**Unit 5 Automated Inputs**

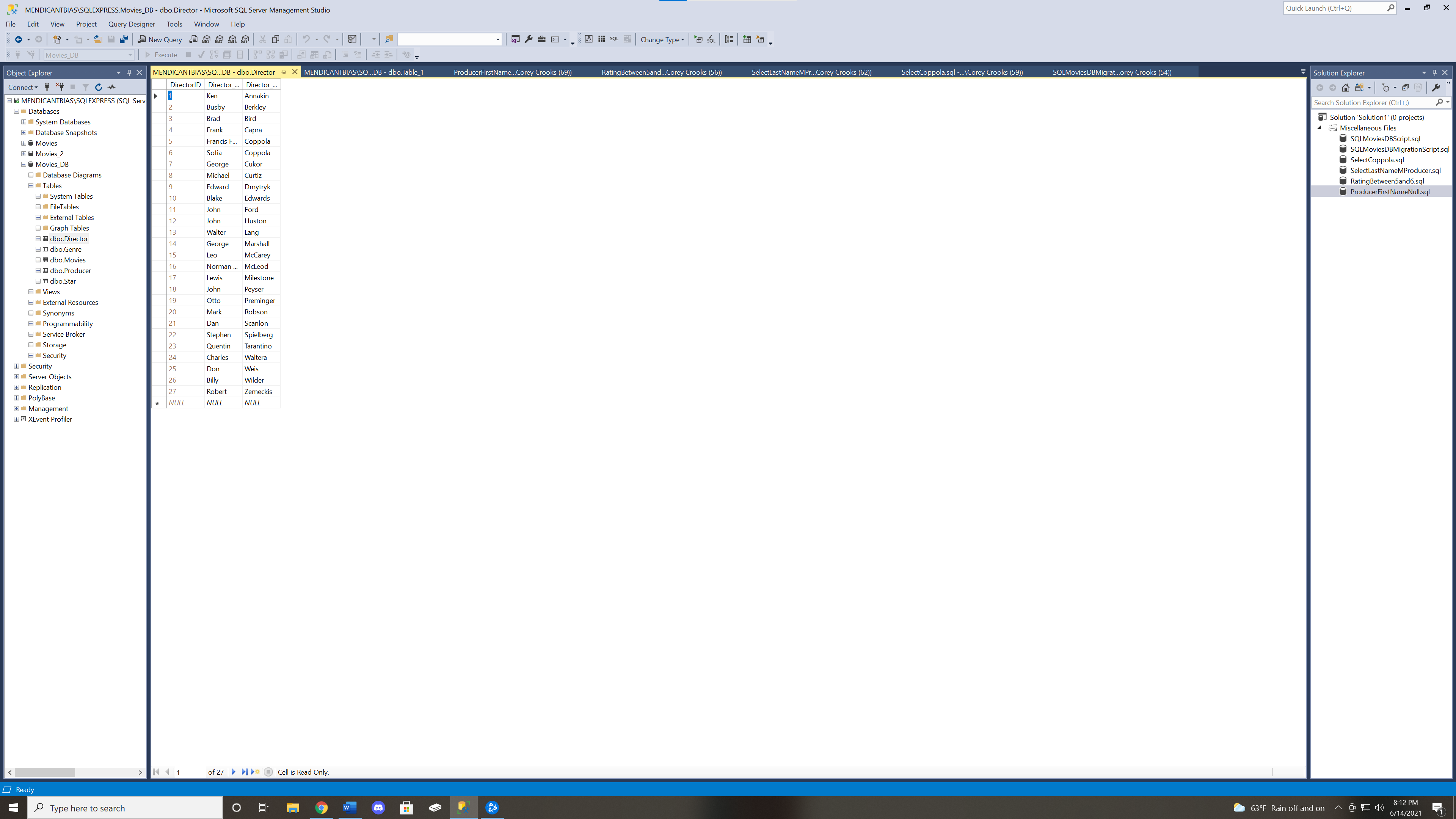
Corey Crooks

Purdue University Global

