SE3309a - Assignment Three

Samuel Mallabone - 250844429

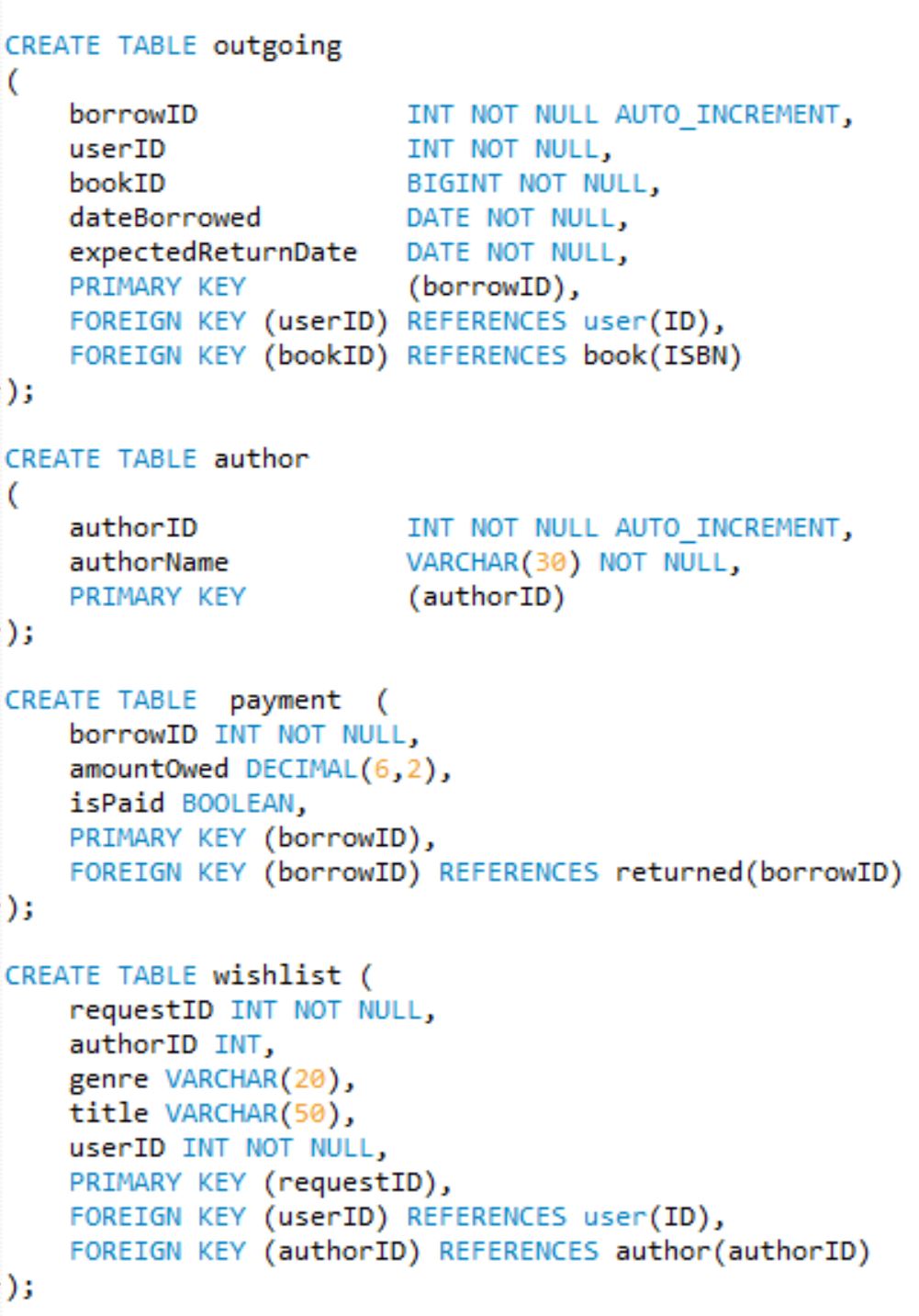
Robert Northmore - 250838145

Jak Terpak - 250846924

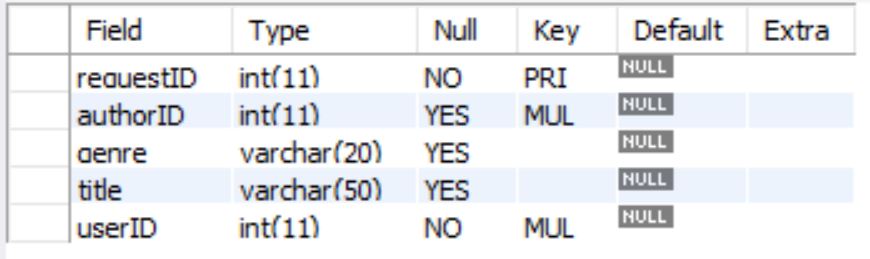
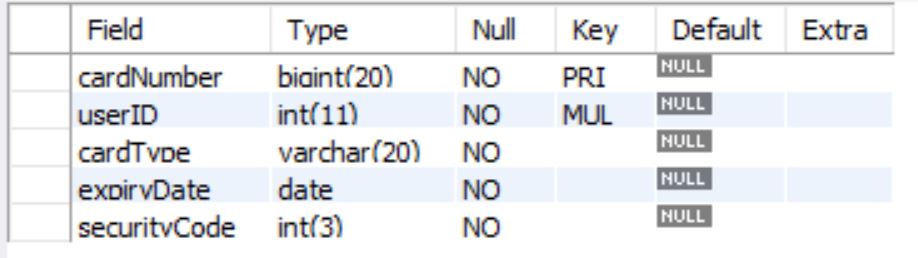
