Q1 The "Googling" Part (15 points)

Give a short definition of each of the following terms:

1. Conceptual Database Design

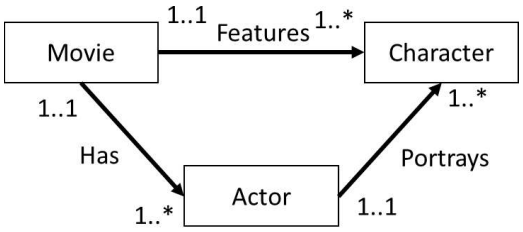
2. Logical Database Design

3. Physical Database Design

Q2 It's a Trap! (or maybe not) (40 points)

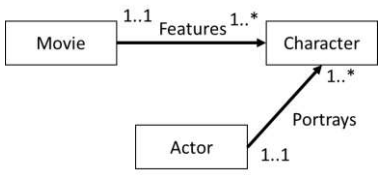
The following E-R diagram represents the initial attempt at defining the relationships between entities for a movie database (like IMDB). Our team has determined that there is a design problem (a trap, redundant relationship, etc.). Each member of our team suggests removing a relationship to correct the problem. Steve says we should remove the Has relationship; Tom says we should remove the Features relationship; Rita suggests the Portrays relationship should be removed (diagrams shown below). Note: Assume for this problem that all the relationship types (1:1, 1:*, etc.) are correct.

1. Identify the type of design problem with the original diagram.

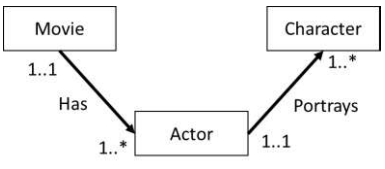


2. Identify which of the solution(s) could correctly solve the problem.
3. For the solution(s) that do not correctly address the problem, identify the reason why (i.e. what new problem would result from the incorrect solution)

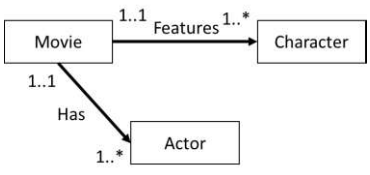
Steve



Tom



Rita



Q3 Six Degrees of Entity Relation (20 points)

We discussed with the customer for our movie database and it seems that we had an even bigger design problem. Among other things, our design does not account for the same character being in multiple movies (such as sequels). Redraw the E-R diagram to reflect the following requirements:

1. The same character can be in more than one movie (such as sequels).
2. The character does not have to be played by the same actor in all movies.
3. An actor can be in multiple movies (like Kevin Bacon).
4. The same actor can play more than one character in the same movie (like Eddie Murphy in Coming to America).

Q4 Abnormal Normalization (40 points)

Given the following data from our customer (on the next page), show a normalized data set that matches your above E-R diagram. This data set should be in 3NF. For this problem show the following:

1. Relation definitions including relation names, attribute names, and attribute types.
2. Primary Key and Foreign Key definitions.
3. Any constraints that might exist.
4. All tables in your database with data entered to match the following example data.

- NOTE: Not all columns below must appear in the normalized tables. You will answer why this is the case in the next question.
- You may abbreviate data within reason (i.e. initials for actor/character/movie names)

