Zeba Khan Rafi
Harie Vashini Dhamodharasamy Kalpana

CS486/586 Introduction to Databases
Fall 2020 Quarter

Domain:
Movies on Netflix, Prime Video, Hulu and Disney+
A collection of movies found on these platforms.

Table creation:
CREATE TABLE movies (Id INT PRIMARY KEY, movie TEXT NOT NULL, year INT, age TEXT, runtime INT);

CREATE TABLE ratings (Movie_id INT, imdb real, rotten_tomatoes INT, FOREIGN KEY(movie_id) REFERENCES movies(Id));

CREATE TABLE language (Id INT PRIMARY KEY, language TEXT);

CREATE TABLE languagerel (Movie_id INT, Lang_id INT, FOREIGN KEY(Movie_id) REFERENCES movies(Id), FOREIGN KEY(lang_id) REFERENCES language(Id));

CREATE TABLE genre (Id INT PRIMARY KEY, genre TEXT);

CREATE TABLE genrerel (Movie_id INT, genre_id INT, FOREIGN KEY(Movie_id) REFERENCES movies(Id), FOREIGN KEY(genre_id) REFERENCES genre(Id));

CREATE TABLE platforms (Id INT PRIMARY KEY, platform TEXT);

CREATE TABLE platformrel (Movie_id INT, Platform_id INT, FOREIGN KEY(Movie_id) REFERENCES movies(Id), FOREIGN KEY(Platform_id) REFERENCES platforms(Id));

CREATE TABLE country (Id INT PRIMARY KEY, country TEXT);

CREATE TABLE countryrel (Movie_id INT, country_id INT, FOREIGN KEY(Movie_id) REFERENCES movies(Id), FOREIGN KEY(country_id) REFERENCES country(Id));

CREATE TABLE directors (Id INT PRIMARY KEY, director TEXT);

CREATE TABLE directorrel (Movie_id INT, director_id INT, FOREIGN KEY(Movie_id) REFERENCES movies(Id), FOREIGN KEY(director_id) REFERENCES directors(Id));

<u>Data Cleaning and Data Insertion</u>:
Two sets of codes in particular were very helpful in data cleaning for some of our tables.

<u>Python code 1</u> :

```
import csv

list = [] #dictinct value of the column
File_Data = open('langu.csv', 'r') # To read our csv file data
Data = File_Data.readlines()
#used this for language, genre, director, country
for line in Data: # To segregate data
        item = line.split(",")
        for eachitem in item:
                eachitem = eachitem.replace('"','')
                eachitem = eachitem.replace('\n','')

                if eachitem in list :
                        continue

                else : #Add distinct values to the list
                        if len(eachitem) > 0  :
                                list.append(eachitem)
                                #print(eachitem)
                                #print(len(list))
i = 1
#used this for languagerel, genrerel, directorrel, countryrel
for line in Data : # To segregate data

        item = line.split(",")
        for eachitem in item:
                eachitem = eachitem.replace('"','')
                eachitem = eachitem.replace('\n','')
                if len(eachitem)==0:
                        itemid = ""
                else:
                        itemid = list.index(eachitem) + 1
                #print(i, itemid)

        i +=1
```

Python code 2 :

```
import csv

#used this for platformrel
with open("allplatforms.csv", 'r') as File_Data : # To read our csv file data
        reader = csv.DictReader(File_Data) # To organize data easy for segregation
        i=1
        for row in reader: # To segregate data
                id = row['ID']
                net = row['netflix']
                hulu = row['Hulu']
                prime = row['Prime Video']
                Disney = row['Disney+']
                net = net.replace('\n','')
                net=int(net)
                if(net==1):
                        print(id,1) #for netflix the platform id is 1
                hulu = hulu.replace('\n','')
                hulu=int(hulu)
                if(hulu==1):
                        print(id,2) #for hulu the platform id is 2
                prime = prime.replace('\n','')
                prime=int(prime)
                if(prime==1):
                        print(id,3) #for prime the platform id is 3
                Disney = Disney.replace('\n','')
                Disney=int(Disney)
                if(Disney==1):
                        print(id,4) #for Disney+ the platform id is 4
```

Data inserted into the table using “Import csv file” @ Postgres site - f20tdb19,f20tdb45

Description of how the database was populated:

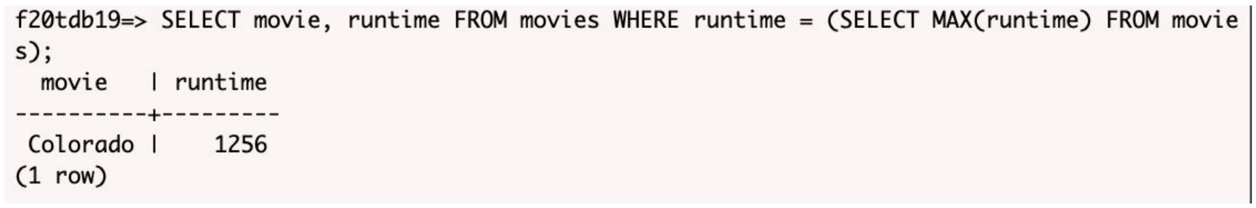
Created table structures in the database along with the key constraints. Individual respective columns were extracted and saved as different csv files using the above Python codes. The data was then populated into our SQL database tables using the import command.