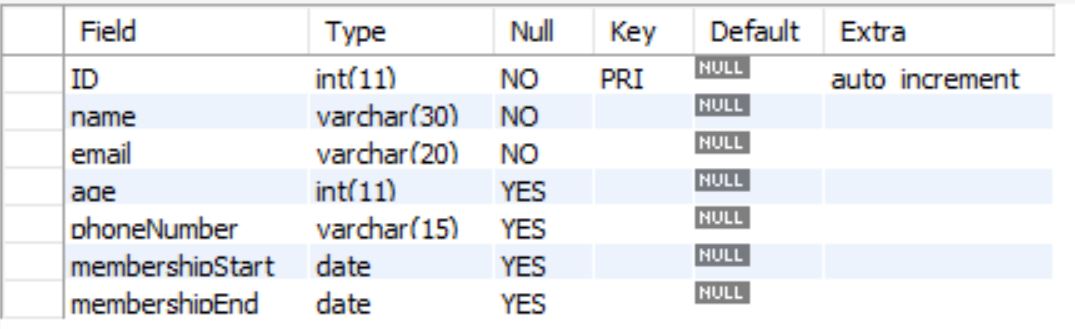
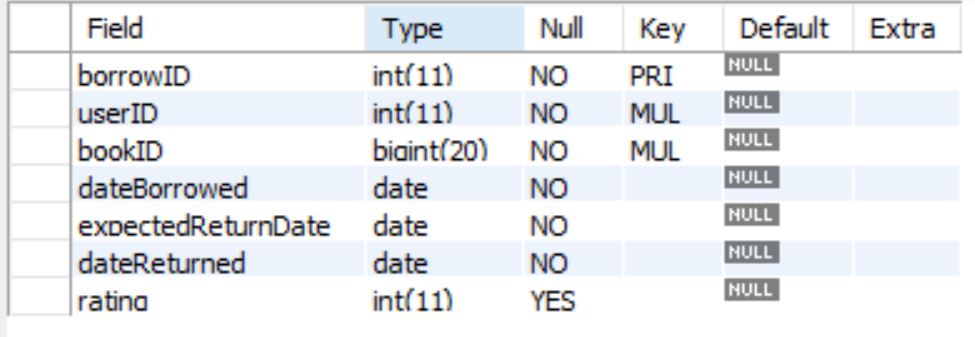
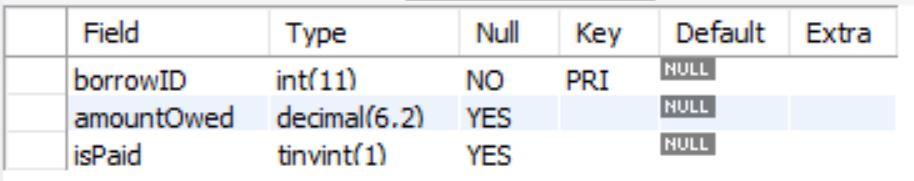
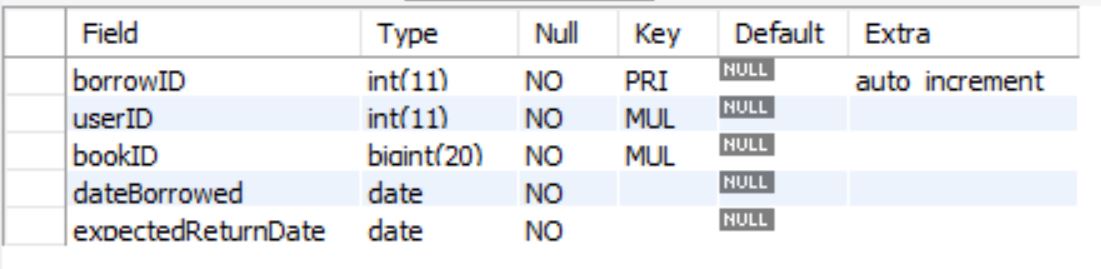
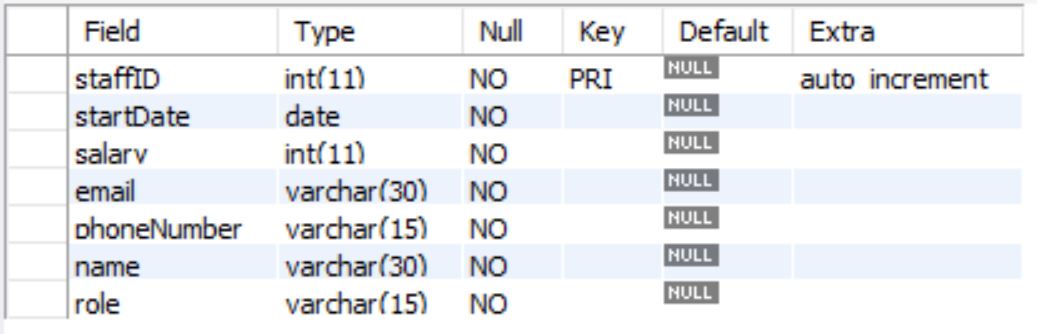
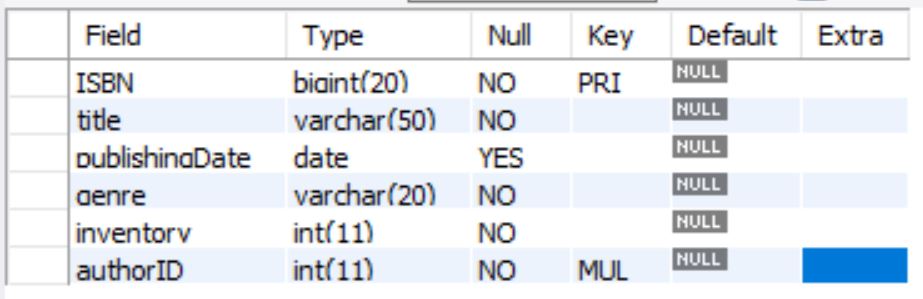
Craig Cook - 250866685