MovieCharacters

Title	Subtitle	Part	Release Date	Minutes	Character Name	Lead	Actor Name	Birthdate	FilmCnt
Batman Begins		1	15.06.2005	140	Batman/Bruce Wayne	TRUE	Christian Bale	30.01.1974	2
Batman Begins		1	15.06.2005	140	Alfred	FALSE	Michael Caine	14.03.1933	2
Batman v Superman	Dawn of Justice	1	25.03.2016	151	Batman/Bruce Wayne	TRUE	Ben Affleck	15.08.1972	1
Batman v Superman	Dawn of Justice	1	25.03.2016	151	Lois Lane	TRUE	Amy Adams	20.08.1974	1
Batman v Superman	Dawn of Justice	1	25.03.2016	151	Superman/Clark Kent	FALSE	Henry Cavill	05.05.1983	1
Coming to America		1	29.06.1988	116	Prince Akeem	TRUE	Eddie Murphy	03.04.1961	5
Coming to America		1	29.06.1988	116	Clarence	FALSE	Eddie Murphy	03.04.1961	5
Coming to America		1	29.06.1988	116	Randy Watson	FALSE	Eddie Murphy	03.04.1961	5
Shrek		1	18.05.2001	90	Shrek	TRUE	Mike Myers	25.05.1963	2
Shrek		1	18.05.2001	90	Princess Fiona	TRUE	Cameron Diaz	30.08.1972	2
Shrek		1	18.05.2001	90	Donkey	FALSE	Eddie Murphy	03.04.1961	5
Shrek		2	19.05.2004	93	Shrek	TRUE	Mike Myers	25.05.1963	2
Shrek		2	19.05.2004	93	Princess Fiona	TRUE	Cameron Diaz	30.08.1972	2
Shrek		2	19.05.2004	93	Donkey	FALSE	Eddie Murphy	03.04.1961	5
Star Wars	A New Hope	4	25.05.1977	121	Luke Skywalker	TRUE	Mark Hamill	25.09.1951	2
Star Wars	A New Hope	4	25.05.1977	121	Han Solo	FALSE	Harrison Ford	13.07.1942	2
Star Wars	The Force Awakens	7	18.12.2015	136	Han Solo	TRUE	Harrison Ford	13.07.1942	2
Star Wars	The Force Awakens	7	18.12.2015	136	Luke Skywalker	FALSE	Mark Hamill	25.09.1951	2
Star Wars	The Force Awakens	7	18.12.2015	136	Rey	FALSE	Daisy Ridley	10.04.1992	1
The Dark Knight		2	18.07.2008	153	Batman/Bruce Wayne	TRUE	Christian Bale	30.01.1974	2
The Dark Knight		2	18.07.2008	153	Alfred	FALSE	Michael Caine	14.03.1933	2
The Dark Knight		2	18.07.2008	153	Joker	FALSE	Heath Ledger	04.04.1979	1

Q5 Road Less Traveled By (30 points)

In the dataset given, what is the term for the type of data that FilmCnt is?

In this course, we discussed two options for how to represent this type of data. Show how both methods would be applied to your database for this dataset, including any queries/views that would be necessary. Which of these two methods is more reasonable for your database (i.e. Which one would you choose)? Explain your reasons why you would choose this option.

Q6 Part 2 of a Film (25 points)

Using the 3NF database **you produced** in the previous questions, produce SQL queries to do the following:

1. Identify all lead actors and actresses in films released in 2015.
2. Identify the oldest person acting in each movie by film.
3. Calculate the total running time for all movies with the same title (if we want to have a marathon).
4. Add your favorite film to the database along with one actor/actress and the role they played.
5. Remove a film you don't like from the database.

Q7 To Merge or Not to Merge (30 points)

Given below is the database definition for a series of tables created for a similar database for plays. We need to merge these two databases into a global database that will track both movies and plays. Show how you would merge these two databases. You may use any means we've discussed in class (E-R diagram, data definition, SQL queries, etc.).

During which phase (Conceptual, Logical, Physical) is this step performed?

NOTE: The other "team" may or may not have done a good job in designing their database. You are free to change any design flaws they may have for your merged model.

```
Play (Title, Author, YearWritten)
Primary Key Title, Author

Performer (PerformerNo, Name, BirthDate, UnderStudyTo)
Primary Key PerformerNo
Foreign Key UnderStudyTo references Performer(PerformerNo) ON UPDATE CASCADE ON DELETE SET
NULL

Role (Name, Gender, Ethnicity, PlayTitle, PlayAuthor)
Primary Key Name, PlayTitle, PlayAuthor
Foreign Key PlayTitle, PlayAuthor references Play(Title, Author) ON UPDATE CASCADE ON DELETE
NO ACTION

PerformerRole (PerformerNo, RoleName, PlayTitle, PlayAuthor, DateTimePerfomed)
Primary Key PerformerNo, RoleName, PlayTitle, PlayAuthor, DateTimePerfomed
Foreign Key PerformerNo references Performer(PerformerNo) ON UPDATE CASCADE ON DELETE NO
ACTION
Foreign Key RoleName, PlayTitle, PlayAuthor references Play(Name, PlayTitle, PlayAuthor) ON
UPDATE CASCADE ON DELETE NO ACTION
```