Movie Database with Discord Bot Controls

Christopher Youn, Marc Domingo, Connor Martindale, Gabriel Garcia, Michael Woodward

Presentation + Demo (60 points): The presentation (power point) will be between 10-15 minutes and occur during the last two weeks of class. Sign up information will be provided before the presentation.

The presentation will contain the following:
1. Problem/Issue you are trying to resolve
2. Your solution to the problem
3. Schema diagram
4. Demo

Presentation + Demo (60 points):

10-15 minutes per group

Make sure to introduce your project's concept and describe the application's features. Your project needs not be finished by the time you present but you must have something to demo.

The Problem

- Do you ever find yourself sitting around, bored, with no idea what to do?
- Do you waste countless minutes of your precious free time trying to make decisions about how to spend it?
- Once you make the inevitable decision to watch a movie, do you spend hours, days, even years trying to find one to watch?
- Do you find yourself 2 hours later, still scrolling through the endless halls of Disney+ having yet to make a decision?
- Do you wish **you had help** getting through that horrible decision making phase?

Making decisions is hard!

Discord Movie Bot™

- Stores and provides data on:
  - Movies
  - Actors
  - Composers
  - Studios
  - And More!!
- Simple, easy to use commands

- Capabilities:
  - Search for movies with a MASSIVE SELECTION of filters and options!
  - Add/Delete movies
  - Add user ratings to movies
  - View information on all of our 8 entities!
  - And most importantly...

Discord Movie Bot™

- Stores and provides data on:
  - Movies
  - Actors
  - Composers
  - Studios
  - And More!!
- Simple, easy to use commands

- Capabilities:
  - Search for movies with a MASSIVE SELECTION of filters and options!
  - Add/Delete movies
  - Add user ratings to movies
  - View information on all of our 8 entities!
  - And most importantly...

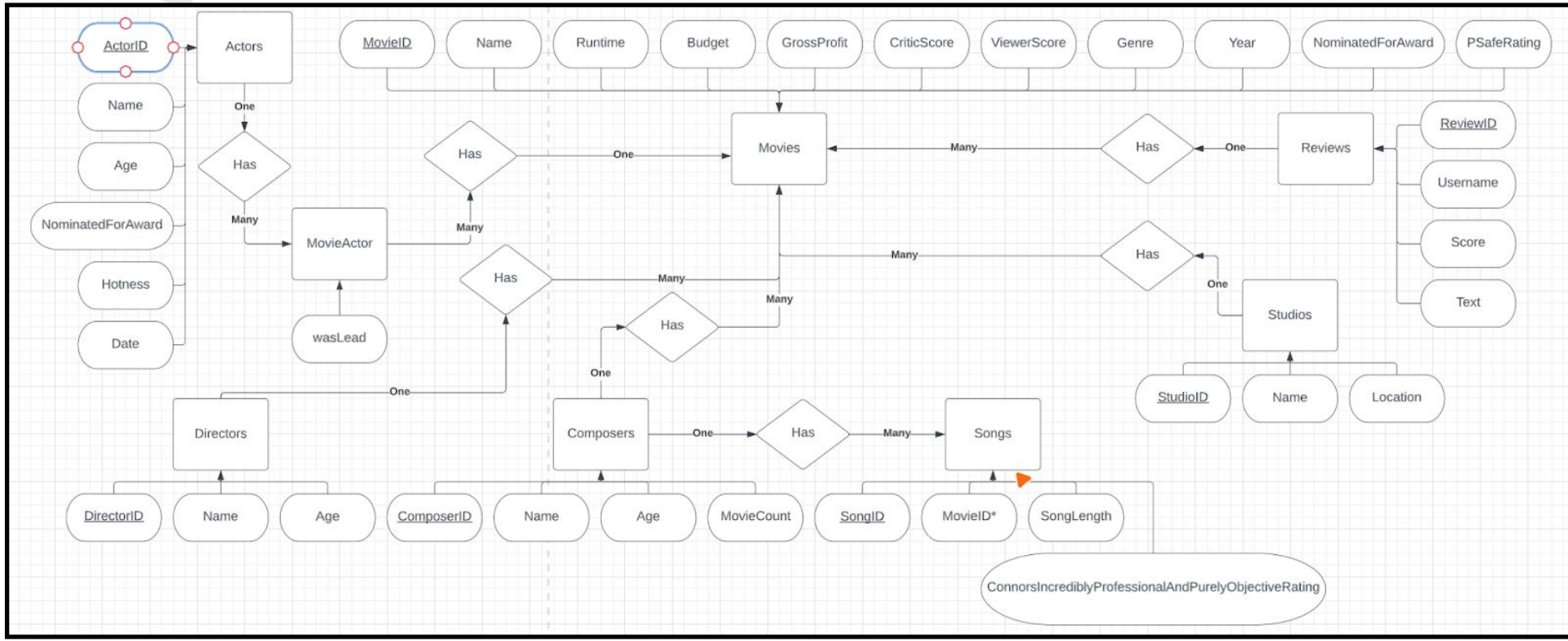
Chooses A Movie, So **You Don't Have To!**

