

In-class Activity #4: Working with SQL, Part 2

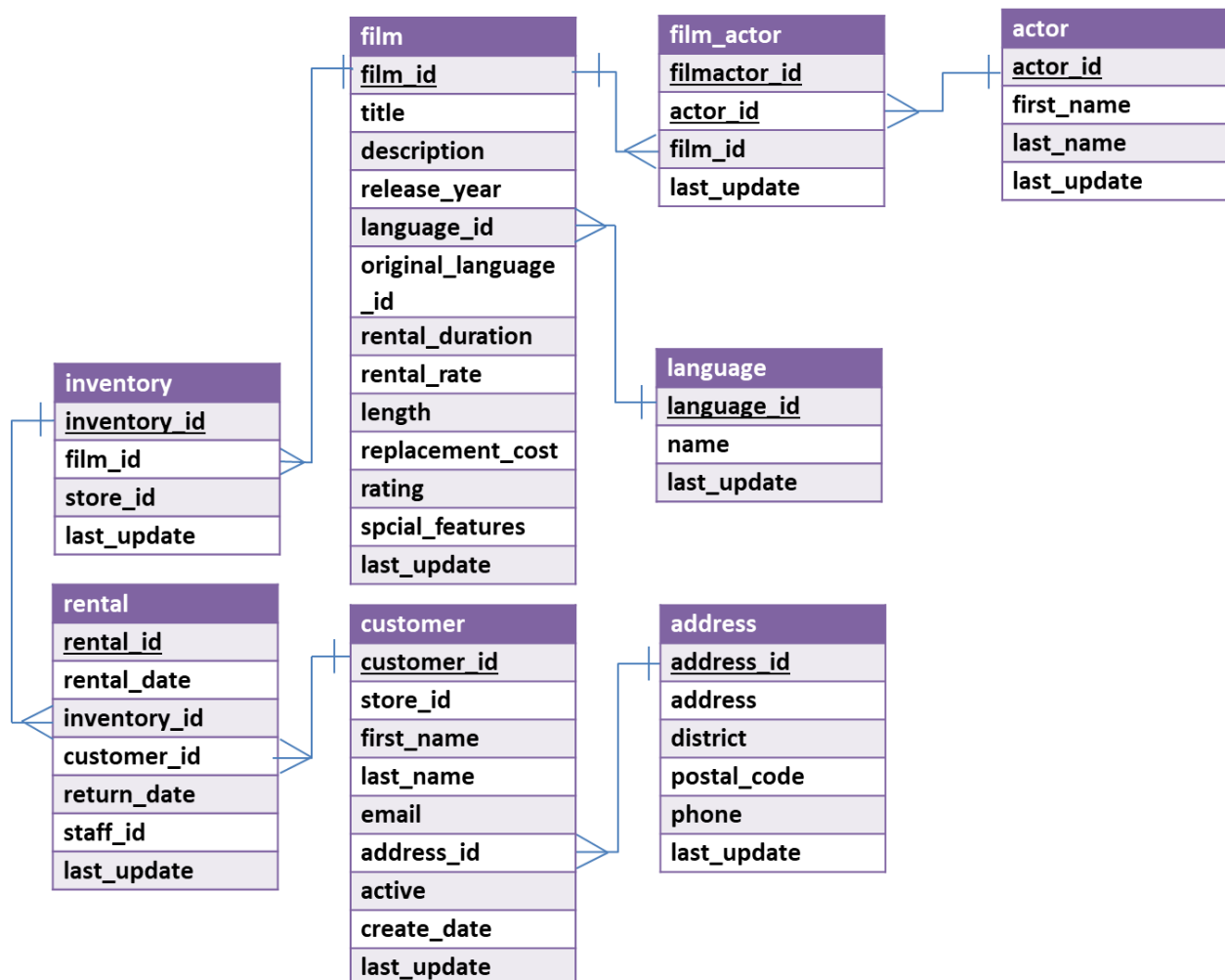
Getting Information out of a Database

Submit your solutions for Part 2 of this exercise using: <https://forms.gle/H89YMuPBok3oHuD9A>

- You must sign in using your @temple.edu email address.
- Other sign-ins will not be accepted.

In this exercise, you will be working with a movie rental database. The schema for this database is provided below. All of the tables are in a schema called simply “**moviedb**” (not m0moviedb, m1moviedb, etc.). You can’t write to any of the tables – you can only use SELECT statements to read from them (so don’t worry about causing any damage).

Spend some time looking at the schema carefully. The field names are self-explanatory.



You can see that a film has a title, description, rating, and length (among other things). You can also see that an actor has a first name and a last name. The film_actor table implements the many-to-many relationship between actor and film (i.e., a film can have more than one actor, and an actor can be in

more than one film). You'll also notice that data types are listed for each field, but they should be pretty obvious – for example, first_name is a VARCHAR because it is a string value.

Write queries on your own

Create the SQL SELECT query to answer each of the questions below. Some of the questions can be answered by querying one table; others will require joining multiple tables to get the answer.

For each question, you'll need to write down **(1) the SQL query** (which you can copy and paste from SQL Workbench) and **(2) the answer you get as a result of the query** (which you can copy and paste from the results).

Part 2: Write queries on your own...

Find the part 2 questions here. <https://forms.gle/H89YMuPBok3oHuD9A>

TIPS AND TRICKS: How to answer SQL questions...

1. First, ask yourself – what tables do I need to solve this problem?
2. If you only need one table... great! Start by writing the simplest possible statement:
select statement: select * from tablename
3. If you need multiple tables, then you need to think a little harder.
 - a. Identify the keys
 - b. Write the simplest select statement that you can **with those keys specified**. (This is **important** – if you do not specify the keys correctly you will end up with an inefficient, incorrect set of results.)
4. Now with your select statement ready, you can refine your results.
 - a. Replace the * with the specific column names you need in your result.
 - b. Add whatever extra criteria to the where clause necessary to answer the question.
 - c. Does the answer require an aggregate function (SUM, COUNT, MAX, MIN, AVG)? If so then you will *probably* need a GROUP BY statement as well.
 - d. Is it a question that can be answered using limit? Add and ORDER BY and LIMIT clause.
5. Remember, GROUP BY and ORDER BY are different things.
6. Double check your answer against the question. Make sure you are returning the correct columns, and that the records are in the correct order.
7. If your query is returning hundreds of records, you probably have a mistake.