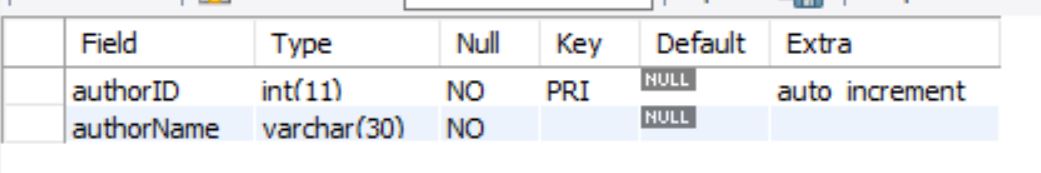
Question 2: Creating Tables:

*The top three screenshots showed the commands to create the tables. All these commands ran properly and created the table as we intended. Below the created tables, we used the describe <table\_name> command to show the tables were successfully created and had the schema we designated.*



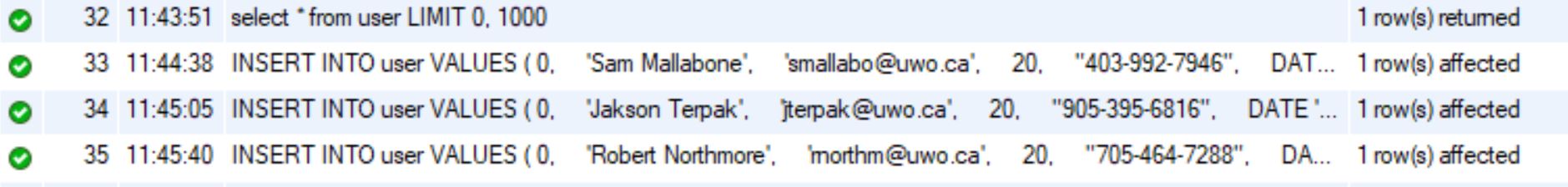
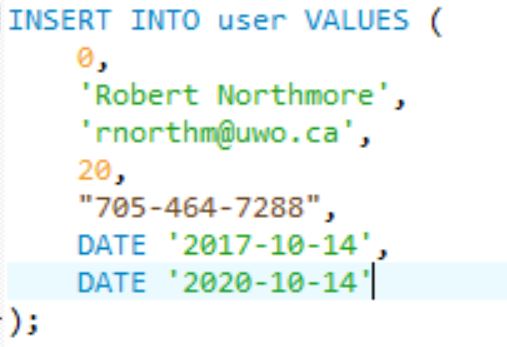
*This section is when we were running the describe <table\_name> commands*