How it Works

- User Input via Discord:
  - Submit commands via the bot to the Python script
    - Use:
    - /command parameters

- Back-End MySQL Database Connection
  - Discord Bot™ connects to a MySQL database
  - Uses separate Python file to handle queries
  - Users' commands are converted to the appropriate input for the Python script
  - Python script returns information to the bot, which outputs to the user

ER Diagram


ActorID
Actors
Name
Age
NominatedForAward
Hotness
Date
One
Has
Many
MovieActor
wasLead
MovieID
Name
Runtime
Budget
GrossProfit
CriticScore
ViewerScore
Genre
Year
NominatedForAward
PSafeRating
Has
One
Movies
Many
Has
One
Reviews
ReviewID
Username
Score
Text
Has
Many
Has
Many
Many
Has
One
Studios
StudioID
Name
Location
One
Directors
DirectorID
Name
Age
One
Composers
One
Has
Many
Songs
ComposerID
Name
Age
MovieCount
SongID
MovieID*
SongLength
ConnorsIncrediblyProfessionalAndPurelyObjectiveRating

Schema

Movies(MovieID, Name, DirectorID*, ComposerID*, StudioID*, Runtime, Budget, Gross Profit, CriticScore, ViewerScore, Genre, Year, NominatedForAward, PSafeRating)

Actors(ActorID, Name, Age, NominatedForAward, Hotness, Date)

MovieActor(ActorID*,MovieID*, wasLead).

Directors(DirectorID, Name, Age)

Composers(ComposerID, Name, Age, MovieCount)

Songs(SongID, ComposerID*, MovieID*, ConnorsIncrediblyProfessionalAndPurelyObjectiveRating, SongLength)

Studios(StudioID, Name, Location)

Reviews(ReviewID, Username, MovieID*, Score, Text)

The Plan

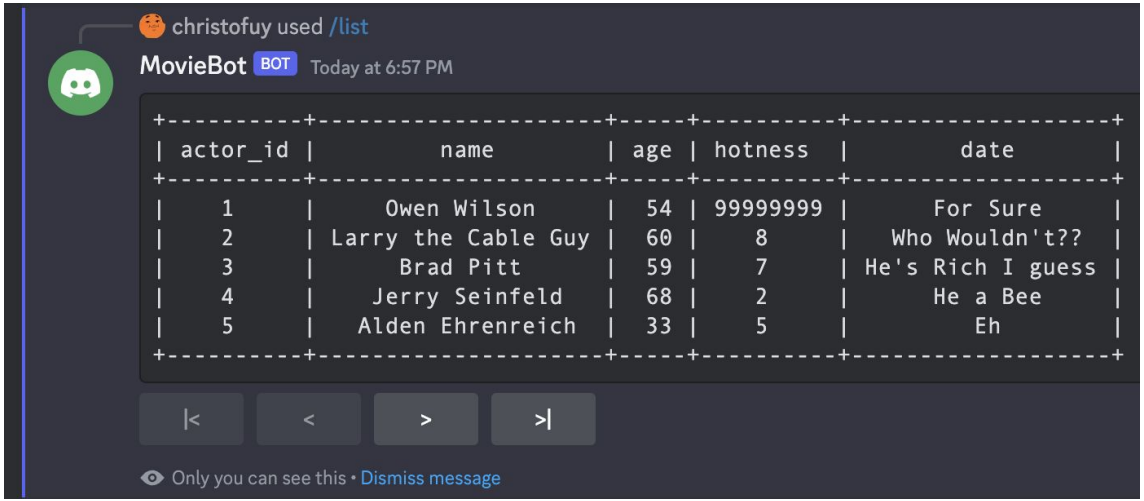
- Our goal is to create an army worthy of Mordor
- Our goal is to create a discord bot that interacts with a database
- The database has tables representing:
- The database is interacted with by text prompts to the discord bot