Queries:

1. **List the movie with the longest duration.**

   SELECT movie, runtime FROM movies WHERE runtime = (SELECT MAX(runtime) FROM movies);

   1 row

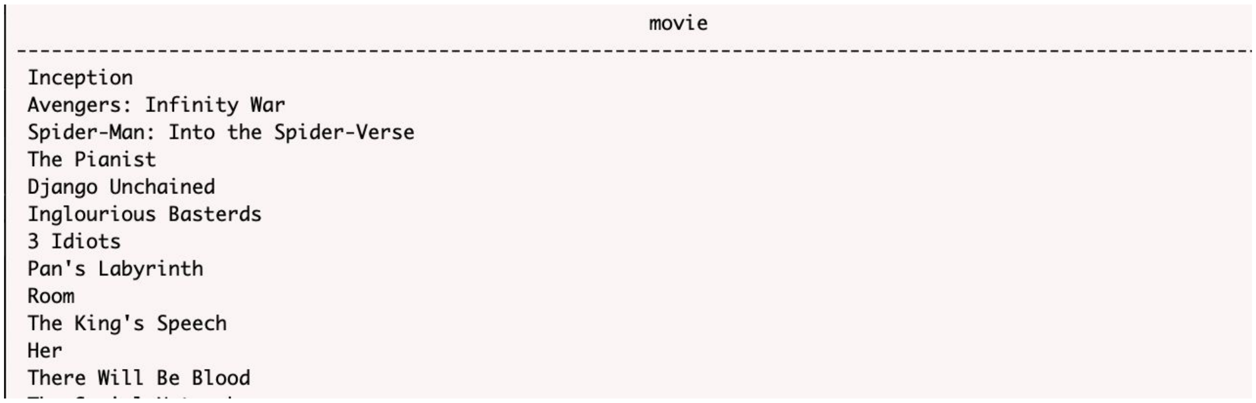


```
f20tdb19=> SELECT movie, runtime FROM movies WHERE runtime = (SELECT MAX(runtime) FROM movie
s);
  movie   | runtime
----------+---------
 Colorado |    1256
(1 row)
```

2. **List movies released since the year 2001 or in the 21st century.**

   SELECT movie FROM movies WHERE year BETWEEN 2001 AND 2100;

   12532 rows

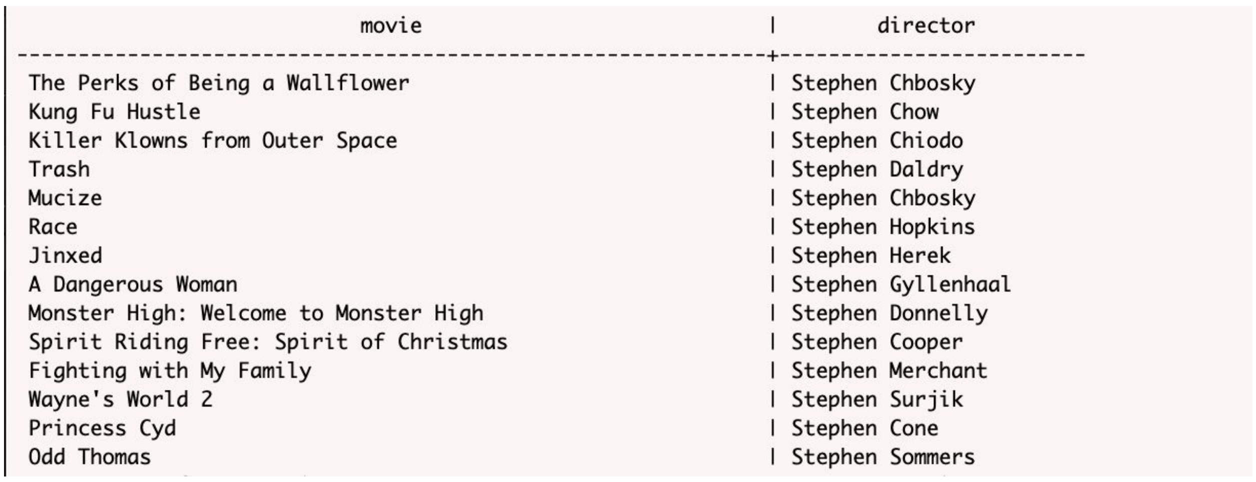


```
                                                movie
-----------------------------------------------------------------------------------------------------
 Inception
 Avengers: Infinity War
 Spider-Man: Into the Spider-Verse
 The Pianist
 Django Unchained
 Inglourious Basterds
 3 Idiots
 Pan's Labyrinth
 Room
 The King's Speech
 Her
 There Will Be Blood
```

3. **List movies and their directors whose first name is ‘Stephen’.**

   SELECT M.movie, D.director FROM movies M, directorrel DR, directors D WHERE M.id=DR.movie_id AND DR.director_id=D.id AND D.director LIKE 'Stephen%';

   72 rows

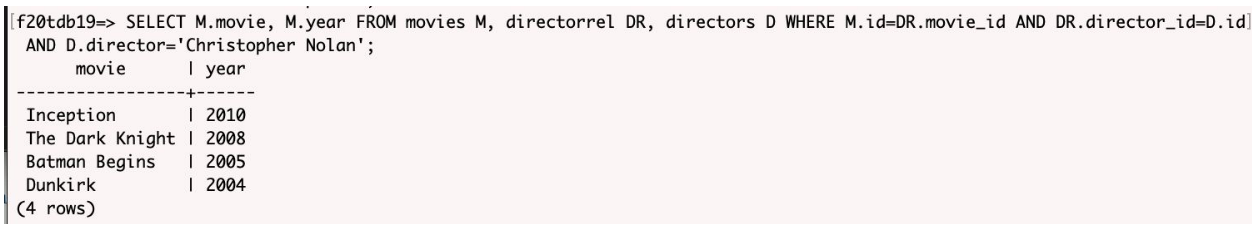


```
                            movie                              |        director
---------------------------------------------------------------+-------------------------
 The Perks of Being a Wallflower                               | Stephen Chbosky
 Kung Fu Hustle                                                | Stephen Chow
 Killer Klowns from Outer Space                                | Stephen Chiodo
 Trash                                                         | Stephen Daldry
 Mucize                                                        | Stephen Chbosky
 Race                                                          | Stephen Hopkins
 Jinxed                                                        | Stephen Herek
 A Dangerous Woman                                             | Stephen Gyllenhaal
 Monster High: Welcome to Monster High                         | Stephen Donnelly
 Spirit Riding Free: Spirit of Christmas                       | Stephen Cooper
 Fighting with My Family                                       | Stephen Merchant
 Wayne's World 2                                               | Stephen Surjik
 Princess Cyd                                                  | Stephen Cone
 Odd Thomas                                                    | Stephen Sommers
```