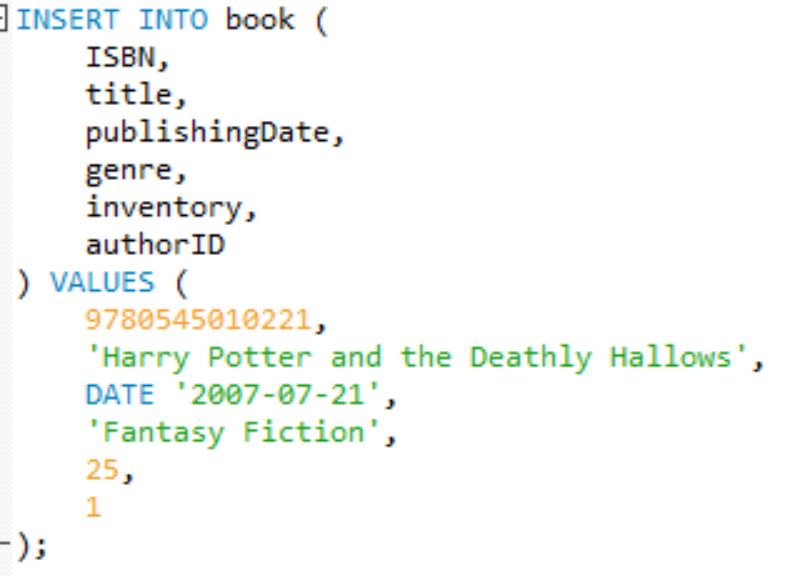


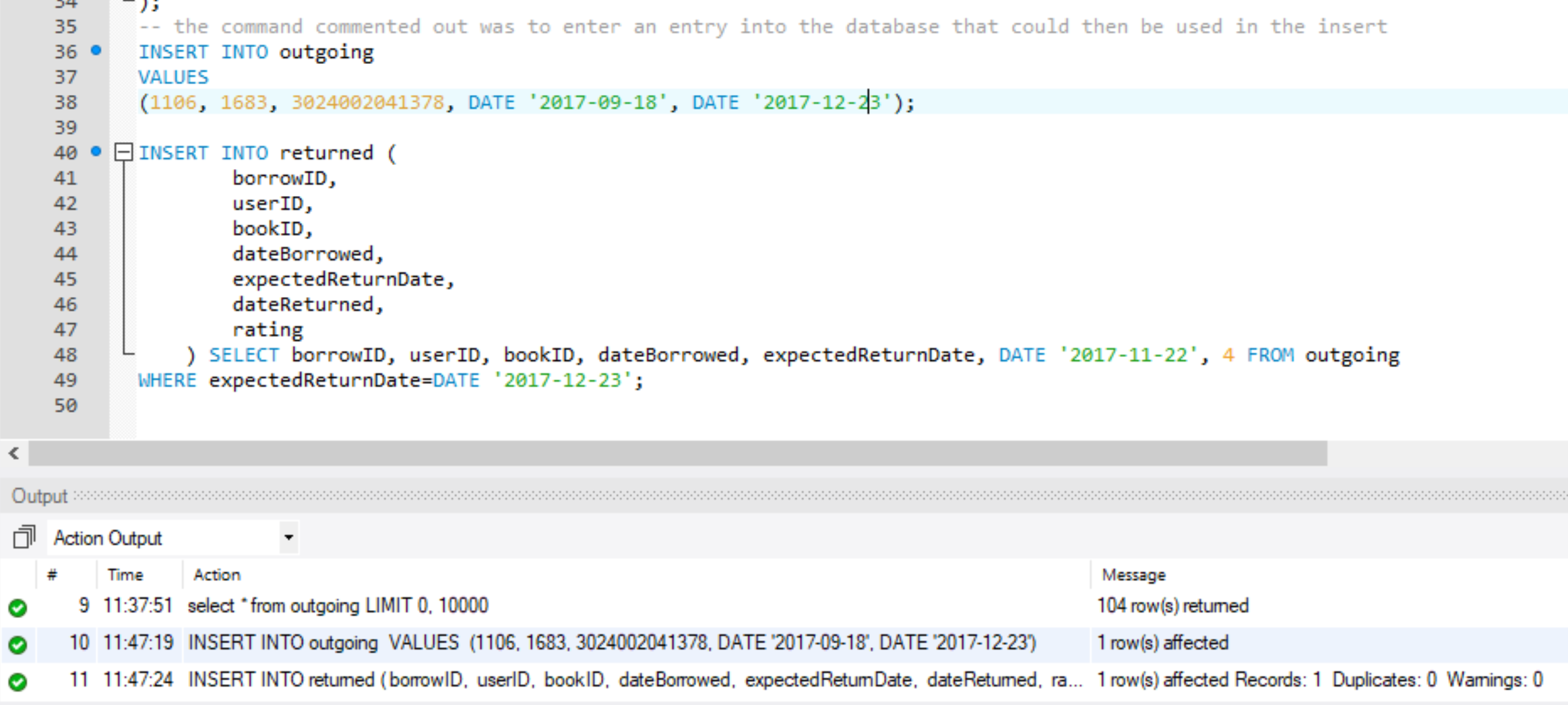


Question 3: Three different types of Insert

*Below shows the three different insert commands for entering data into tables. All three of these different types of commands worked for us and the screenshots show both the command and the output. Instead of running the select \* statement for the table to show that the values had been entered, we showed the success statement of the output because we thought this would exemplify that the command worked as intended.*





