

Movie Ratings over Time¶

Here are some made up movie ratings for us to practice some SQL. This problem was made by Shengjia Guo and edited by the instructors. Wayne Lee assumes any mistakes and typos generated.

Here are the tables and their respective columns:

- rating
 - rid: reviewer id, int
 - mid: movie id, int
 - stars: stars given to a movie, 0 to 5, int
 - ratingdate: date the review was given, text (SQLite does not support Date datatypes natively)
- movie
 - mid: movie id, int
 - title: title of the movie, text
 - year: year of the movie was released, integer
 - director: director of the movie, text
- reviewer
 - rid: reviewer id, int
 - name: name of reviewer, text

Task 1¶

Write the SELECT query that finds all the movies that have a rating range (max stars-min stars) less than or equal to 2 stars but with at least 2 reviews. Your query should return the title of each movie and the rating range for the corresponding movie. Your SQL query should be written as a string variable called query in Python

In [8]:

```
import sqlite3
```

In [9]:

```
conn = sqlite3.connect('movie_ratings.db')
cursor = conn.cursor()
```

In [2]:

```
#### TEST FUNCTION: test_tight_range
# DO NOT REMOVE THE LINE ABOVE
```

```
query = ""
"""
```

Task 2¶

Write the SELECT query, as a string variable called query, that returns the summary of the movies, i.e. for each movie you should report, in the following order:

- the movie title
- the director of the movie. If a movie's director is NULL, assign 'Unknown Director' to those observations. Hint: COALESCE
- the year that movie was filmed
- the number of ratings that movie receives
- the average stars of that movie across all ratings

In [3]:

```
#### TEST FUNCTION: test_summary
# DO NOT REMOVE THE LINE ABOVE
```

```
query = ""
"""
```

In [14]:

Task 3 - creating variables¶

To answer the question whether movies before 1980 were better, please calculate the average rating for each movie, then calculate the average of these averages before vs after 1980. Your SELECT statement should return 2 averages, one for before and one for after 1980. Please assign the difference between the two averages to the variable `old_vs_new`.

In [4]:

```
### TEST FUNCTION: test_old_vs_new
# DO NOT REMOVE THE LINE ABOVE
```

```
query = ""
```

```
""
```

Task 4 - identifying bad data and multiple joins¶

Some reviews do not have a rating date (that is, the `ratingdate` is NULL). Return the title of the movie, the name of reviewer, and the stars the reviewer gave the movie for all ratings without a rating date. Make sure your SELECT statement is in a string variable called `query`.

In [5]:

```
### TEST FUNCTION: test_null
# DO NOT REMOVE THE LINE ABOVE
```

```
query = ""
```

```
""
```

In [13]:

<https://powcoder.com>

Add WeChat powcoder