SQL Back End Progress

- Creates each of the empty entities
  - Applies constraints where needed
- Imports pre-existing data to fill the table
  - Takes in one .csv file per entity, cleans it, and adds the data
- Created sample data for each entity
- Basic Command Line Interface
  - For testing purposes only
- Allows users to view contents of each entity
  - Receive one or five random entries, or all entries
- Returns Filtered Results
  - Sort by genre, budget, average critic score, actors' age, and more!
- Uses Inner Joins to display detailed information about a movie when searching for user-made reviews
- Created a log table for each entity to track changes within the database
- Recommends a random movie
  - Provides title, runtime, and genre
- Allows user to add a new movie to the database
  - Also prompts users to add a Director, Studio, and Composer for the movie if matching names are not found

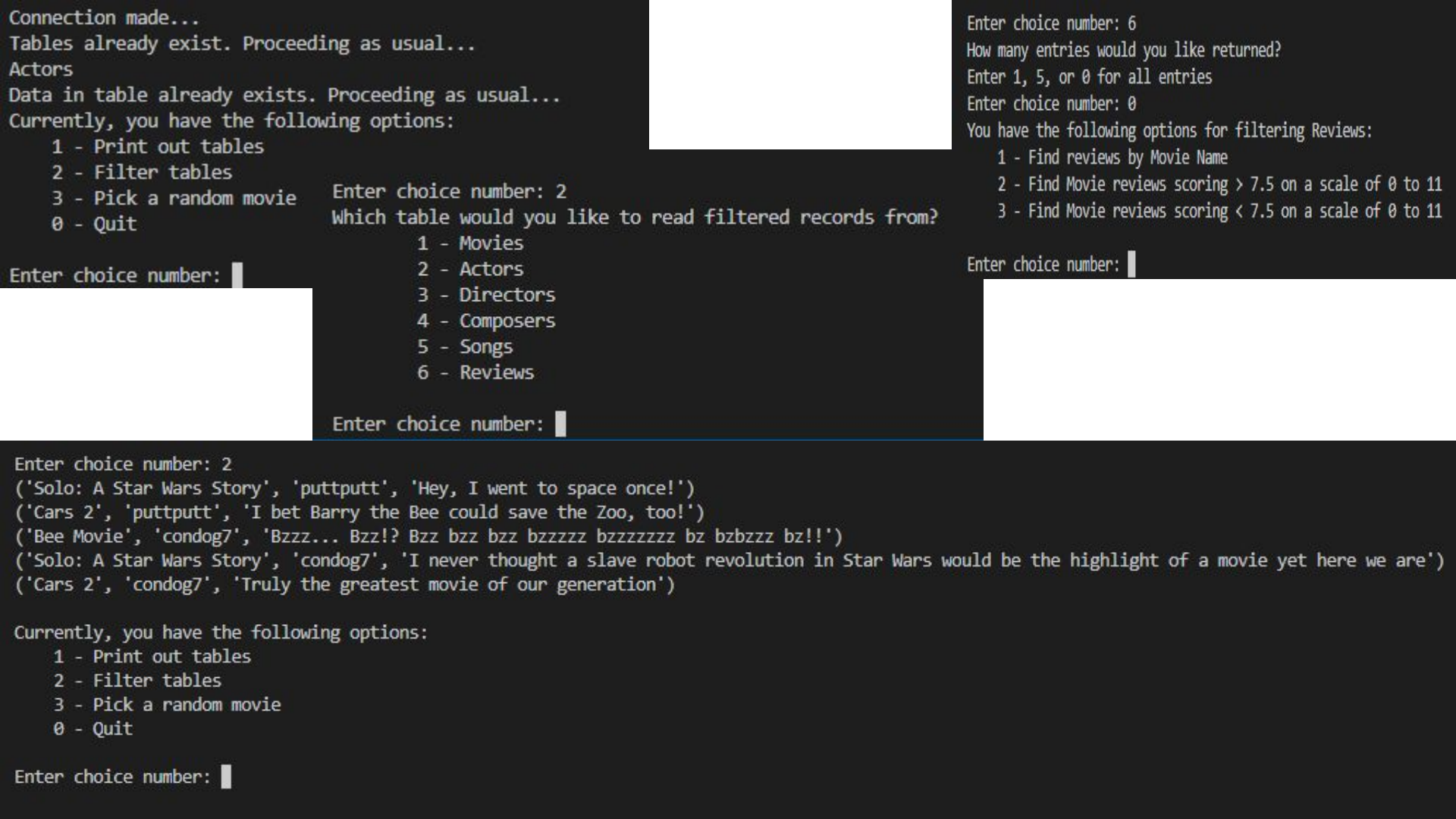
Demo Time!


