

```
createdb moviesdb_test
psql -d moviesdb_test -a -f jtang45-code.sql
```

```
/* database creation */
CREATE TABLE Ratings(
userid numeric,
movieid numeric,
rating double precision,
timestamp numeric);
CREATE TABLE
CREATE TABLE Tags(
userid numeric,
movieid numeric,
tag text,
timestamp numeric);
CREATE TABLE
CREATE TABLE Movies(
movieid numeric PRIMARY KEY,
title text,
year numeric);
CREATE TABLE
CREATE TABLE Users(
userid numeric PRIMARY KEY);
CREATE TABLE
CREATE TABLE genres(
title text);
CREATE TABLE
CREATE TABLE has_genre(
movieid numeric,
genre text);
CREATE TABLE
/* Relations creation

ALTER TABLE ratings add constraint uc foreign
key(userid) references users(userid);

*/
/* Relations data population */
```

```

\copy ratings from 'movies/ratings.txt'
(DELIMITER(":"));
COPY 10000054
\copy tags from 'movies/tags.csv' (DELIMITER(":"));
COPY 95580
\copy movies from 'movies/movies.csv' (DELIMITER(":"));
COPY 10681
\copy genres from 'movies/genres.txt'
COPY 19
\copy has_genre from 'movies/has_genre.txt'
(DELIMITER(":"));
COPY 21564
/*reads the users from the ratings table and adds them
to the users table.*/
insert into users(userid) select distinct userid from
ratings;
INSERT 0 69878
/*reads the *remaining* users from the tags and adds
them to the users table.*/
insert into users(userid) select distinct userid from
tags where userid not in(select distinct userid from
ratings);
INSERT 0 1689
/* Relations creation*/
ALTER TABLE ratings add constraint uc foreign
key(userid) references users(userid);
ALTER TABLE
/* List your tables*/
\dt

```

```

          List of relations
 Schema |   Name   | Type  | Owner
-----+-----+-----+-----
 public | genres   | table | jennytang
 public | has_genre | table | jennytang
 public | movies   | table | jennytang
 public | ratings  | table | jennytang
 public | tags     | table | jennytang
 public | users    | table | jennytang
(6 rows)

```

/*Data types of your tables.*/

\d genres

Table "public.genres"

Column	Type	Collation	Nullable	Default
title	text			

\d movies

Table "public.movies"

Column	Type	Collation	Nullable	Default
movieid	numeric		not null	
title	text			
year	numeric			

Indexes:

"movies_pkey" PRIMARY KEY, btree (movieid)

\d ratings

Table "public.ratings"

Column	Type	Collation	Nullable	Default
userid	numeric			
movieid	numeric			
rating	double precision			
timestamp	numeric			

Foreign-key constraints:

"uc" FOREIGN KEY (userid) REFERENCES users(userid)

\d tags

Table "public.tags"

Column	Type	Collation	Nullable	Default
userid	numeric			
movieid	numeric			
tag	text			
timestamp	numeric			

```
\d users
```

Table "public.users"				
Column	Type	Collation	Nullable	Default
userid	numeric		not null	

```
Indexes:
```

```
    "users_pkey" PRIMARY KEY, btree (userid)
```

```
Referenced by:
```

```
    TABLE "ratings" CONSTRAINT "uc" FOREIGN KEY  
(userid) REFERENCES users(userid)
```

```
\d has_genre
```

Table "public.has_genre"				
Column	Type	Collation	Nullable	Default
movieid	numeric			
genre	text			

```
/*Sizes of your tables*/
```

```
select count(*) from genres;  
count
```

```
-----  
      19  
(1 row)
```

```
select count(*) from movies;  
count
```

```
-----  
  10681  
(1 row)
```

```
select count(*) from ratings;  
count
```

```
-----  
10000054  
(1 row)
```

```
select count(*) from tags;
```

```
count
-----
95580
(1 row)
```

```
select count(*) from users;
count
-----
71567
(1 row)
```

```
select count(*) from has_genre;
count
-----
21564
(1 row)
```

```
/* Data values */
select * from movies limit 5;
movieid | title | year
-----+-----+-----
1 | Toy Story | 1995
2 | Jumanji | 1995
3 | Grumpier Old Men | 1995
4 | Waiting to Exhale | 1995
5 | Father of the Bride Part II | 1995
(5 rows)
```

```
select count(title) from movies;
count
-----
10681
(1 row)
```

```
select * from movies order by year desc limit 5;
movieid | title | year
-----+-----+-----
-+-----
```

```

    55830 | Be Kind Rewind
| 2008
    56949 | 27 Dresses
| 2008
    53207 | 88 Minutes
| 2008
    55603 | My Mom's New Boyfriend
| 2008
    57326 | In the Name of the King A Dungeon Siege Tale
| 2008
(5 rows)