4. **List name of movies and their years of release by the director Christopher Nolan.**

   SELECT M.movie, M.year FROM movies M, directorrel DR, directors D WHERE M.id=DR.movie_id AND DR.director_id=D.id AND D.director='Christopher Nolan';

   4 rows

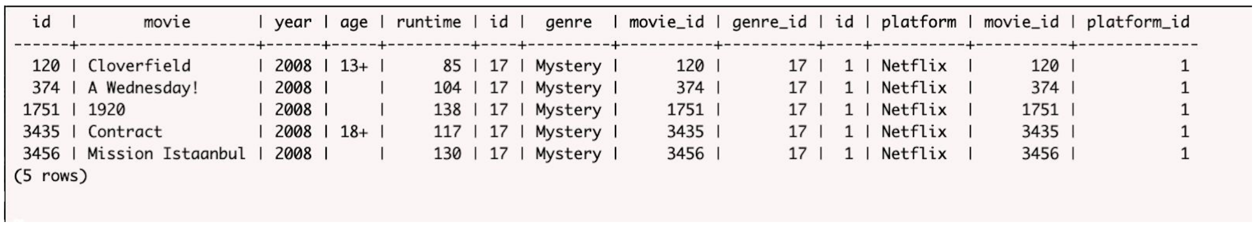


```
f20tdb19=> SELECT M.movie, M.year FROM movies M, directorrel DR, directors D WHERE M.id=DR.movie_id AND DR.director_id=D.id
 AND D.director='Christopher Nolan';
      movie      | year
-----------------+------
 Inception       | 2010
 The Dark Knight | 2008
 Batman Begins   | 2005
 Dunkirk         | 2004
(4 rows)
```

5. **List the details of mystery movies released in 2008 available only on Netflix.**

   SELECT * FROM movies M, genre G, genrerel GR, Platforms P, platformrel PR WHERE M.Id = GR.movie_id AND G.ID = GR.genre_id AND G.genre ='Mystery' AND M.Id=PR.movie_id AND P.Id = PR.platform_id AND P.Id = 1 AND M.year = 2008;

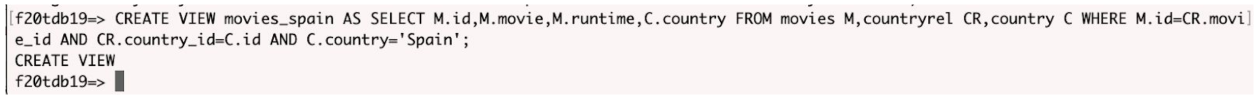
   5 rows



```
  id  |       movie       | year | age | runtime | id |  genre  | movie_id | genre_id | id | platform | movie_id | platform_id
------+-------------------+------+-----+---------+----+---------+----------+----------+----+----------+----------+-------------
  120 | Cloverfield       | 2008 | 13+ |      85 | 17 | Mystery |      120 |       17 |  1 | Netflix  |      120 |           1
  374 | A Wednesday!      | 2008 |     |     104 | 17 | Mystery |      374 |       17 |  1 | Netflix  |      374 |           1
 1751 | 1920              | 2008 |     |     138 | 17 | Mystery |     1751 |       17 |  1 | Netflix  |     1751 |           1
 3435 | Contract          | 2008 | 18+ |     117 | 17 | Mystery |     3435 |       17 |  1 | Netflix  |     3435 |           1
 3456 | Mission Istaanbul | 2008 |     |     130 | 17 | Mystery |     3456 |       17 |  1 | Netflix  |     3456 |           1
(5 rows)
```

6. **To display the genre of the movie with shorter duration and country of origin, Spain.**

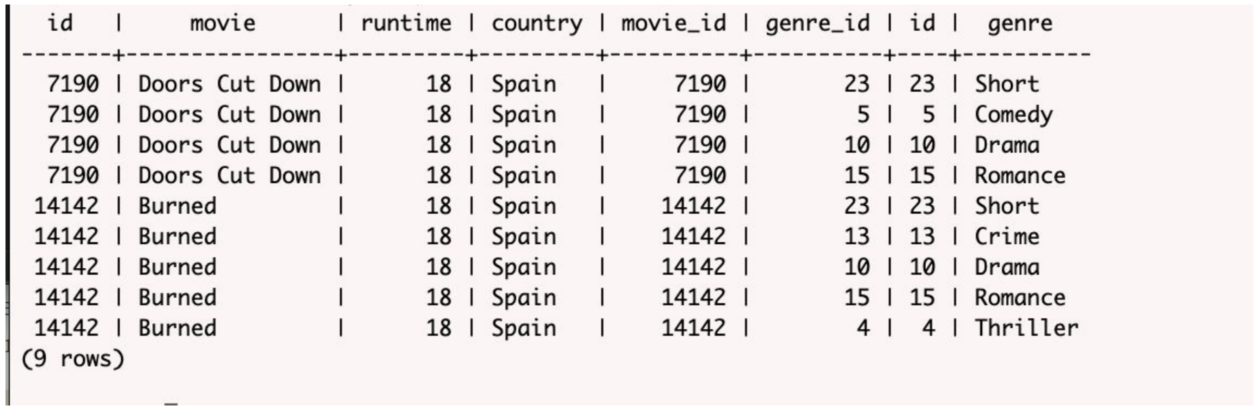
CREATE VIEW movies_spain AS SELECT M.id,M.movie,M.runtime,C.country FROM movies M,countryrel CR,country C WHERE M.id=CR.movie_id AND CR.country_id=C.id AND C.country='Spain';



```
f20tdb19=> CREATE VIEW movies_spain AS SELECT M.id,M.movie,M.runtime,C.country FROM movies M,countryrel CR,country C WHERE M.id=CR.movi
e_id AND CR.country_id=C.id AND C.country='Spain';
CREATE VIEW
f20tdb19=>
```