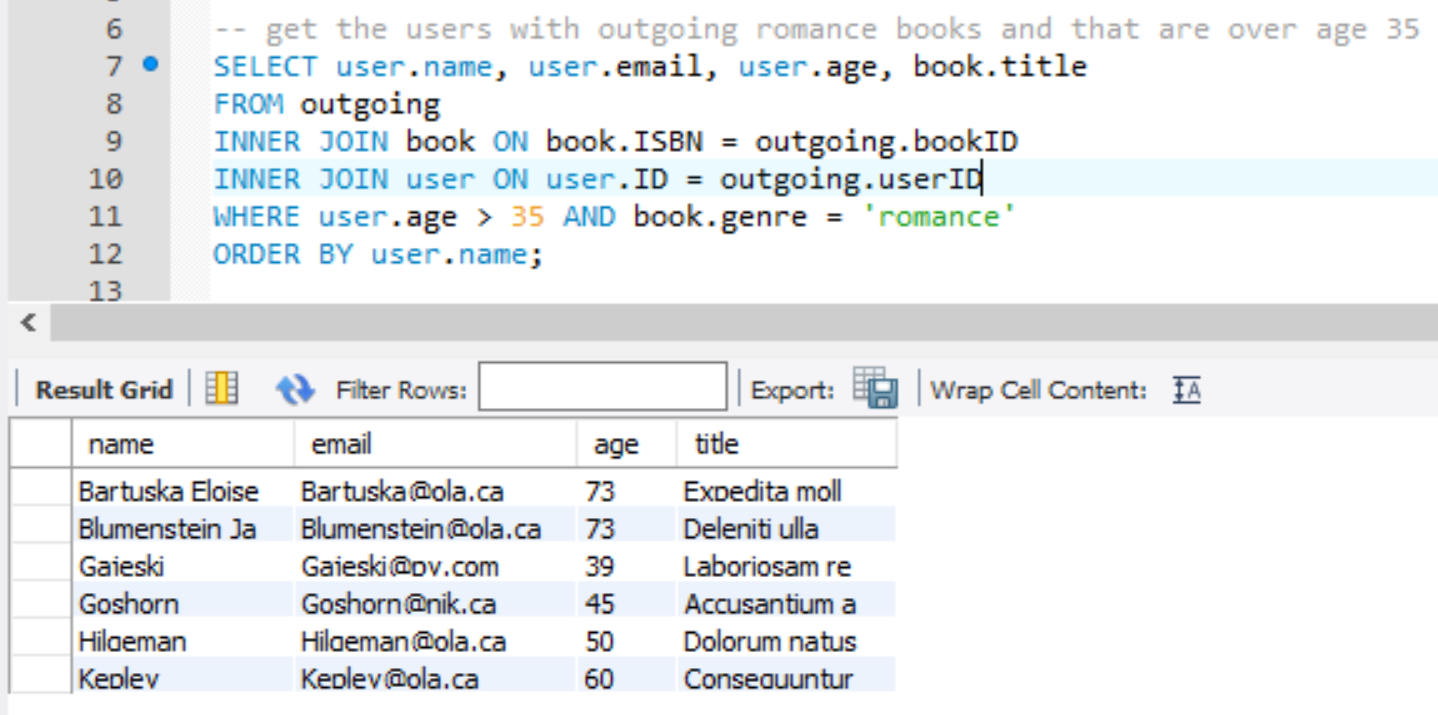


Question 4: Inserting data into the tuples.

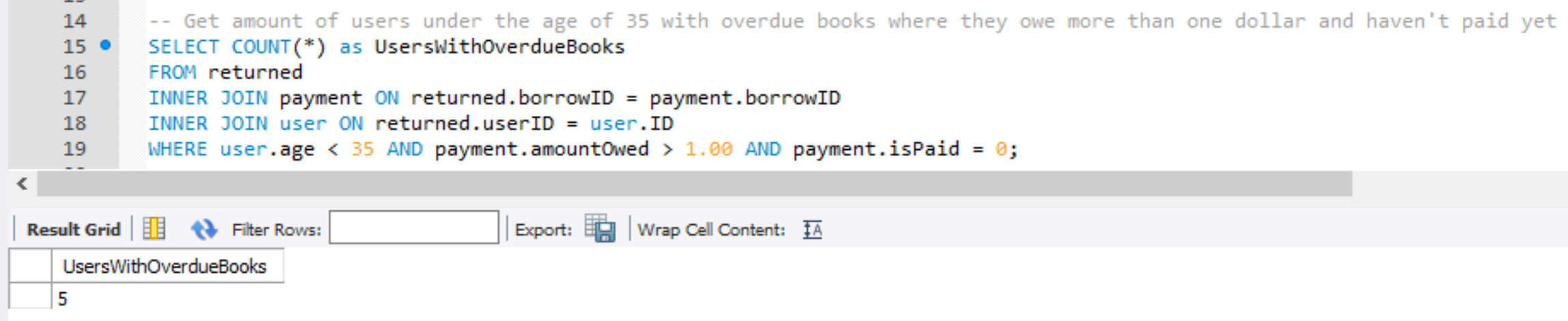
*To insert our data into the tuples, we used python (the files name is insertmysql.py). This python file would run script that would enter data into the database. We ensured to create the tables in such a way that we could use existing primary keys from a created table for the foreign keys on another table. This ensured that we could properly use all the joins and our queries would run as expected. In the python file a lot of the code is commented out and in weird order. we would run several sections of the code to insert the data and that is why the file is in the order it is. It worked perfectly and we were able to enter over 5000 tuples in two tables, a several hundred in another table and over one hundred tuples in all the rest. To get our data, we used a combination of csv files with names in them as well as a python library called Faker which allowed us to generate random names and random text which we took advantage of.*

