

SOEN 363

## Project Phase 1

**Team Name:**

The Better Group

**Team Members:**

Nimit Jaggi - 40032159

Adrien Kamran - 40095393

Pierre-Olivier Trottier - 40059235

Tiffany Zeng - 40063115

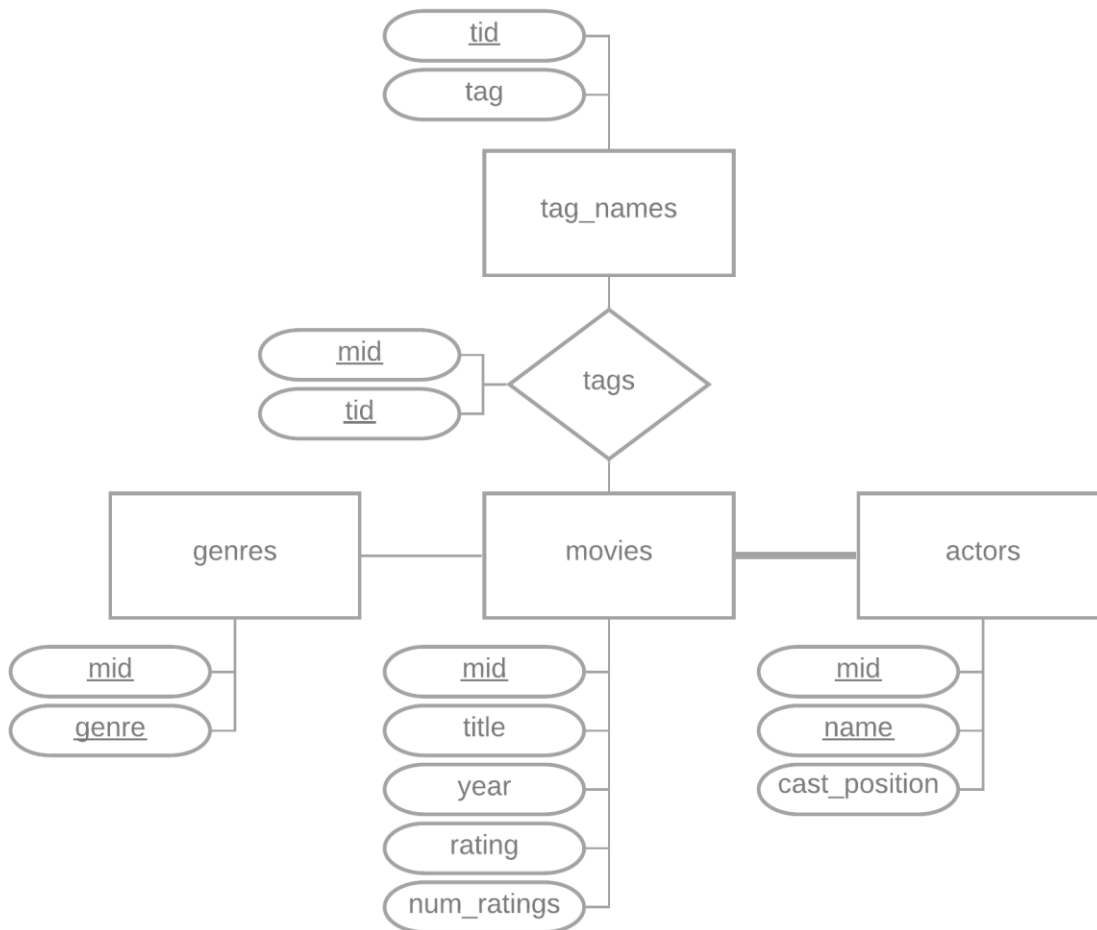
Report submitted in partial fulfillment of the requirements of SOEN 363

Concordia University

March 14<sup>th</sup>, 2021

### **Question 2A**

ER Diagram:



The queries required in this question can be found in the TXT file with the appropriate name.

### **Question 3A**

The queries required in this question can be found in the TXT file with the appropriate name.