SELECT * FROM movies_spain MS, genrerel GR, genre G WHERE MS.id=GR.movie_id AND GR.genre_id=G.id AND MS.runtime IN (SELECT MIN(runtime) FROM movies_spain);

9 rows

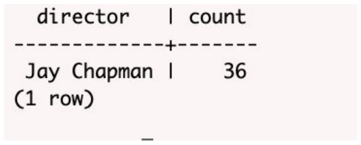


```
  id   |     movie      | runtime | country | movie_id | genre_id | id |  genre
-------+----------------+---------+---------+----------+----------+----+----------
  7190 | Doors Cut Down |      18 | Spain   |     7190 |       23 | 23 | Short
  7190 | Doors Cut Down |      18 | Spain   |     7190 |        5 |  5 | Comedy
  7190 | Doors Cut Down |      18 | Spain   |     7190 |       10 | 10 | Drama
  7190 | Doors Cut Down |      18 | Spain   |     7190 |       15 | 15 | Romance
 14142 | Burned         |      18 | Spain   |    14142 |       23 | 23 | Short
 14142 | Burned         |      18 | Spain   |    14142 |       13 | 13 | Crime
 14142 | Burned         |      18 | Spain   |    14142 |       10 | 10 | Drama
 14142 | Burned         |      18 | Spain   |    14142 |       15 | 15 | Romance
 14142 | Burned         |      18 | Spain   |    14142 |        4 |  4 | Thriller
(9 rows)
```

7. **List the name of Director with maximum number of movie releases.**

SELECT dt.* FROM (SELECT d.director, count(*) FROM directors d, directorrel dr WHERE d.id=dr.director_id GROUP BY d.id) AS dt ORDER BY dt.count DESC LIMIT 1;

1 row

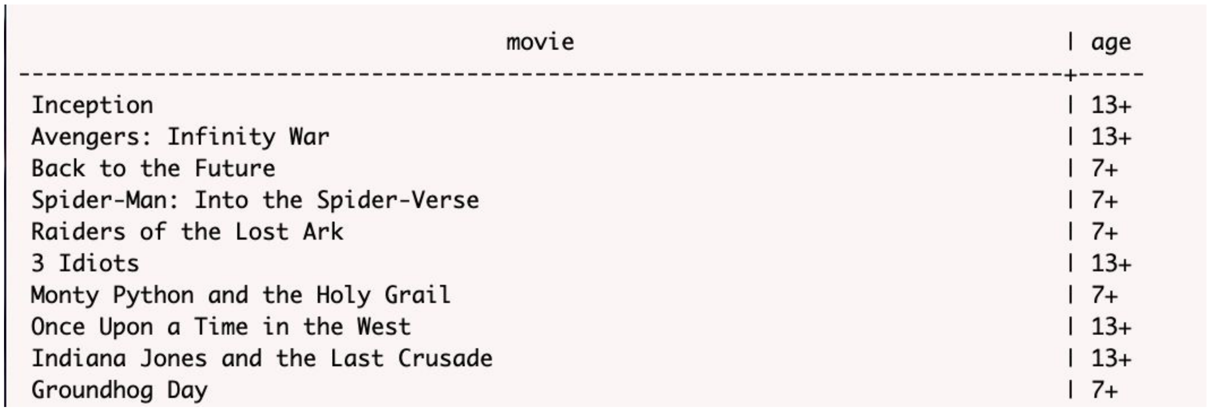


```
  director   | count
-------------+-------
 Jay Chapman |    36
(1 row)
```

8. List the movies that are made for a person who is 15 years old.

SELECT movie,age FROM movies WHERE age IN ('all','7+','13+');

3560 rows

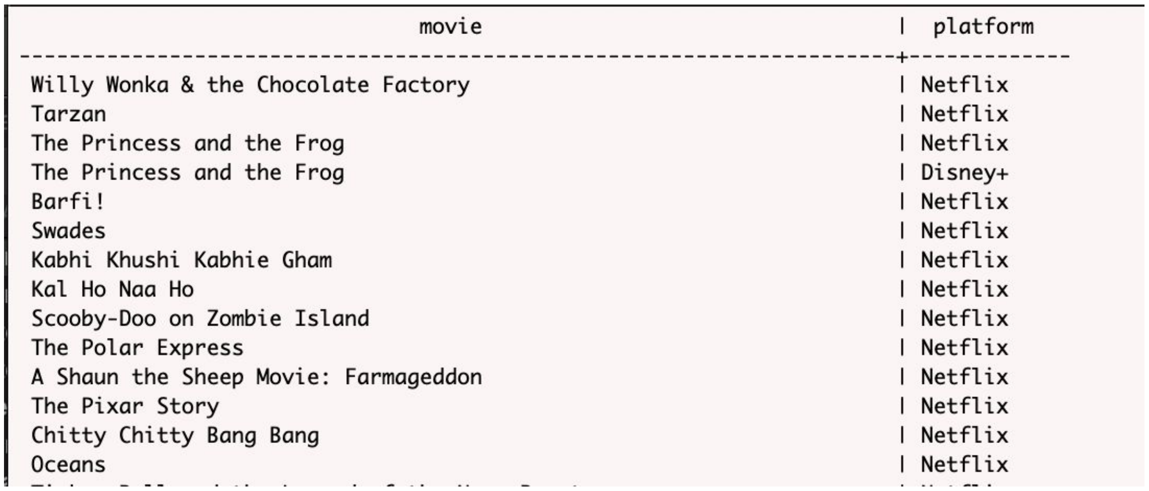


```
                              movie                                        | age
---------------------------------------------------------------------------+-----
 Inception                                                                 | 13+
 Avengers: Infinity War                                                    | 13+
 Back to the Future                                                        | 7+
 Spider-Man: Into the Spider-Verse                                         | 7+
 Raiders of the Lost Ark                                                   | 7+
 3 Idiots                                                                  | 13+
 Monty Python and the Holy Grail                                           | 7+
 Once Upon a Time in the West                                              | 13+
 Indiana Jones and the Last Crusade                                        | 13+
 Groundhog Day                                                             | 7+
```

