TRUE,5), ('moon6',6,TRUE,6), ('moon7',1,TRUE,7), ('moon8',1,TRUE,8), ('moon9',1,TRUE,9), ('moon10',1,TRUE,10), ('moon11',1,TRUE,11), ('moon12',1,TRUE,12), ('moon13',1,TRUE,12), ('moon14',1,TRUE,11), ('moon15',1,TRUE,10), ('moon16',1,TRUE,9), ('moon17',1,TRUE,8), ('moon18',1,TRUE,12), ('moon19',1,TRUE,1), ('moon20',1,FALSE,5);

INSERT INTO black\_hole(name, galaxy\_id) VALUES(‘black hole 1’, 1), (‘black hole 2’, 3), (‘black hole 3’, 5);

pg\_dump -cC --inserts -U freecodecamp universe > universe.sql

psql -U postgres < universe.sql