```

```

select * from movies order by year limit 5;
movieid | title | year
-----+-----+-----
    7065 | Birth of a Nation, The | 1915
    7243 | Intolerance | 1916
   62383 | 20,000 Leagues Under the Sea | 1916
   48374 | Father Sergius (Otets Sergiy) | 1917
    8511 | Immigrant, The | 1917
(5 rows)

```

```

select * from movies order by year limit 5;
movieid | title | year
-----+-----+-----
    7065 | Birth of a Nation, The | 1915
    7243 | Intolerance | 1916
   62383 | 20,000 Leagues Under the Sea | 1916
   48374 | Father Sergius (Otets Sergiy) | 1917
    8511 | Immigrant, The | 1917
(5 rows)

```

```

select count(year) from movies;
count
-----
   10681
(1 row)

```

```

select count(year) from movies where year = 0;

```

```

count
-----
      0
(1 row)

```

```

select count(year) from movies where year > 1500;
count
-----
10681
(1 row)

```

```

Select movieid from has_genre where genre is null;
movieid
-----
(0 rows)

```

```

/*Find unknown or invalid data in any of the attributes
for all of the tables, movies, ratings, tags, users,
genres.*/
select genre, count(genre) from has_genre where genre
not in (select title from genres) group by genre;
      genre          | count
-----+-----
(no genres listed) |      1
(1 row)

```

```

/*Find the distribution of the values for attribute
"year" of table "movies".*/
select year, count(year) from movies group by year
order by year asc;
year | count
-----+-----
1915 |      1
1916 |      2
1917 |      2
1918 |      2
1919 |      4
1920 |      5

```

1921		3
1922		7
1923		6
1924		6
1925		10
1926		10
1927		19
1928		10
1929		7
1930		15
1931		16
1932		22
1933		23
1934		18
1935		18
1936		32
1937		30
1938		19
1939		37
1940		40
1941		28
1942		38
1943		40
1944		37
1945		36
1946		38
1947		39
1948		46
1949		37
1950		44
1951		44
1952		40
1953		55
1954		43
1955		57
1956		53
1957		62
1958		62
1959		61

1960		66
1961		57
1962		69
1963		63
1964		72
1965		72
1966		87
1967		68
1968		72
1969		64
1970		71
1971		73
1972		83
1973		81
1974		75
1975		74
1976		75
1977		83
1978		82
1979		87
1980		161
1981		178
1982		170
1983		111
1984		137
1985		158
1986		166
1987		205
1988		214
1989		212
1990		200
1991		188
1992		212
1993		258
1994		307
1995		362
1996		384
1997		370
1998		384

1999		357
2000		405
2001		403
2002		441
2003		366
2004		342
2005		332
2006		345
2007		364
2008		251

(94 rows)

/*Find the distribution of the movies across different decades.*/

select decade, count(*) as count from (select
floor(year/10) * 10 as decade from movies) as t group
by decade order by decade asc;

decade		count
-----+-----		
1910		11
1920		83
1930		230
1940		379
1950		521
1960		690
1970		784
1980		1712
1990		3022
2000		3249

(10 rows)

/*Find the distribution of the genres across the movies.*/

select genre, count(genre) from has_genre where genre
in (select title from genres) group by genre;

genre		count
-----+-----		
Thriller		1706
Romance		1685

Film-Noir		148
Adventure		1025
Musical		436
Comedy		3703
Animation		286
Western		275
Crime		1118
Fantasy		543
Horror		1013
Drama		5339
Sci-Fi		754
War		511
Documentary		482
Mystery		509
Action		1473
IMAX		29
Children		528

(19 rows)

```
/*Find the distribution of the ratings values (how many
movies were rated with 5, how many with 4, etc.).*/
select rating, count(rating) from ratings group by
rating order by rating asc;
```

rating		count
0.5		94988
1		384180
1.5		118278
2		790306
2.5		370178
3		2356676
3.5		879764
4		2875850
4.5		585022
5		1544812

(10 rows)

```

/*Find how many movies have no tags, but they have
ratings*/
select count(distinct(r.movieid)) from ratings r left
join tags t on r.movieid = t.movieid where t.movieid is
null;
count
-----
    3080
(1 row)

```

```

/*no ratings, but they have tags*/
select count(distinct(t.movieid)) from ratings r right
join tags t on r.movieid = t.movieid where r.movieid is
null;
count
-----
        4
(1 row)

```

```

/*no tags and no ratings*/
select count(distinct(t.movieid)) from ratings r full
outer join tags t on r.movieid = t.movieid where
t.movieid not in (select movieid from movies);
count
-----
        0
(1 row)

```

```

/* both tags and ratings*/
select count(distinct(t.movieid)) from ratings r inner
join tags t on r.movieid = t.movieid;
count
-----
    7597
(1 row)

```