9. List the platforms that have movies available for kids under 7 (age = 'all').

SELECT M.movie,P.platform FROM movies M, platformrel PR, platforms P WHERE M.id=PR.movie_id AND PR.platform_id=P.id AND M.age='all';

868 rows

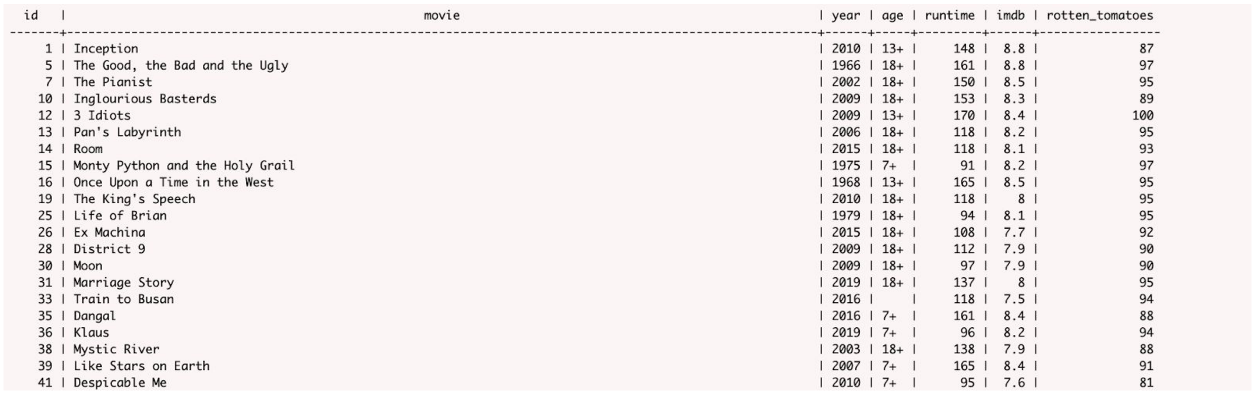


```
                              movie                                | platform
-------------------------------------------------------------------+-------------
 Willy Wonka & the Chocolate Factory                               | Netflix
 Tarzan                                                            | Netflix
 The Princess and the Frog                                         | Netflix
 The Princess and the Frog                                         | Disney+
 Barfi!                                                            | Netflix
 Swades                                                            | Netflix
 Kabhi Khushi Kabhie Gham                                          | Netflix
 Kal Ho Naa Ho                                                     | Netflix
 Scooby-Doo on Zombie Island                                       | Netflix
 The Polar Express                                                 | Netflix
 A Shaun the Sheep Movie: Farmageddon                              | Netflix
 The Pixar Story                                                   | Netflix
 Chitty Chitty Bang Bang                                           | Netflix
 Oceans                                                            | Netflix
```

10. List the details of the movies along with ratings that are released in other countries and not in the USA.

SELECT DISTINCT M.*,R.imdb,R.rotten_tomatoes FROM movies M JOIN ratings R ON M.Id = R.movie_id JOIN countryrel CR ON M.Id = CR.movie_id JOIN country C ON C.Id = CR.country_id AND C.country NOT IN (SELECT country FROM country WHERE country='United States') ORDER BY M.id;

7533 rows

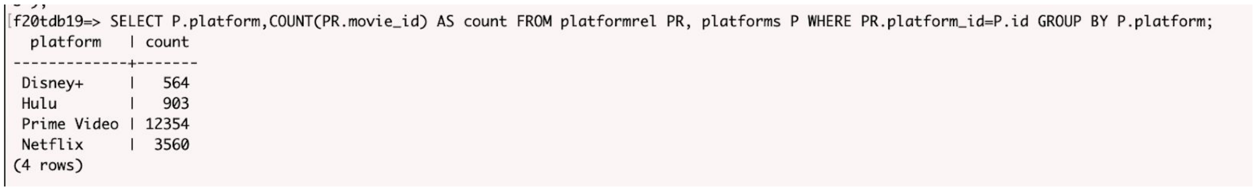


```
  id   |                  movie                   | year | age | runtime | imdb | rotten_tomatoes
-------+------------------------------------------+------+-----+---------+------+-----------------
     1 | Inception                                | 2010 | 13+ |     148 |  8.8 |              87
     5 | The Good, the Bad and the Ugly           | 1966 | 18+ |     161 |  8.8 |              97
     7 | The Pianist                              | 2002 | 18+ |     150 |  8.5 |              95
    10 | Inglourious Basterds                     | 2009 | 18+ |     153 |  8.3 |              89
    12 | 3 Idiots                                 | 2009 | 13+ |     170 |  8.4 |             100
    13 | Pan's Labyrinth                          | 2006 | 18+ |     118 |  8.2 |              95
    14 | Room                                     | 2015 | 18+ |     118 |  8.1 |              93
    15 | Monty Python and the Holy Grail          | 1975 | 7+  |      91 |  8.2 |              97
    16 | Once Upon a Time in the West             | 1968 | 13+ |     165 |  8.5 |              95
    19 | The King's Speech                        | 2010 | 18+ |     118 |    8 |              95
    25 | Life of Brian                            | 1979 | 18+ |      94 |  8.1 |              95
    26 | Ex Machina                               | 2015 | 18+ |     108 |  7.7 |              92
    28 | District 9                               | 2009 | 18+ |     112 |  7.9 |              90
    30 | Moon                                     | 2009 | 18+ |      97 |  7.9 |              90
    31 | Marriage Story                           | 2019 | 18+ |     137 |    8 |              95
    33 | Train to Busan                           | 2016 |     |     118 |  7.5 |              94
    35 | Dangal                                   | 2016 | 7+  |     161 |  8.4 |              88
    36 | Klaus                                    | 2019 | 7+  |      96 |  8.2 |              94
    38 | Mystic River                             | 2003 | 18+ |     138 |  7.9 |              88
    39 | Like Stars on Earth                      | 2007 | 7+  |     165 |  8.4 |              91
    41 | Despicable Me                            | 2010 | 7+  |      95 |  7.6 |              81
```