christofuy used /get movie
MovieBot BOT Today at 6:56 PM
Inglourious Basterds
Director
Quentin Tarantino
Composer
Rupert Gregson-Williams
Studio
A Band Apart
Runtime
153 min
Budget
70000000
Gross Profit
321400000
Critic Score
89
Viewer Score
88
Genre
War/Comedy
Year
2009
Nominated for Award
Yes
P Safe Rating
NOTSAFE NOTSAFE NOTSAFE
Only you can see this • Dismiss message


christofuy used /list
MovieBot BOT Today at 6:57 PM
actor_id	name	age	hotness	date
1	Owen Wilson	54	99999999	For Sure
2	Larry the Cable Guy	60	8	Who Wouldn't??
3	Brad Pitt	59	7	He's Rich I guess
4	Jerry Seinfeld	68	2	He a Bee
5	Alden Ehrenreich	33	5	Eh
<
<
>
>|
Only you can see this • Dismiss message

Next Steps - Front End

- create command
- update command
- recommend command
- command for generating reports
- add more to get command
  - display statistics
- add more to list command
  - add filter options


Connection made...
Tables already exist. Proceeding as usual...
Actors
Data in table already exists. Proceeding as usual...
Currently, you have the following options:
1 - Print out tables
2 - Filter tables
3 - Pick a random movie
0 - Quit
Enter choice number:
Enter choice number: 2
Which table would you like to read filtered records from?
1 - Movies
2 - Actors
3 - Directors
4 - Composers
5 - Songs
6 - Reviews
Enter choice number:
Enter choice number: 6
How many entries would you like returned?
Enter 1, 5, or 0 for all entries
Enter choice number: 0
You have the following options for filtering Reviews:
1 - Find reviews by Movie Name
2 - Find Movie reviews scoring > 7.5 on a scale of 0 to 11
3 - Find Movie reviews scoring < 7.5 on a scale of 0 to 11
Enter choice number:
Enter choice number: 2
('Solo: A Star Wars Story', 'puttputt', 'Hey, I went to space once!')
('Cars 2', 'puttputt', 'I bet Barry the Bee could save the Zoo, too!')
('Bee Movie', 'condog7', 'Bzzz... Bzz!? Bzz bzz bzz bzzzzz bzzzzzzz bz bzbzzz bz!!')
('Solo: A Star Wars Story', 'condog7', 'I never thought a slave robot revolution in Star Wars would be the highlight of a movie yet here we are')
('Cars 2', 'condog7', 'Truly the greatest movie of our generation')
Currently, you have the following options:
1 - Print out tables
2 - Filter tables
3 - Pick a random movie
0 - Quit
Enter choice number:

Next Steps - Back End

- Expand range of user interaction to include:
  - Create new records
    - TODO: Actors, Songs, Reviews
  - Soft delete records
  - Update records
  - Generate reports
    - Uses groupby, subqueries, and 3+ table join
- Implement transactions to maintain referential integrity
- Incorporate logs with CRUD actions

Thanks for Listening


THAT WAS NOT VERY
HEALTHY PANTHER
OF YOU