Postgres 4

Build the Movie Database

- https://pragprog.com/titles/pwrdata/seven-databases-in-seven-weeks-second-edition/
-

Build the Movie Database : create_movies.sql

```
CREATE TABLE genres (
	name text UNIQUE,
	position integer
);

CREATE TABLE movies (
	movie_id SERIAL PRIMARY KEY,
	title text,
	genre cube
);

CREATE TABLE actors (
	actor_id SERIAL PRIMARY KEY,
	name text
);

CREATE TABLE movies_actors (
	movie_id integer REFERENCES movies NOT NULL,
	actor_id integer REFERENCES actors NOT NULL,
	UNIQUE (movie_id, actor_id)
);

CREATE INDEX movies_actors_movie_id ON movies_actors (movie_id);
CREATE INDEX movies_actors_actor_id ON movies_actors (actor_id);
CREATE INDEX movies_genres_cube ON movies USING gist (genre);
```

Build the Movie Database: movies_data.sql

```
INSERT INTO genres (name,position) VALUES
('Action',1),
('Adventure',2),
(‘Animation',3),
…

INSERT INTO movies (movie_id,title,genre) VALUES
(1,'Star Wars','(0,7,0,0,0,0,0,0,0,7,0,0,0,0,10,0,0,0)'),
(2,'Forrest Gump','(0,0,0,5,0,0,0,7,0,0,0,0,0,0,0,0,0,0)'),
(3,'American Beauty','(0,0,0,0,0,0,0,7,0,0,0,0,0,0,0,0,0,0)'),
(4,'Citizen Kane’,'(0,0,0,0,0,0,0,5,0,0,0,0,0,0,0,0,0,0)'),
…
```

Build the Movie Database: movies_data.sql

```
INSERT INTO actors (actor_id,name) VALUES
(1,'50 Cent'),
(2,'A Martinez'),
(3,'A. Michael Baldwin'),
(4,'Aaron Eckhart'),
(5,'Aaron Paul'),
…

INSERT INTO movies_actors (movie_id,actor_id) VALUES
(1,3165),
(1,644),
(1,1753),
(1,3768),
(2,4666),
…
```

Build the Movie Database

```
$ psql -h localhost -U postgres -p 5433
Password for user postgres:
psql (13.1)
Type "help" for help.

postgres=# create database movies;
CREATE DATABASE
postgres=# \list
                                List of databases
       Name       |  Owner   | Encoding | Collate | Ctype |   Access privileges
------------------+----------+----------+---------+-------+-----------------------
 cs464            | postgres | UTF8     | C       | C     |
 movies           | postgres | UTF8     | C       | C     |
 mydb             | cs464    | UTF8     | C       | C     |
 postgres         | postgres | UTF8     | C       | C     |
 template0        | postgres | UTF8     | C       | C     | =c/postgres          +
                  |          |          |         |       | postgres=CTc/postgres
 template1        | postgres | UTF8     | C       | C     | =c/postgres          +
                  |          |          |         |       | postgres=CTc/postgres
 template_postgis | postgres | UTF8     | C       | C     |
(7 rows)

postgres=# \c movies
You are now connected to database "movies" as user "postgres"
```

Build Movie Database

```
movies=# create extension cube;
CREATE EXTENSION

movies=# \cd /Users/gendreau/classes/spring21/cs464/codeSeven/code/postgres
movies=# \! pwd /Users/gendreau/classes/spring21/cs464/codeSeven/code/postgres

movies=# \i create_movies.sql
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE INDEX
CREATE INDEX
CREATE INDEX
```

Build Movie Database

```
movies=# \d
                 List of relations
 Schema |        Name         |   Type   |  Owner
--------+---------------------+----------+----------
 public | actors              | table    | postgres
 public | actors_actor_id_seq | sequence | postgres
 public | genres              | table    | postgres
 public | movies              | table    | postgres
 public | movies_actors       | table    | postgres
 public | movies_movie_id_seq | sequence | postgres
(6 rows)
```

Build Movie Database

```
movies=# \d movies
                                Table "public.movies"
  Column  |  Type   | Collation | Nullable |                 Default
----------+---------+-----------+----------+------------------------------------------
 movie_id | integer |           | not null | nextval('movies_movie_id_seq'::regclass)
 title    | text    |           |          |
 genre    | cube    |           |          |
Indexes:
    "movies_pkey" PRIMARY KEY, btree (movie_id)
    "movies_genres_cube" gist (genre)
Referenced by:
    TABLE "movies_actors" CONSTRAINT "movies_actors_movie_id_fkey" FOREIGN KEY (movie_id) REFERENCES
movies(movie_id)

movies=# \i movies_data.sql
INSERT 0 18
INSERT 0 2861
INSERT 0 4986
INSERT 0 11168
```