11. Count of movies that are available on all platforms (Netflix, Hulu, Prime, Disney+)

SELECT P.platform,COUNT(PR.movie_id) AS count FROM platformrel PR, platforms P WHERE PR.platform_id=P.id GROUP BY P.platform;

4 rows

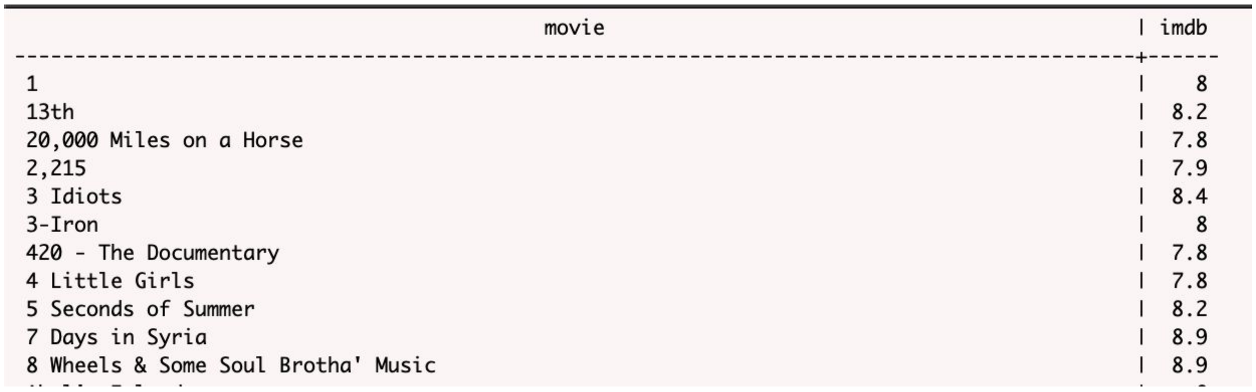


```
f20tdb19=> SELECT P.platform,COUNT(PR.movie_id) AS count FROM platformrel PR, platforms P WHERE PR.platform_id=P.id GROUP BY P.platform;
  platform   | count
-------------+-------
 Disney+     |   564
 Hulu        |   903
 Prime Video | 12354
 Netflix     |  3560
(4 rows)
```

12. List the best IMDb rated movies (above 7.8)

SELECT DISTINCT M.movie,R.imdb FROM movies M, ratings R WHERE R.movie_id in (SELECT M.id FROM movies M1 WHERE M.id=M1.id) AND R.imdb >7.8;

932 rows

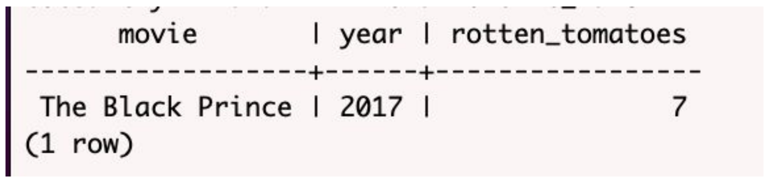


```
                                          movie                                           | imdb
------------------------------------------------------------------------------------------+------
 1                                                                                        |    8
 13th                                                                                     |  8.2
 20,000 Miles on a Horse                                                                  |  7.8
 2,215                                                                                    |  7.9
 3 Idiots                                                                                 |  8.4
 3-Iron                                                                                   |    8
 420 - The Documentary                                                                    |  7.8
 4 Little Girls                                                                           |  7.8
 5 Seconds of Summer                                                                      |  8.2
 7 Days in Syria                                                                          |  8.9
 8 Wheels & Some Soul Brotha' Music                                                       |  8.9
```

13. **List the year of the lowest rated (Rotten Tomatoes) Indian movies that were released.**

SELECT DISTINCT m.movie, m.year, r.rotten_tomatoes FROM movies m, country c, countryrel cr, ratings r WHERE m.id=cr.movie_id AND cr.country_id=c.id AND c.country='India' AND m.id=r.movie_id ORDER BY r.rotten_tomatoes ASC LIMIT 1;

1 row

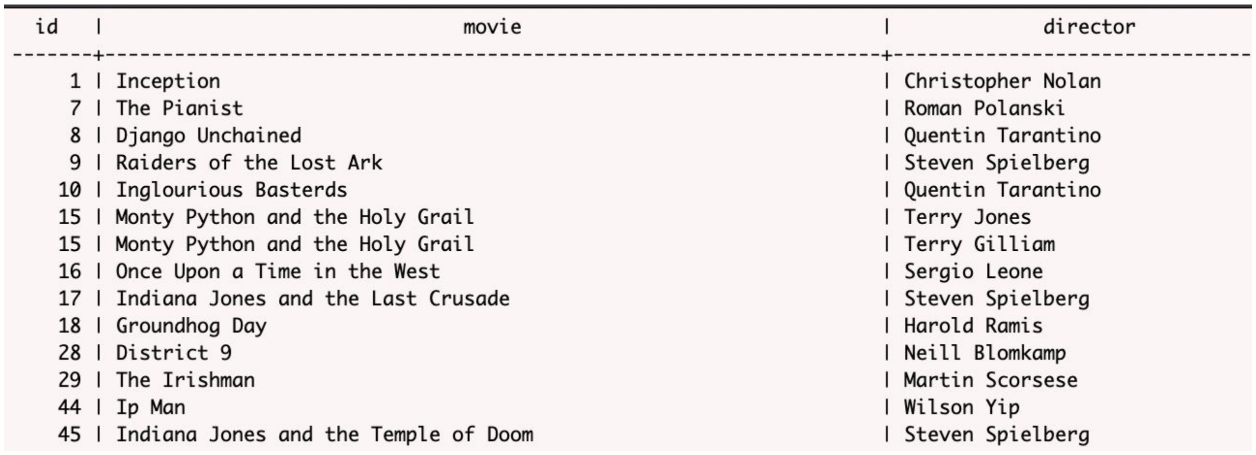


```
      movie       | year | rotten_tomatoes
------------------+------+-----------------
 The Black Prince | 2017 |               7
(1 row)
```