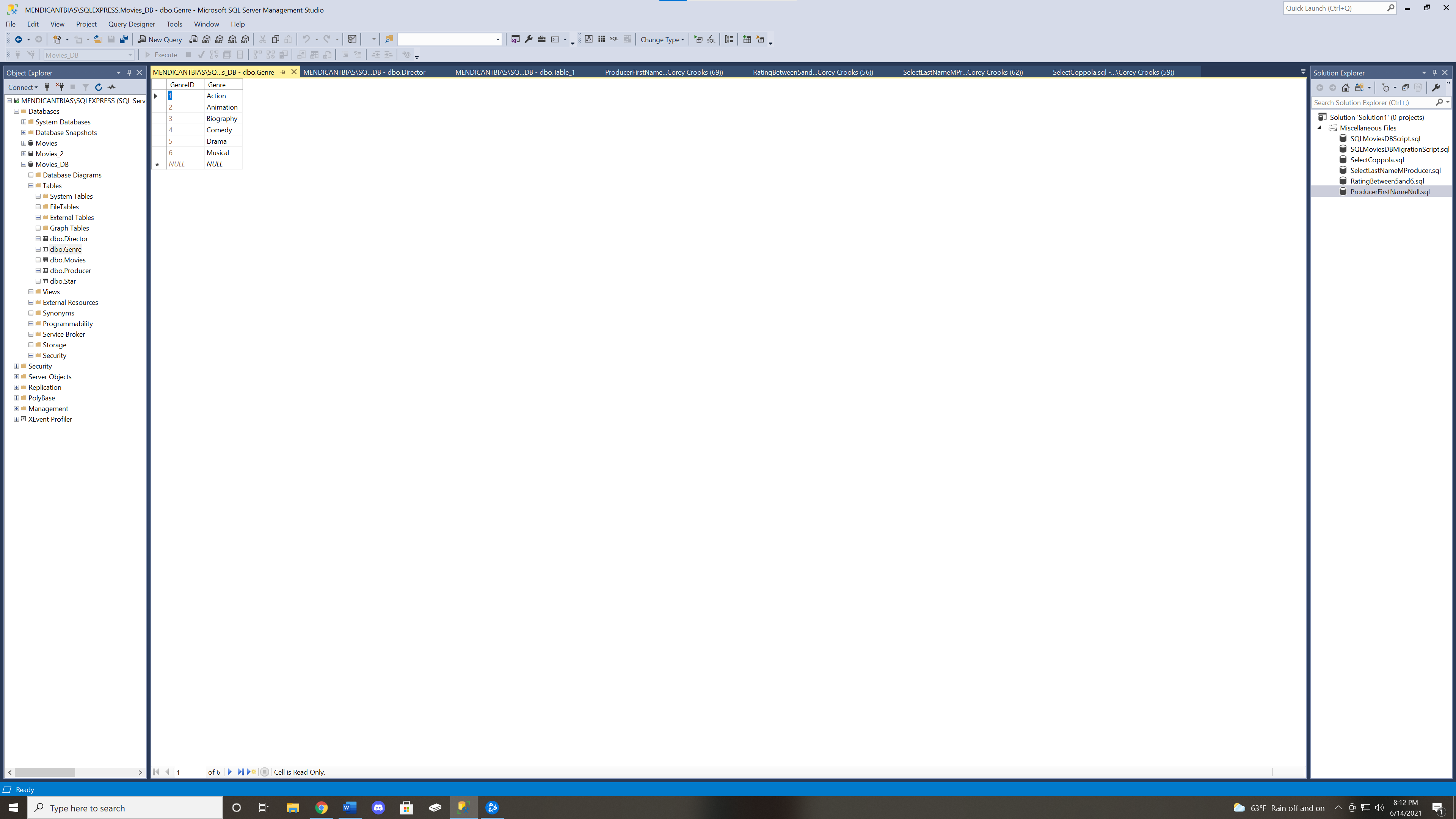
IT234 – Leon King

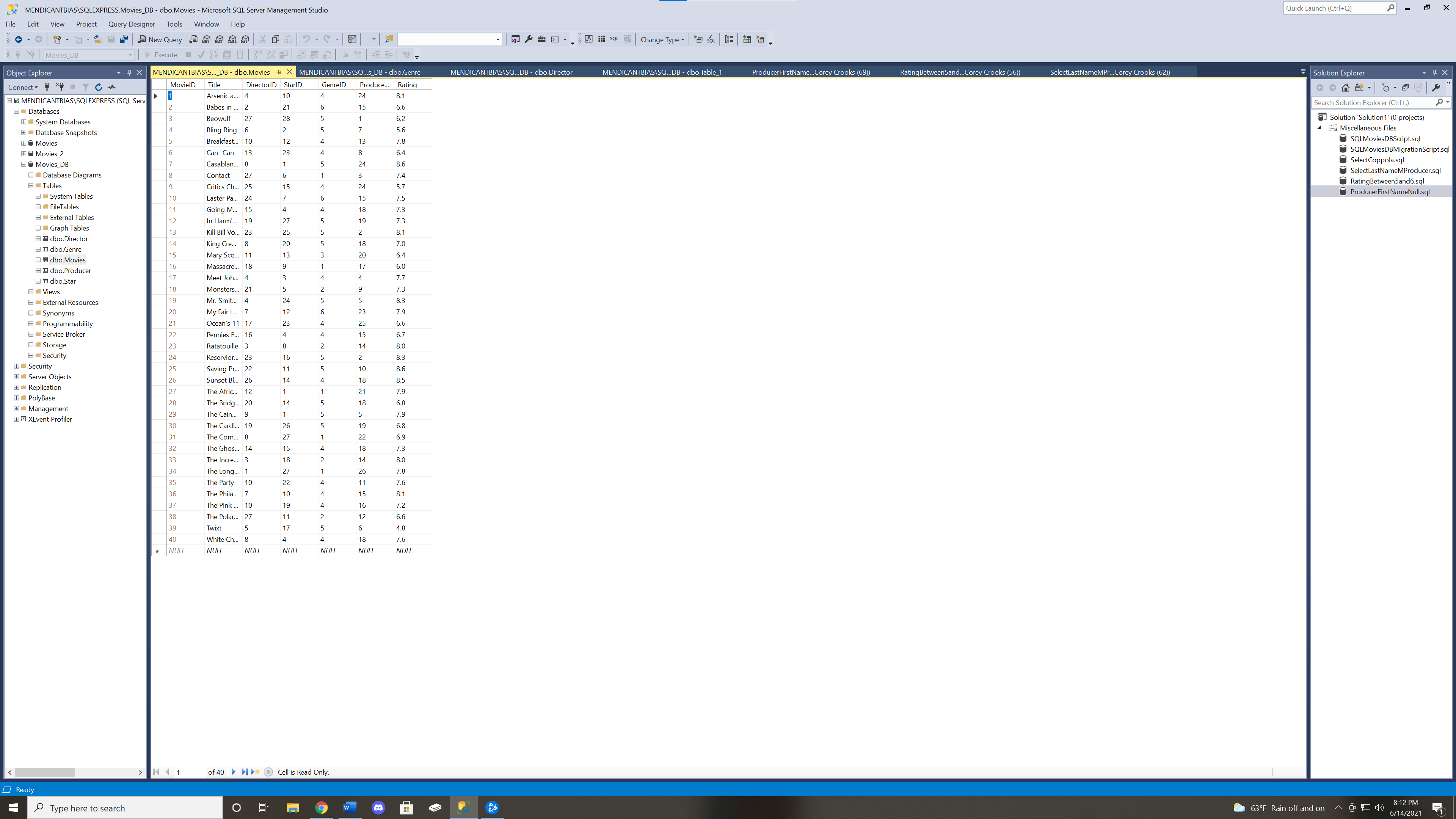
