Lab 9, Formulating Queries

CS 0131, Software for Personal Computing
Timothy J Parenti

10 April 2013

Last time, we designed database tables to hold information about movies, the actors in those movies, their directors, and reviews written about them. Today, we'll continue from where we left off to link up our relational data and ask some meaningful queries about it.

1 Setting Up Relationships

1. Log into CourseWeb and go to "My Grades". Click the green exclamation point in the **Lab 8** record to view your submission from last week, and download the `movies.accdb` file you submitted to your Desktop. Keep the original file handy, and create a copy of it called `movies2.accdb`.

   Note: If you missed last week, you will need to either create the database designed in **Lab 8** yourself or get a copy of a classmate's starting file. If you choose the latter, please make note of who you got the file from in the comments field of your submission.

2. Recall that we have a *Movies* table which holds release date, movie title, and an ID representing the person who directed the movie. Then, we have a *People* table which holds information about each person (their ID, which is generated automatically by Access, and their actual name). The *ActorInfo* table then keeps a big list of which people were actors in which movies. A person can appear in many different films, and can also be the director of a movie that they're an actor in.

   Before we continue, we need to establish the relationships between our tables. As people familiar with this database, we know the relationships between various tables (for example, the *ActorInfo* table maps people to the movies they were in). but Access needs to be explicitly told these relationships.

   Go to Database Tools » Relationships. Cause all four tables to appear. Go ahead and link all the fields that should be linked. You should have four links in total. If you want to double-check your work, I've listed the links you should have at the end of this lab.

Now that Access knows how all the information is related, we can create some queries to help view the information.

2 Formulating Queries

When making these queries, you will probably find it useful to create a few more records in our database to be sure what it's doing. You can go ahead and make up data for this purpose. The keyboard shortcut F5 re-runs a query on the new data.

1. Let's start off by creating a simple query. We would like to simply see that a movie has a certain director, instead of seeing that director's ID. Using the simple query wizard, let's do just that. Specifically, we would like to create a query which shows the title, release date, and revenue from the *Movies* table.

From the *People* table, we would like to just show the *PersonName*. Name your query *MovieInfoWithDirectorName*. Finish the wizard and view the query. Does it show what you want it to show?

2. We've only seen queries that list data or filter data. Queries can also compute values. Let's see how much each director made on average across all of the movies he/she has directed. Using the simple query wizard, we would like to show from the *MovieInfoWithDirectorName* query the following fields: revenue, *DirectorName*. However, before finishing the wizard, when you get to the step that asks whether you'd like the query to be a detail query or a summary query, click the summary option. Then, click the summary options button. Access is telling us what we can summarize about our fields. What can we calculate about the records we have? Which fields can we look at?

   For now, let's click the check boxes to calculate the average revenue, total revenue (sum), as well as the minimum and maximum revenue. We will ignore the box saying "Count records in movies" as we do not care how many records we have listed in movies (i.e., how many movies we have listed). Then, let's click OK. Click next and name your query, "money". Make sure the "Open query to view information" option is clicked. Click Finish, then adjust the width's of the fields so we can see all of the data.

   Our query is pretty useful, but it might be improved to help someone new use it. For example, the query says *PersonName* but really, it might be better to say, *Director*. Also, instead of *Avg of Revenue* we might prefer to call it *Average Revenue*.

   Open that query in design view and take a look at how the query is designed. For example, we see that the *PersonName* has "group by" (meaning that we're grouping our data based on director's names) and that the sum, minimum, maximum, and average fields are being computed from the revenue field. If we look in the "field" section we see a pattern. The fields that were computed say something like "Min of revenue: revenue" but the *PersonName* field, which wasn't computed by the query, just says *PersonName*.

   If we want to rename how a field is displayed in a query, we add or modify the part before the colon. So, where it says *PersonName*, let's change that to be "Director: PersonName". Access will still pull the information from the *PersonName* field, but it will pretend that the field is called "Director." Let's also rename the calculated fields to instead display "Total Revenue", "Average Revenue", "Minimum Revenue" and "Maximum Revenue." To do this, let's for example change it from "Min of revenue: revenue" to instead say, "Minimum Revenue: revenue". Re-run or re-open the query and make sure that things have changed. Looking at the query, it's now easier to see exactly what we're looking at.

3. We might also want a big listing of which actors were in which movies. Using the simple query wizard, let's make a new query. We would like to show the movie title and the actors in those movies. For starters, let's first make the title field from the movies table one of our selected fields.

   Now, we have to think about how we want to tell Access that it should show all the actors. If we simply added the *PersonName* field from the people table, Access might get confused and think that we meant a director. This is to be expected, because there is a direct relationship link (representing the director of each movie) between the people table and the movie table. We have to coax Access to instead use the actors table, even though we don't want to show any information from the Access table.

   Since we already have movies.title added as a selected field, let's also add both fields from the actors tables. By doing this, we're helping Access to know that we want a listing of actors. Then, let's add the *PersonName* field from the people table. By adding all four of these fields, instead of just two, we're telling Access how it should decide which people names to display. In our first query, we wanted just those people were directors. Now, we just want those who were directors.

   We just want a big listing of actors, so this is a detail query. Call the query *ActorsListing* and then look at the results of running the query.

   The query is working like we want it to. It shows the name of each movie and also, the actors who were in that movie. More importantly, it is **NOT** showing the directors of the movies. We can still improve this query though! Let's get rid of those useless fields that we don't care about (e.g., the *ActorsTitle* field, and the *ActorID* field).

Go to the design view of the query. Notice the checkboxes in the row labelled "Show:". We need those fields to be in our query, so that Access can make the link between the actor names and their movies, but we do not want them to appear. So, uncheck the boxes for the 2 fields we mentioned we don't want to see. Look at your query in design view again, to see the changes. Does it look better? Are the results still correct?

Let's rework the query a little more to make it easier to understand. Just like we did in our second query, make it so that instead of saying *MoviesTitle* and *PersonName* the query shows "Film Title" and "Actor/Actress". View the query to make sure everything is still correct.

4. When you're done with all three queries, call over the instructor to review them! While you're waiting, think about how we might make a query to show the movies that a specific actor appeared in.

   Once your database has been checked by the instructor, close it to ensure that all pending changes are saved before submission.

Submission

To receive credit for these exercises, call over the instructor, who will check that you have completed the assignment. Then, log into CourseWeb and use the "Assignment Submission" section to submit your "`movies2.accdb`" file for **Lab 9**.

You should be able to complete this lab in the allotted class time; however, if you are running low on time, the instructor will give you further instructions for completing the rest of this lab at home.

Your lab must be checked by the instructor BEFORE you leave the room AND you must submit your files to CourseWeb in order to receive credit for the lab! Don't forget!

Relationships

- Link the *Reviews* table and the *Movies* table by their shared title field.
- Link the *Movies* table with the *People* table by linking the movie director with the ID of the person.
- Link the *Movies* table with the *ActorInfo* table by linking the title of the movie with the "title" field of the *ActorInfo* table.
- Link the *ActorInfo* table with the *People* table by linking the Actor ID with the Person ID.