14. **List name of Directors and their movies released in more than 2 languages.**

SELECT m1.id,m1.movie,d.director FROM movies m1, directors d, directorrel dr WHERE m1.id=dr.movie_id AND dr.director_id=d.id AND m1.id IN (SELECT m.id from languagerel lr,movies m where m.id=m1.id AND m.id=lr.movie_id group by m.id having count(lr.lang_id)>2 ORDER BY count(*));

1109 rows

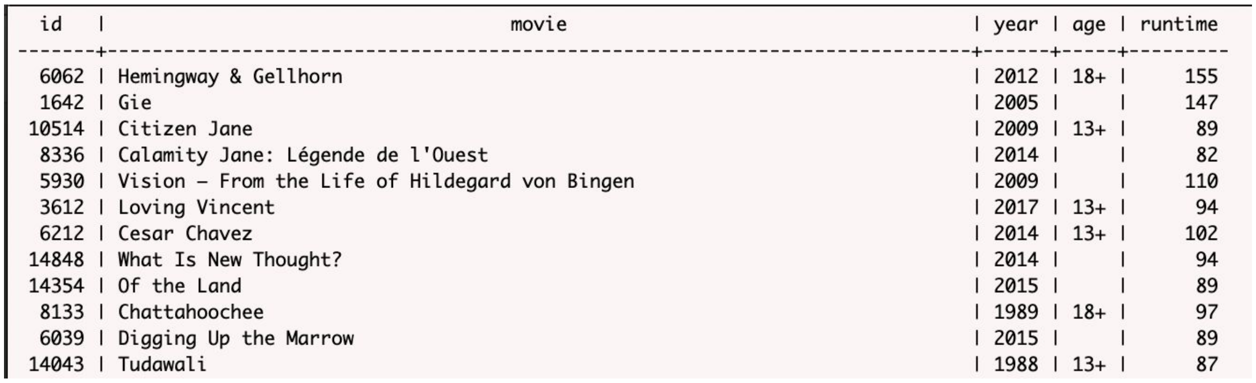


```
  id   |                             movie                              |           director
-------+----------------------------------------------------------------+------------------------------
     1 | Inception                                                      | Christopher Nolan
     7 | The Pianist                                                    | Roman Polanski
     8 | Django Unchained                                               | Quentin Tarantino
     9 | Raiders of the Lost Ark                                        | Steven Spielberg
    10 | Inglourious Basterds                                           | Quentin Tarantino
    15 | Monty Python and the Holy Grail                                | Terry Jones
    15 | Monty Python and the Holy Grail                                | Terry Gilliam
    16 | Once Upon a Time in the West                                   | Sergio Leone
    17 | Indiana Jones and the Last Crusade                             | Steven Spielberg
    18 | Groundhog Day                                                  | Harold Ramis
    28 | District 9                                                     | Neill Blomkamp
    29 | The Irishman                                                   | Martin Scorsese
    44 | Ip Man                                                         | Wilson Yip
    45 | Indiana Jones and the Temple of Doom                           | Steven Spielberg
```

15. **List of details of a movie of a Biography genre.**

SELECT * FROM movies WHERE id in (SELECT movie_id FROM genrerel WHERE genre_id in (SELECT id FROM genre WHERE genre='Biography'));

821 rows

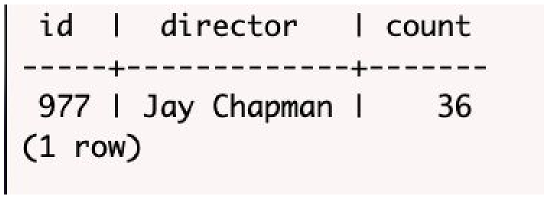


```
  id   |                                 movie                                  | year | age | runtime
-------+------------------------------------------------------------------------+------+-----+---------
  6062 | Hemingway & Gellhorn                                                   | 2012 | 18+ |     155
  1642 | Gie                                                                    | 2005 |     |     147
 10514 | Citizen Jane                                                           | 2009 | 13+ |      89
  8336 | Calamity Jane: Légende de l'Ouest                                      | 2014 |     |      82
  5930 | Vision - From the Life of Hildegard von Bingen                         | 2009 |     |     110
  3612 | Loving Vincent                                                         | 2017 | 13+ |      94
  6212 | Cesar Chavez                                                           | 2014 | 13+ |     102
 14848 | What Is New Thought?                                                   | 2014 |     |      94
 14354 | Of the Land                                                            | 2015 |     |      89
  8133 | Chattahoochee                                                          | 1989 | 18+ |      97
  6039 | Digging Up the Marrow                                                  | 2015 |     |      89
 14043 | Tudawali                                                               | 1988 | 13+ |      87
```

16. **List the most famous director who makes movies in the comical genre. Return the name and duration of the movie.**

SELECT d.id,d.director,count(*) FROM movies m1, directors d, directorrel dr WHERE m1.id=dr.movie_id AND dr.director_id=d.id AND m1.id IN (SELECT m.id FROM movies m, genre g, genrerel gr WHERE m.id=gr.movie_id AND gr.genre_id=g.id AND g.genre='Comedy') GROUP BY d.id ORDER BY count(*) DESC LIMIT 1;

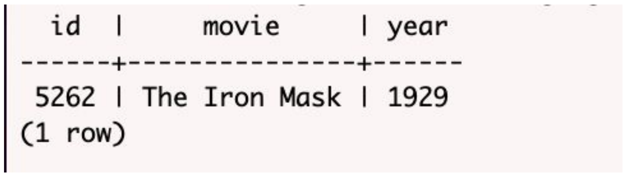
1 row


```
 id  |   director   | count
-----+-------------+-------
 977 | Jay Chapman |    36
(1 row)
```