May 16, 2021

**Begin Table Proofs**

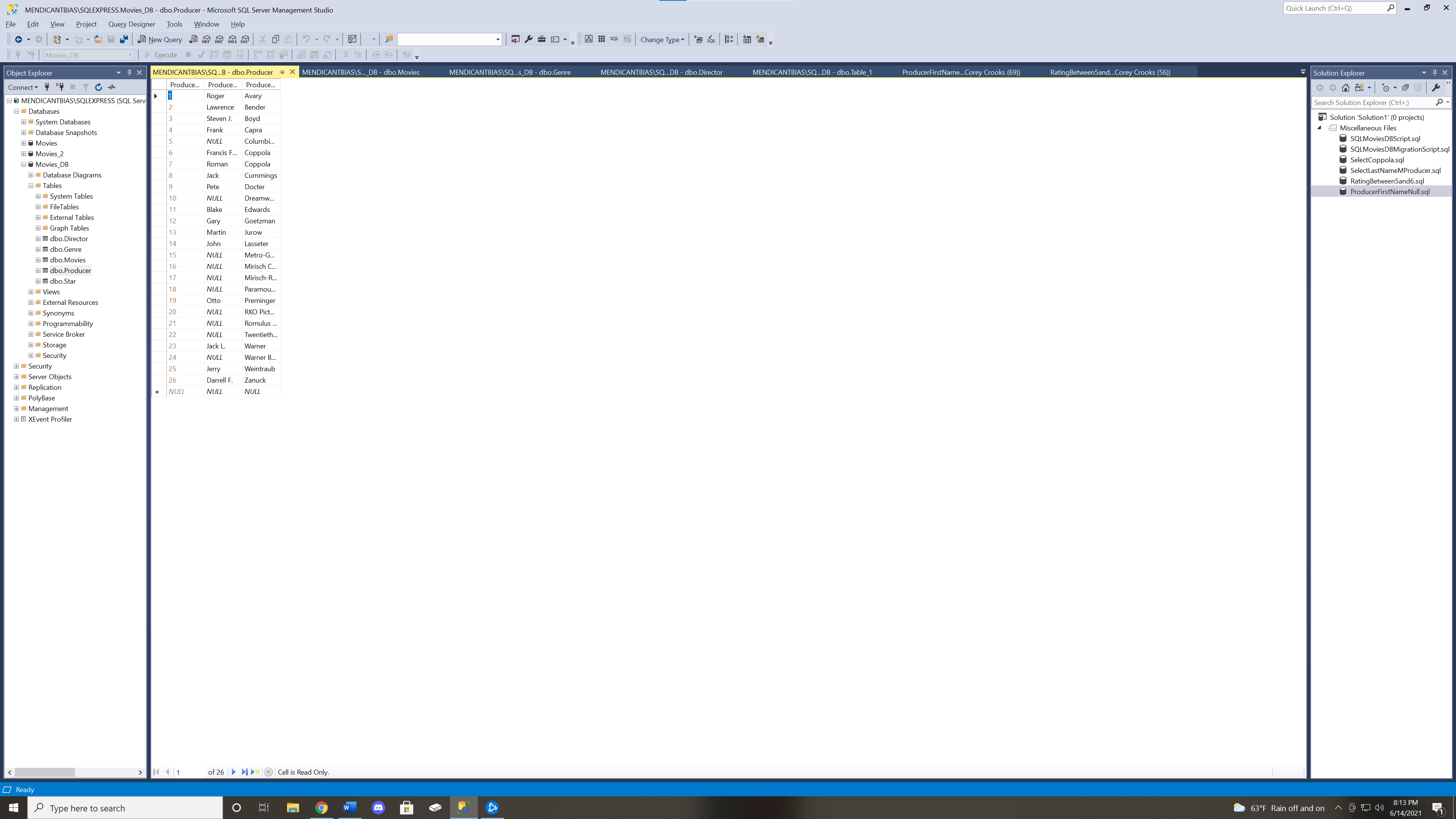


Director