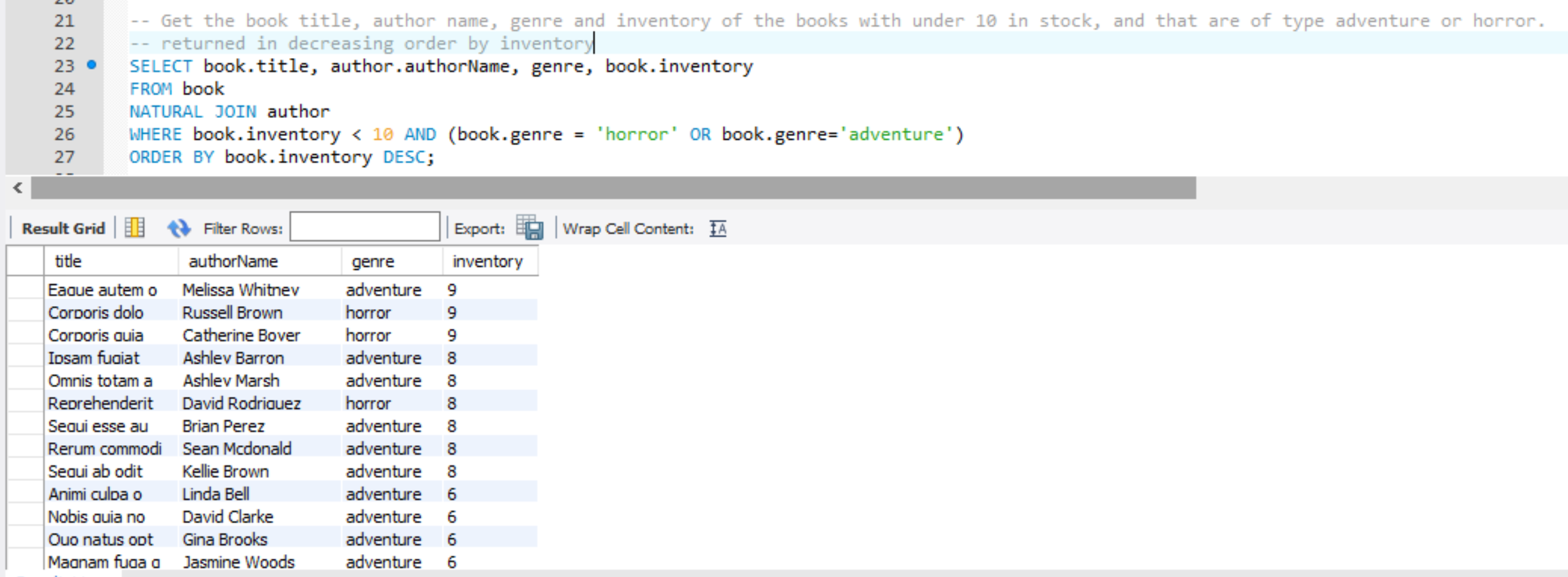
Question 5: Six Interesting queries

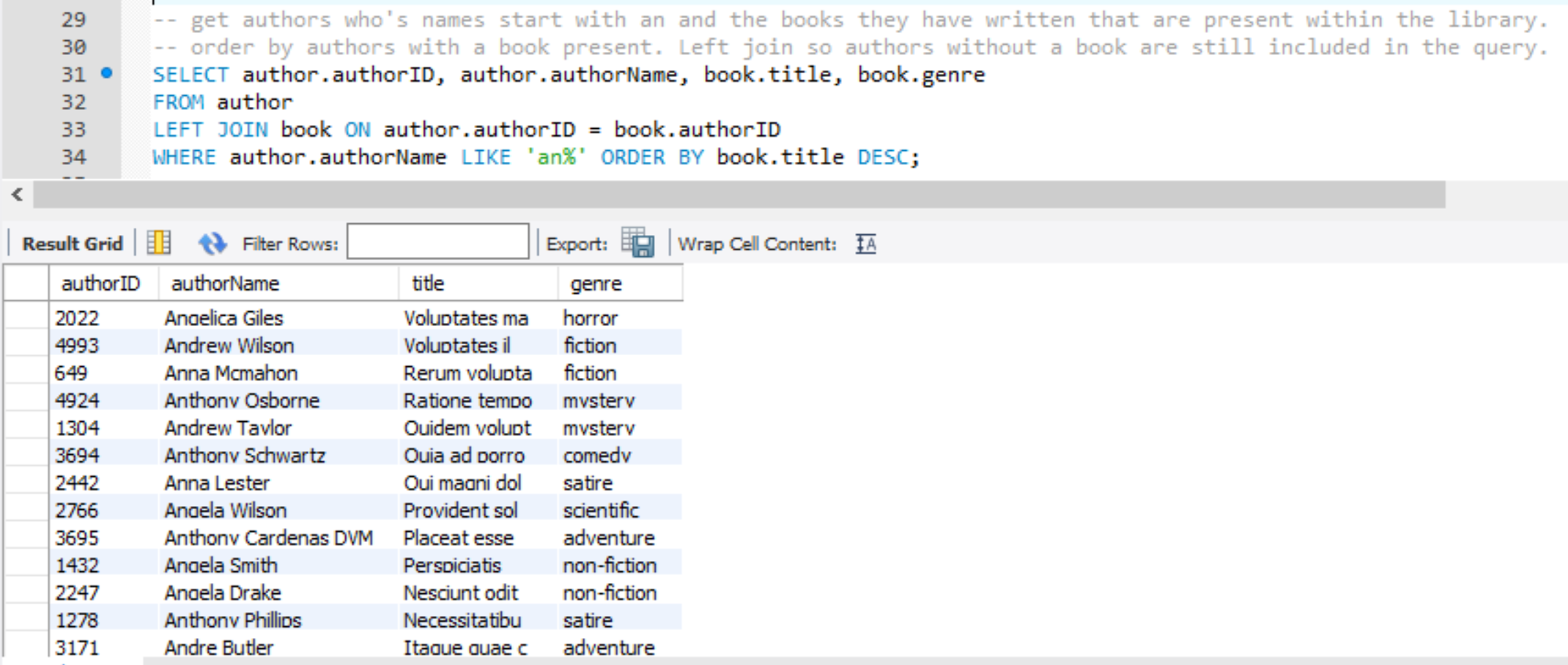
*Below are screenshots of our six interesting queries and the outputs that they result in.* 