17. **List the oldest Chinese Action movie.**

    SELECT m.id, m.movie, m.year FROM movies m, country c, countryrel cr, genrerel gr, genre g WHERE m.id=cr.movie_id AND cr.country_id=c.id AND c.country='China' AND m.id = gr.movie_id AND gr.genre_id = g.id AND g.genre = 'Action' ORDER BY m.year ASC LIMIT 1;

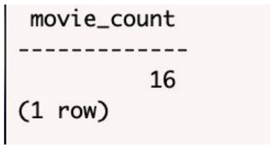
    1 row


```
  id  |     movie     | year
------+---------------+------
 5262 | The Iron Mask | 1929
(1 row)
```

18. **Count of best rated Romantic movies of the decade available on Amazon Prime. (imdb >7.8)**

    select count(distinct(m.id)) as movie_count from movies m, genre g, genrerel gr, platformrel pr, platforms p, ratings r where gr.movie_id=m.id and gr.genre_id=g.id and g.genre='Romance' AND pr.movie_id=m.id AND pr.platform_id=p.id AND p.platform='Prime Video' AND r.movie_id=m.id AND r.imdb>7.8 AND m.year BETWEEN 2011 AND 2020;

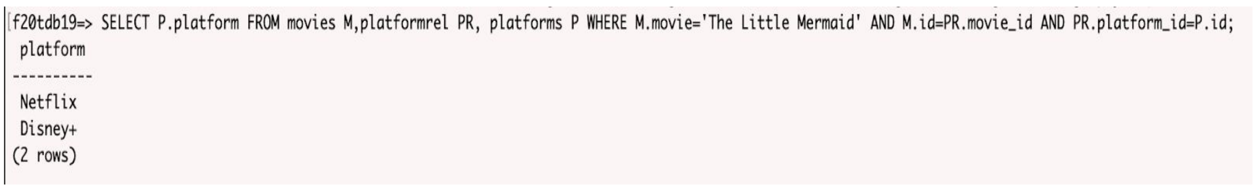
    1 row


```
 movie_count
-------------
          16
(1 row)
```

19. **List the availability of the movie "The Little Mermaid" on all the platforms.**

    SELECT P.platform FROM movies M,platformrel PR, platforms P WHERE M.movie='The Little Mermaid' AND M.id=PR.movie_id AND PR.platform_id=P.id;

2 rows

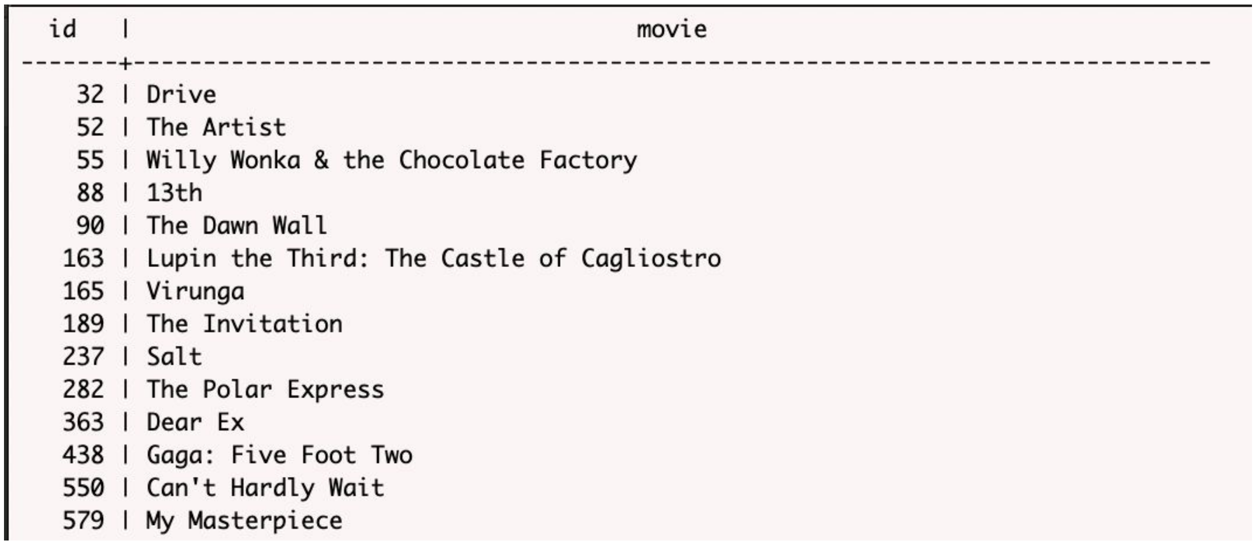


```
f20tdb19=> SELECT P.platform FROM movies M,platformrel PR, platforms P WHERE M.movie='The Little Mermaid' AND M.id=PR.movie_id AND PR.platform_id=P.id;
 platform
----------
 Netflix
 Disney+
(2 rows)
```

20. List the Id and titles of movies with runtime duration of 100 mins.

SELECT id, movie FROM movies WHERE runtime=100;

402 rows



```
  id   |                                  movie
-------+--------------------------------------------------------------------------
    32 | Drive
    52 | The Artist
    55 | Willy Wonka & the Chocolate Factory
    88 | 13th
    90 | The Dawn Wall
   163 | Lupin the Third: The Castle of Cagliostro
   165 | Virunga
   189 | The Invitation
   237 | Salt
   282 | The Polar Express
   363 | Dear Ex
   438 | Gaga: Five Foot Two
   550 | Can't Hardly Wait
   579 | My Masterpiece
```

Queries 8, 13 are modified. Query 8 uses a column named age which was initially predicted to take integer datatype but due to the data values we had to change it to text datatype. Query 13, here we were unable to query results based on both imdb and rotten_tomatoes ratings so we queried it using just rotten_tomatoes. Also we have changed the select statement parts of query 10, this is because some movies are released in more than one genres and languages.