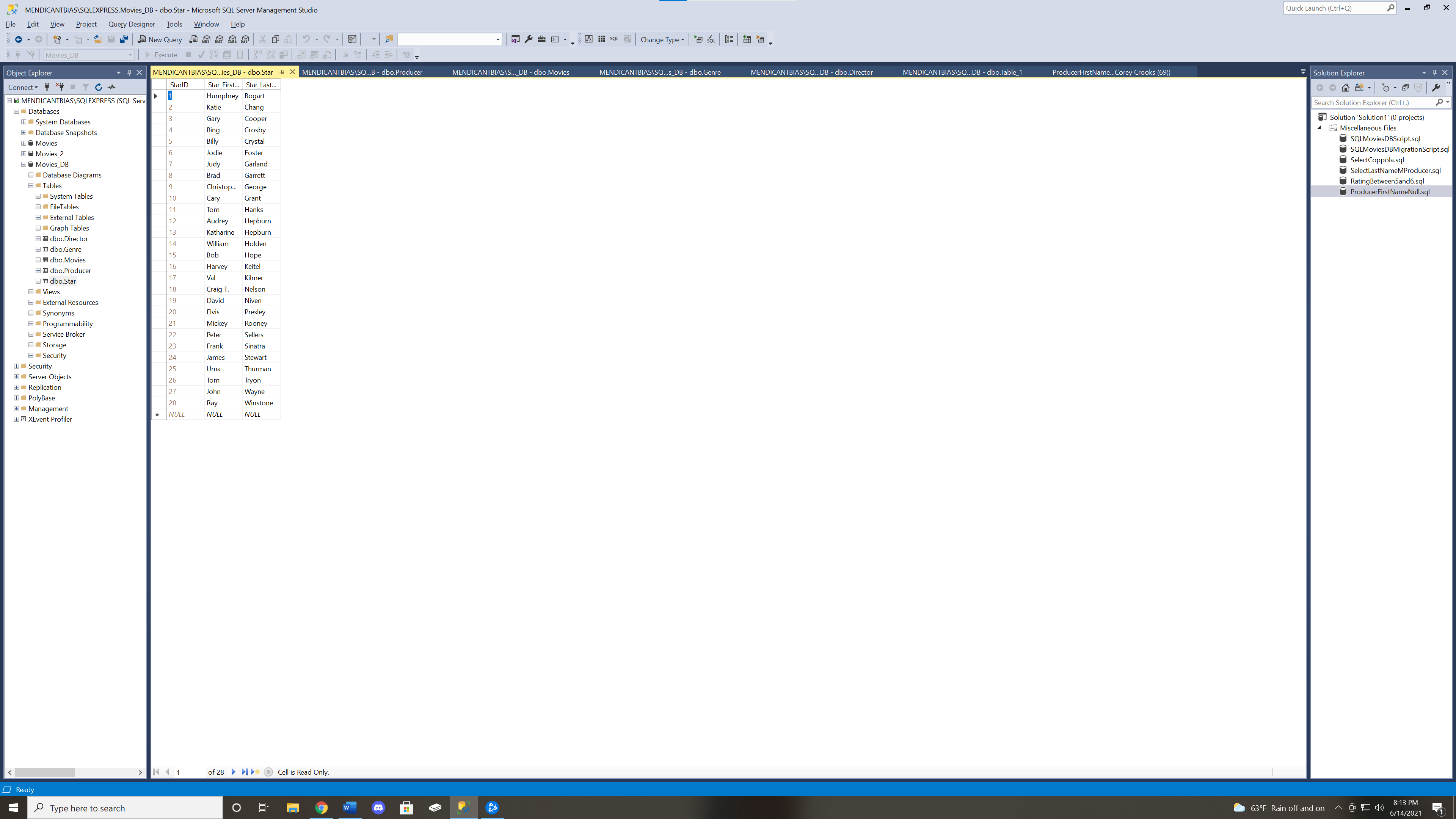
Genre



Movies

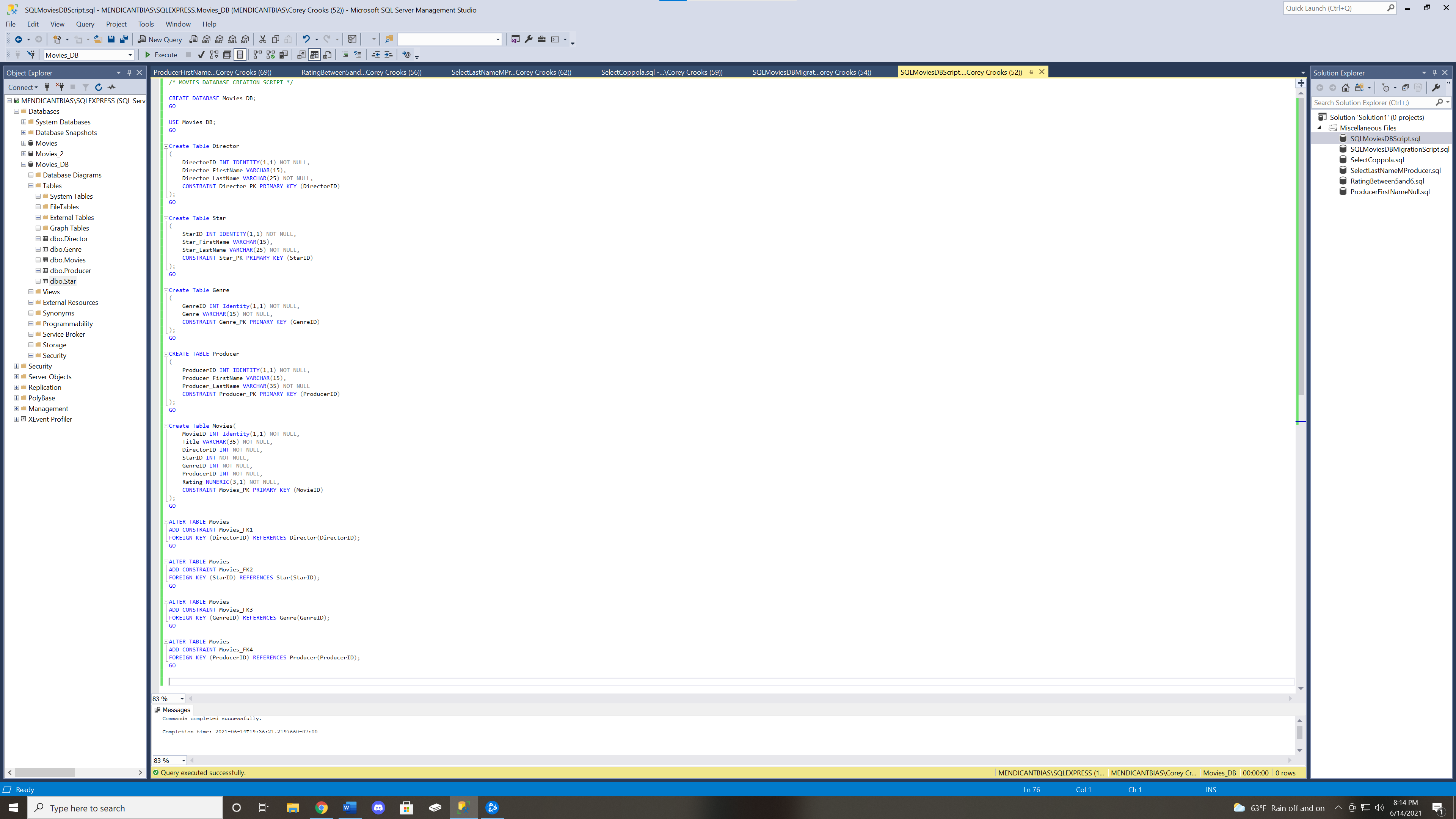


