HW - Movies - SQL Problems & Answers

1. In `1.sql`, write a SQL query to list the titles of all movies released in 2008.
   Your query should output a table with a single column for the title of each movie.

```
SELECT title
FROM movies
WHERE year = 2008;
```

2. In `2.sql`, write a SQL query to determine the birth year of Emma Stone.
   Your query should output a table with a single column and a single row (plus optional header) containing Emma Stone's birth year.
   You may assume that there is only one person in the database with the name Emma Stone.

```
SELECT birth
FROM people
WHERE name = "Emma Stone";
```

3. In `3.sql`, write a SQL query to list the titles of all movies with a release date on or after 2018, in alphabetical order.
   Your query should output a table with a single column for the title of each movie.
   Movies released in 2018 should be included, as should movies with release dates in the future.

```
SELECT title
FROM movies
WHERE year >= 2018
ORDER BY title;
```

4. In `4.sql`, write a SQL query to determine the number of movies with an IMDb rating of 10.0.
   Your query should output a table with a single column and a single row (plus optional header) containing the number of movies with a 10.0 rating.

```
SELECT COUNT(*)
FROM ratings
WHERE rating = 10;
```

5. In `5.sql`, write a SQL query to list the titles and release years of all Harry Potter movies, in chronological order.

   Your query should output a table with two columns, one for the title of each movie and one for the release year of each movie.

   You may assume that the title of all Harry Potter movies will begin with the words “Harry Potter”, and that if a movie title begins with the words “Harry Potter”, it is a Harry Potter movie.

```
SELECT year, title
FROM movies
WHERE title like "Harry Potter%"
ORDER BY year;
```

6. In `6.sql`, write a SQL query to determine the average rating of all movies released in 2012.

   Your query should output a table with a single column and a single row (plus optional header) containing the average rating.

```
SELECT AVG(rating)
FROM movies
JOIN ratings ON movies.id = ratings.movie_id
WHERE year = 2012;
```

7. In `7.sql`, write a SQL query to list all movies released in 2010 and their ratings, in descending order by rating. For movies with the same rating, order them alphabetically by title.

   Your query should output a table with two columns, one for the title of each movie and one for the rating of each movie.

   Movies that do not have ratings should not be included in the result.

```
SELECT title, rating
FROM movies
JOIN ratings ON movies.id = ratings.movie_id
WHERE year = 2010 AND rating NOT NULL
ORDER BY rating DESC, title ASC;
```

8. In `8.sql`, write a SQL query to list the names of all people who starred in Toy Story.

   Your query should output a table with a single column for the name of each person.

   You may assume that there is only one movie in the database with the title Toy Story.

```
SELECT name FROM movies
INNER JOIN stars ON movies.id = stars.movie_id
INNER JOIN people ON people.id = stars.person_id
WHERE title = "Toy Story";
```

9. In `9.sql`, write a SQL query to list the names of all people who starred in a movie released in 2004, ordered by birth year.

   Your query should output a table with a single column for the name of each person.

   People with the same birth year may be listed in any order.

   No need to worry about people who have no birth year listed, so long as those who do have a birth year are listed in order.

   If a person appeared in more than one movie in 2004, they should only appear in your results once.

```
SELECT DISTINCT name FROM movies
INNER JOIN stars ON movies.id = stars.movie_id
INNER JOIN people ON people.id = stars.person_id
WHERE year = 2004
ORDER BY birth;
```

10. In `10.sql`, write a SQL query to list the names of all people who have directed a movie that received a rating of at least 9.0.

    Your query should output a table with a single column for the name of each person.

```
SELECT name FROM movies
INNER JOIN directors ON movies.id = directors.movie_id
INNER JOIN people ON people.id = directors.person_id
INNER JOIN ratings ON movies.id = ratings.movie_id
WHERE rating >= 9;
```

11. In `11.sql`, write a SQL query to list the titles of the five highest rated movies (in order) that Chadwick Boseman starred in, starting with the highest rated.

    Your query should output a table with a single column for the title of each movie.

    You may assume that there is only one person in the database with the name Chadwick Boseman.

```sql
SELECT title FROM movies
INNER JOIN stars ON movies.id = stars.movie_id
INNER JOIN people ON people.id = stars.person_id
INNER JOIN ratings ON movies.id = ratings.movie_id
WHERE name = "Chadwick Boseman"
ORDER BY rating DESC
LIMIT 5;
```

12. In `12.sql`, write a SQL query to list the titles of all movies in which both Johnny Depp and Helena Bonham Carter starred.

    Your query should output a table with a single column for the title of each movie.

    You may assume that there is only one person in the database with the name Johnny Depp.

    You may assume that there is only one person in the database with the name Helena Bonham Carter.

```sql
SELECT DISTINCT title
FROM movies
INNER JOIN stars ON movies.id = stars.movie_id
INNER JOIN people ON people.id = stars.person_id
WHERE name  = 'Johnny Depp'
OR name = 'Helena Bonham Carter'
GROUP BY title
HAVING COUNT(distinct people.name) = 2;
```

13. In `13.sql`, write a SQL query to list the names of all people who starred in a movie in which Kevin Bacon also starred.

    Your query should output a table with a single column for the name of each person.

    There may be multiple people named Kevin Bacon in the database. Be sure to only select the Kevin Bacon born in 1958.

    Kevin Bacon himself should not be included in the resulting list.

```
SELECT DISTINCT name
FROM movies
INNER JOIN stars ON movies.id = stars.movie_id
INNER JOIN people ON people.id = stars.person_id
WHERE movie_id IN (SELECT DISTINCT movie_id
FROM movies
INNER JOIN stars ON movies.id = stars.movie_id
INNER JOIN people ON people.id = stars.person_id
WHERE name  = 'Kevin Bacon'
AND birth = '1958')
AND name != 'Kevin Bacon';
```