### **Question 3B**

The queries required in this question can be found in the TXT file with the appropriate name.

### **Question 3C**

The queries required in this question can be found in the TXT file with the appropriate name.

### **Question 3D**

The queries required in this question can be found in the TXT file with the appropriate name.

### **Question 3E**

The queries required in this question can be found in the TXT file with the appropriate name.

### **Question 3F**

- The queries required in this question can be found in the TXT file with the appropriate name.
- The queries required in this question can be found in the TXT file with the appropriate name.
- From the data returned from the last two queries, we know that this isn't the case; the movie with the highest rating isn't among the movies with the highest number of ratings.
  - Highest number of ratings: mid 4201/53125, title Pirates of the Caribbean: At World's End
  - Highest rating: mid 4311, title 1732 Høtten
- The queries required in this question can be found in the TXT file with the appropriate name.
- From the data returned from the last query, we know that the lowest rating (for a movie which has been rated more than 0 times) is 1.5 This means that it is not the case for the movie with the highest number of ratings (currently 1'768'593 ratings at 3.8) to be the movie with the lowest rating (currently 3 ratings at 1.5).
- According to the results from the previous queries, it can be safely said that the hypothesis is false for this database.

### **Question 3G**

The queries required in this question can be found in the TXT file with the appropriate name.

### **Question 3H**

- The queries required in this question can be found in the TXT file with the appropriate name.
- The queries required in this question can be found in the TXT file with the appropriate name.
- The queries required in this question can be found in the TXT file with the appropriate name.

### **Question 3I**

The queries required in this question can be found in the TXT file with the appropriate name.

### **Question 3J**

- The queries required in this question can be found in the TXT file with the appropriate name.
- The queries required in this question can be found in the TXT file with the appropriate name.
- The queries required in this question can be found in the TXT file with the appropriate name.
- The queries required in this question can be found in the TXT file with the appropriate name.

### **Question 3K**

- The queries required in this question can be found in the TXT file with the appropriate name.
- The queries required in this question can be found in the TXT file with the appropriate name

### **Question 3L**

The queries required in this question can be found in the TXT file with the appropriate name.

### **Question 3M**

The queries required in this question can be found in the TXT file with the appropriate name.

### **Question 4A**

The queries required in this question can be found in the TXT file with the appropriate name.

Since *Postgres* automatically generates indexes for primary keys and foreign keys, the team simply had to focus on attributes that were often being queried and that were not keys. Such attributes could then be indexed to allow the queries that used them to be optimized.

#### **Question 4B**

##### *BEFORE*

- 3A: 0.021s
- 3B: 0.005s
- 3C: 0.014s
- 3D: 0.085s
- 3E: 0.014s
- 3F: 0.009s | 0.008s | 0.008s
- 3G: 0.009s
- 3H: 0.101s | 0.101s | 0.171s
- 3I: 0.127s
- 3J: 0.021s | 0.051s | 0.100s | 0.063s
- 3K: 0.016s | 12.39s
- 3L: 23.666s
- 3M: 0.019s

##### *AFTER*

- 3A: 0.006s
- 3B: 0.007s
- 3C: 0.014s
- 3D: 0.082s
- 3E: 0.015s
- 3F: 0.006s | 0.008s | 0.008s
- 3G: 0.009s
- 3H: 0.100s | 0.105s | 0.154s
- 3I: 0.124s
- 3J: 0.008s | 0.050s | 0.102s | 0.138s
- 3K: 0.014s | 11.169s
- 3L: 12.301s
- 3M: 0.024s

As can be seen in the results, most of the query performances were improved or unchanged, which is expected for queries that used previously non-index attributes or previously indexed keys. One result stands out by being worse after adding indexes though: *Question 3J part 4*. We reproduced these results on many occasions, and every time the indexed query ended up being twice as slow as the non-indexed version.

#### **Question 4C**

The queries required in this question can be found in the TXT file with the appropriate name.

#### **Question 4D**

The queries required in this question can be found in the TXT file with the appropriate name.