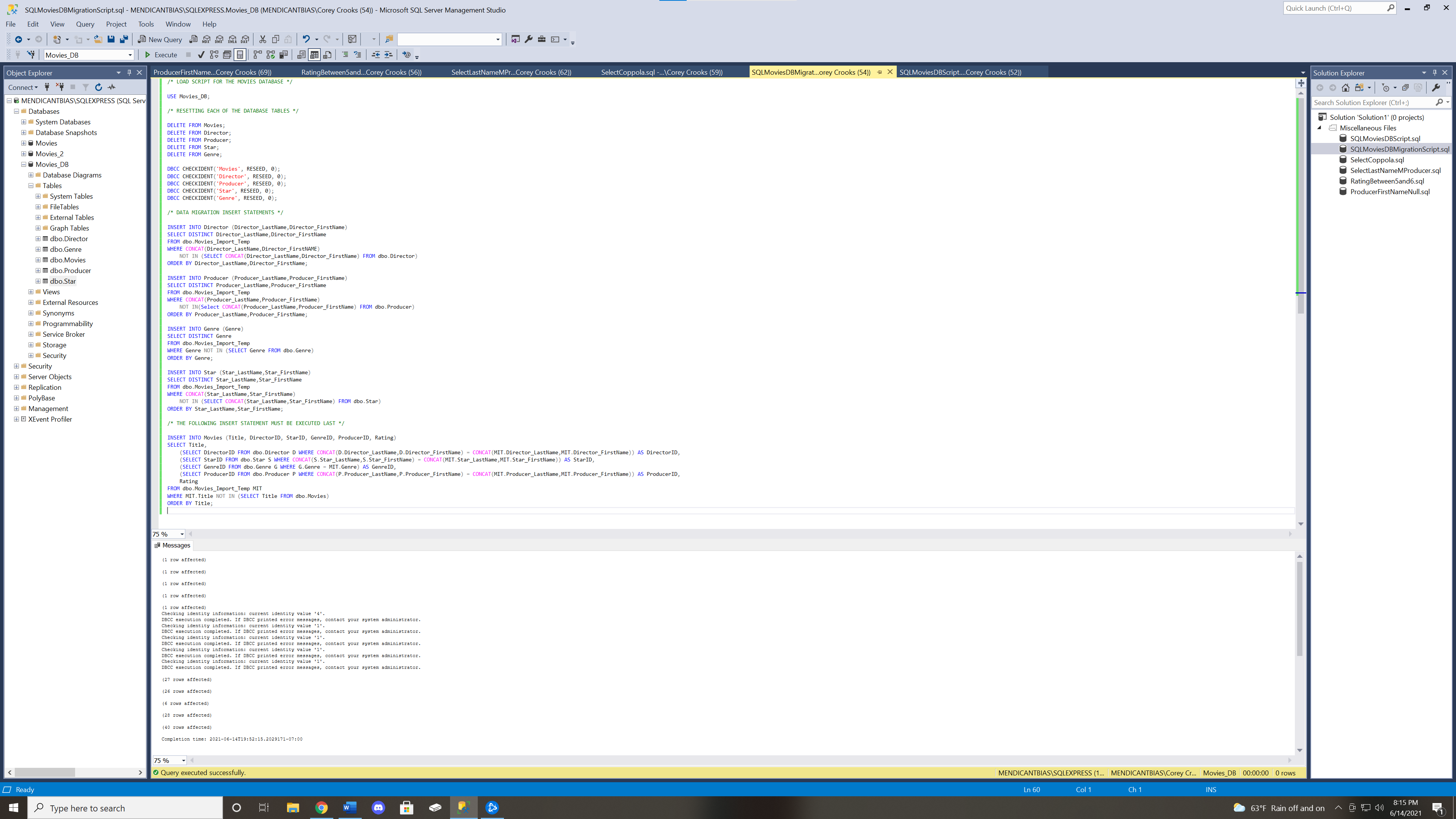
Producer

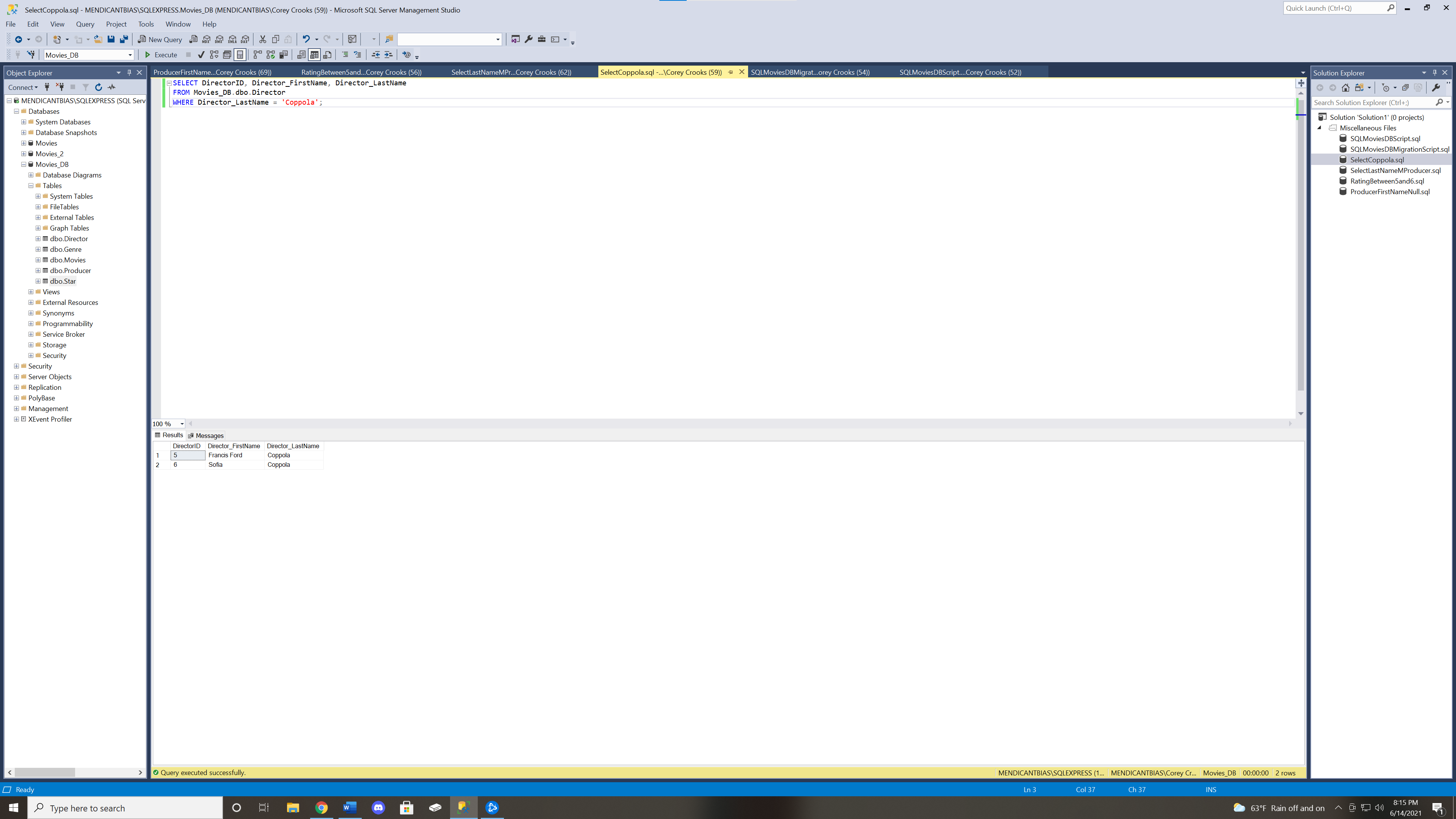


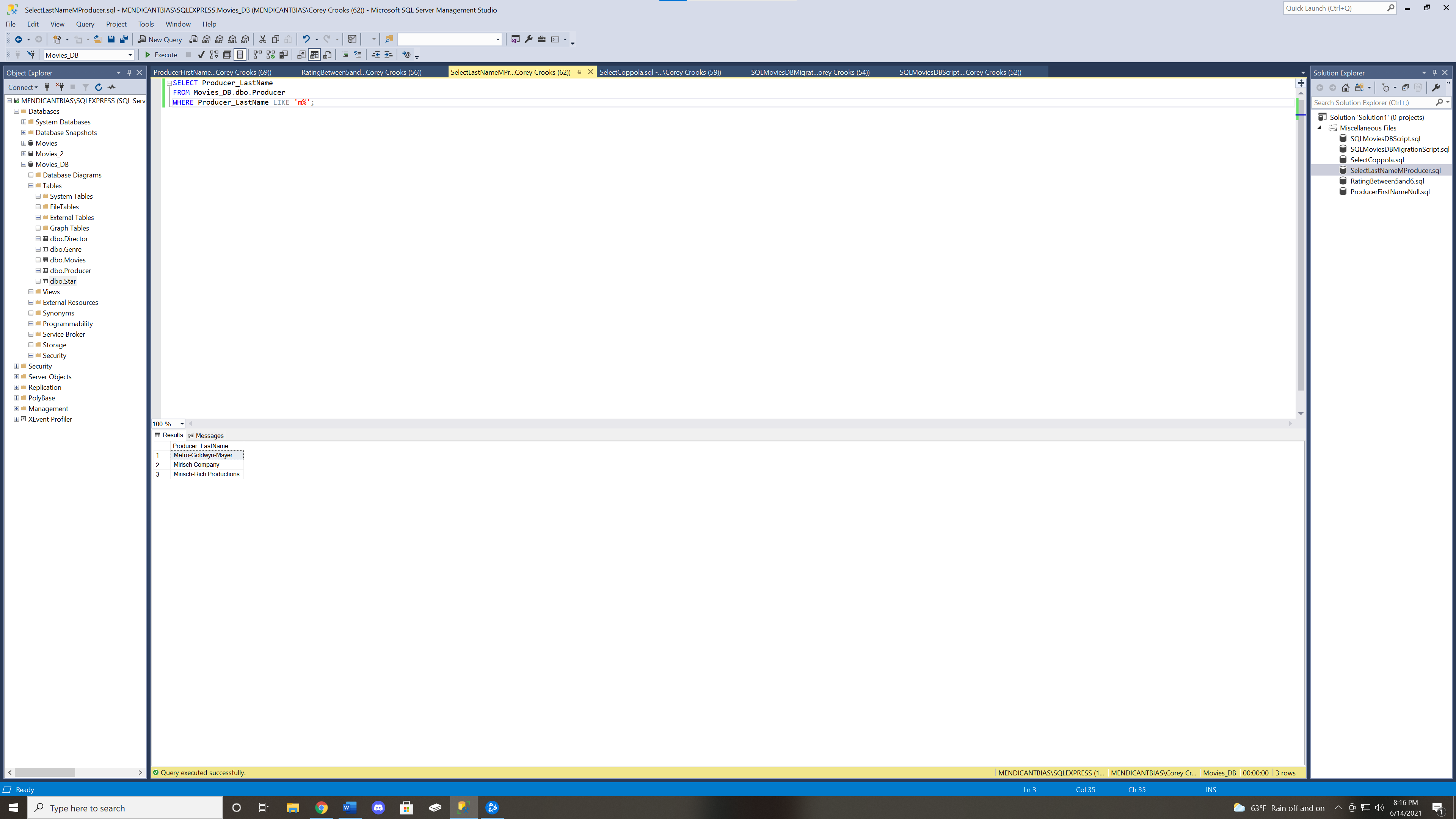