*Note: a count was returned*







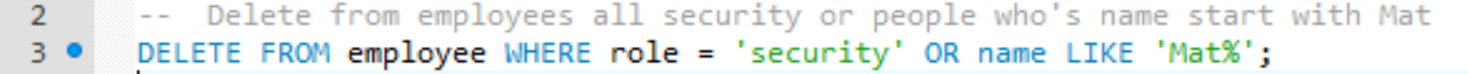
*Note: Only the one tuple was returned*



Question 6: Delete and Update

*Below shows three interesting modification commands. We choose to do one interesting delete and two interesting updates.*

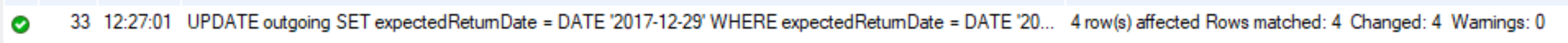
Delete:



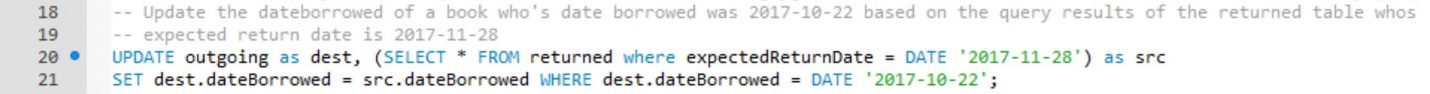


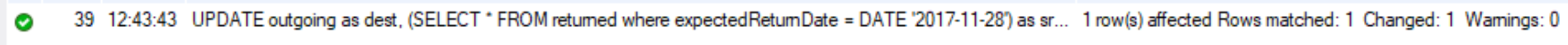
Update:



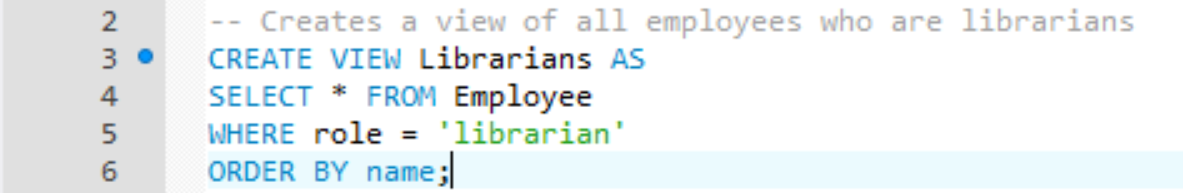


Update 2:

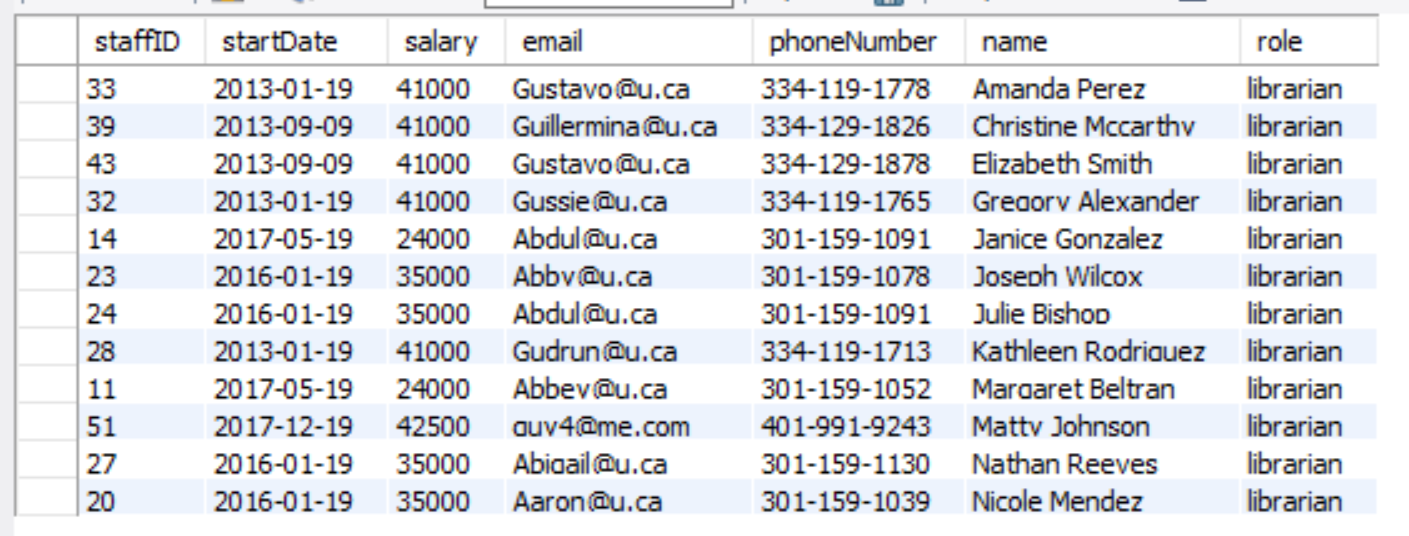




Question 7: Views



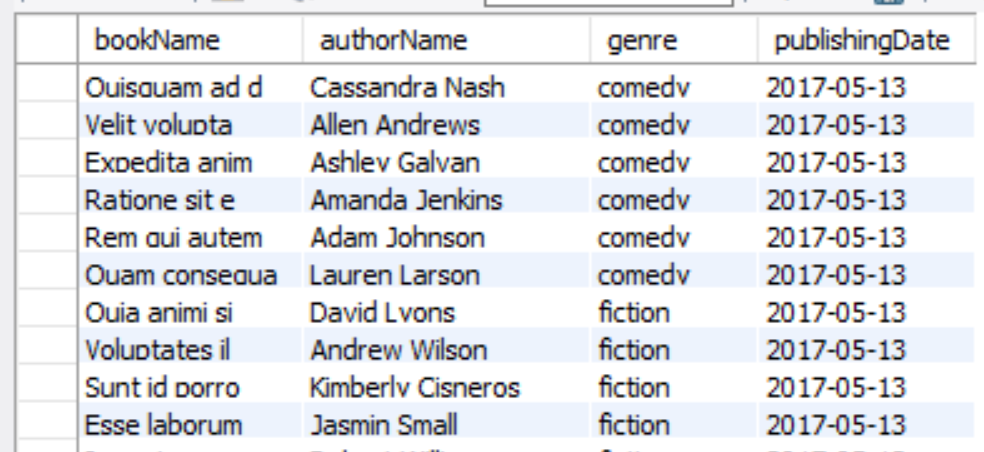
Result from querying all tuples in the above view



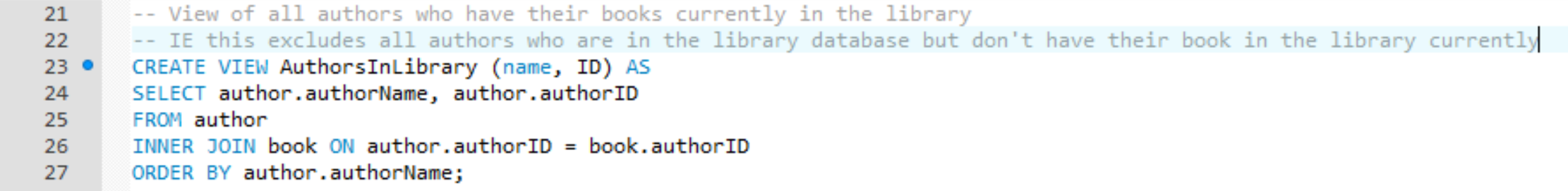
VIEW 2:

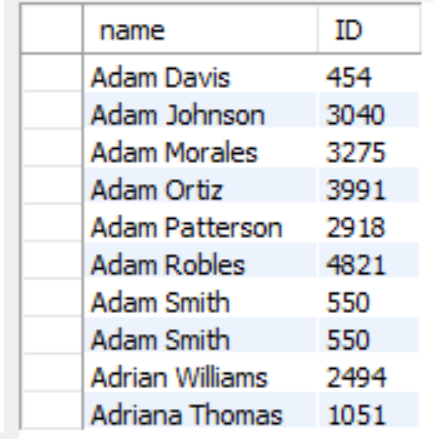


Result from querying all tuples within the view:



VIEW 3:





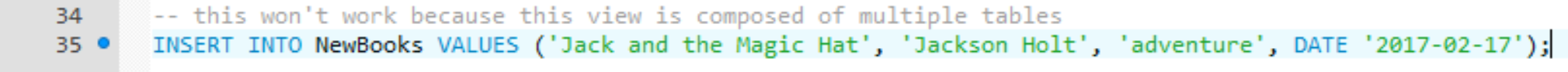
Inserting into views:

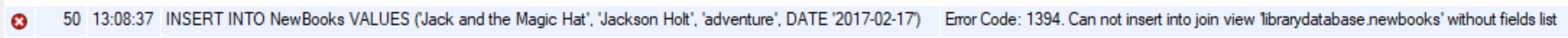
Working insert for view 1:





Not working insert for view 2:





The third view also will not be able to have values inserted because it is composed of a join of multiple tables. We chose not to include the screenshots of the failed insert because it is the same as the one above.

Question 8: Non-Existent MySQL clause.

The SQL clause that we learned in class but was not implemented in MySQL is the *assertion* statement. An assertion is a predicate expressing a condition we wish the database to always satisfy. Through our experience with MySQL we found out that this assertion was unavailable in MySQL and thus we would not be able to use it if necessary.