Star

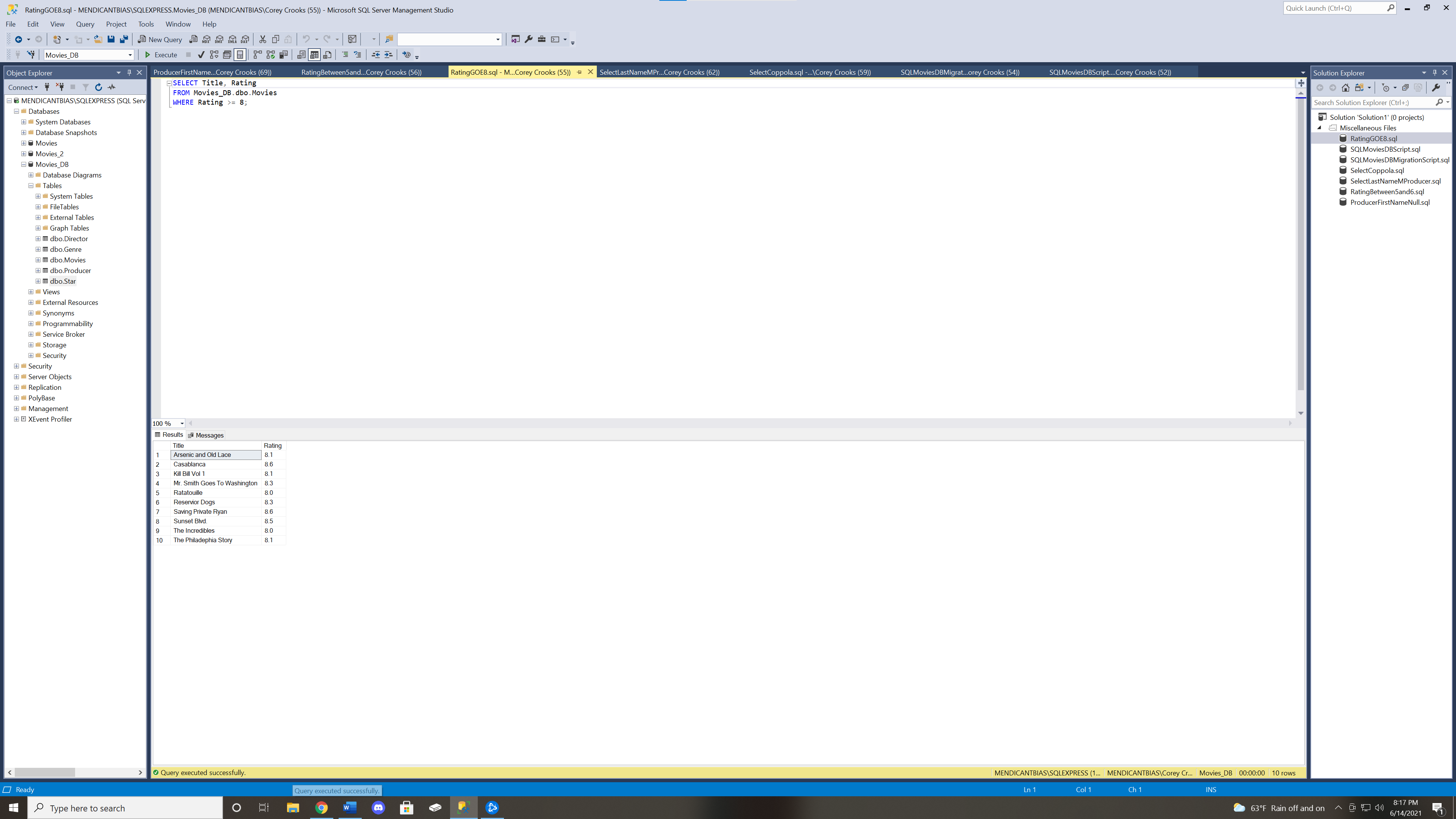
**Begin Code Proofs**

Creation Script Successful completion.

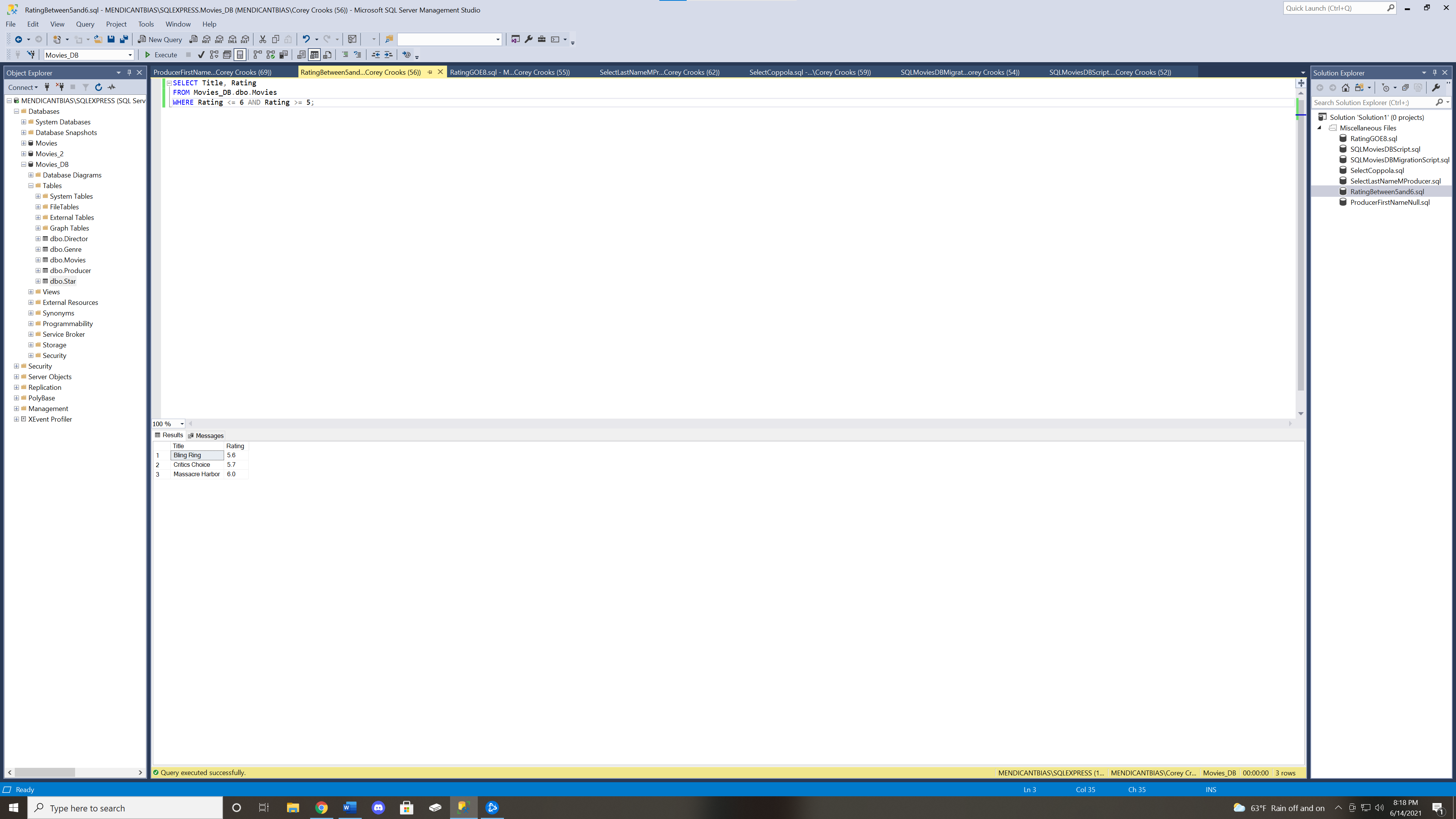
Migration Script Successful completion.

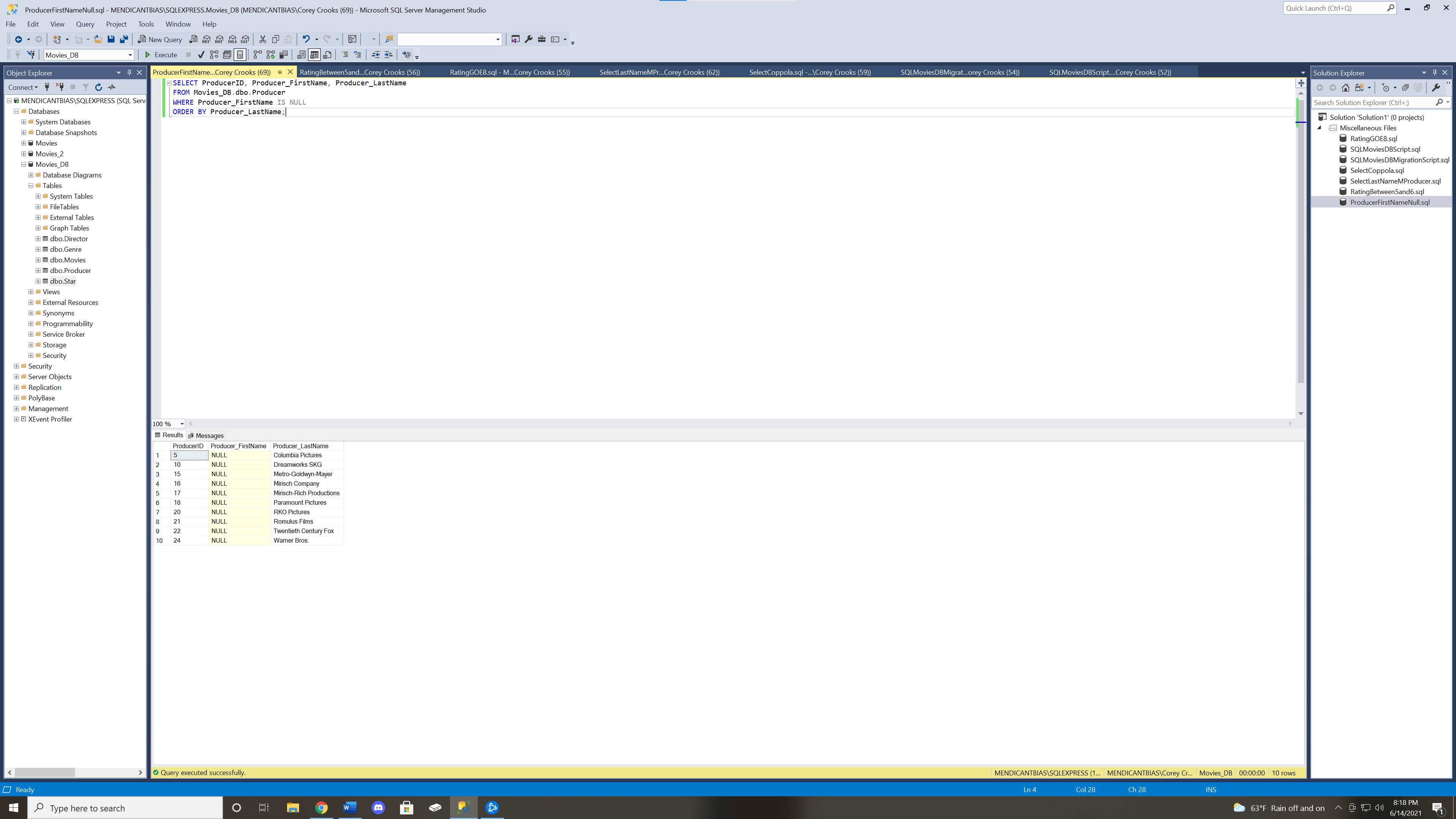
Select directors with last name Coppola successful completion.

Select Producer Last Name starting with M successful completion.



Select Movies with Rating Greater than or equal to 8 successful completion.

Rating between 5 and 6 successful completion.



Producer First name is NULL successful completion.

**Begin Writing Responses**

The individual INSERT statements have quite a bit of utility to further expand your existing database. According to a popular programming language learning website, “The INSERT INTO statement is used to insert new records in a table.” (W3Schools, 2021). This could be useful in nearly every database you may work with. If you record thousands of records over the course of the day, then all that information is going to be excruciating to input by hand. The INSERT INTO statements found in the SQL Data Migration script aim to take the burden off of you, and input all those records for you. When recording values for your database, it is easy to order them in a way that could be exported as a csv file, and then using a combination of scripts, you could automate data entry like we’ve done in this assignment. The INSERT INTO statements are at the heart of that.

Let’s take a look at the different methods available. Firstly, we have manual inputs. As described above, this style definitely has its fair share of drawbacks. Manually inputting data could be just fine if you are doing small, incremental records by only a few at a time, but as soon as the numbers start growing, the entire process is going to get incredibly tedious incredibly fast. That being said it does have its pros, too. Incorporating the human element into any technological stream helps to solve a number of problems that could arise with computers. Firstly, humans are mobile. If you are able to upload a database to the cloud, a human could access this and input records from anywhere. The script you write can only input records locally. Which means if there’s a power outage at the shop, you’d have better luck with a human writing records down on his machine, and pushing the information to a server when the outage is resolved.

The second method to be discussed is automated inputs. This method uses a script written by a worker to integrate a file output into the database in question. It may seem like this style may be the best case for nearly everybody, but I assure you this may be more situational than you realize. The pros of this style nearly speak for themselves. With automation, a business doesn’t need to pay an employee to have the mundane task of being in front of their keyboard and monitor 8 hours a day to input names, numbers and other values into your database. With automation, you could save money on working costs, and reduce errors from typos. The cons are sometimes easily overlooked, however. It is simple to write a single mistake in code. Maybe you reference a temporary location for the database you’re using. Maybe you accidentally make an ID a static number, rather than one that’s auto-generated per record. The issues are wide in variety, and immense in potential damage. If a mistake isn’t caught early enough, then you risk needing to rewrite your entire database.

# **References**

W3Schools. (2021, June 14). *SQL INSERT INTO Statement*. Retrieved from w3schools.com: https://www.w3schools.com/sql/sql\_insert.asp