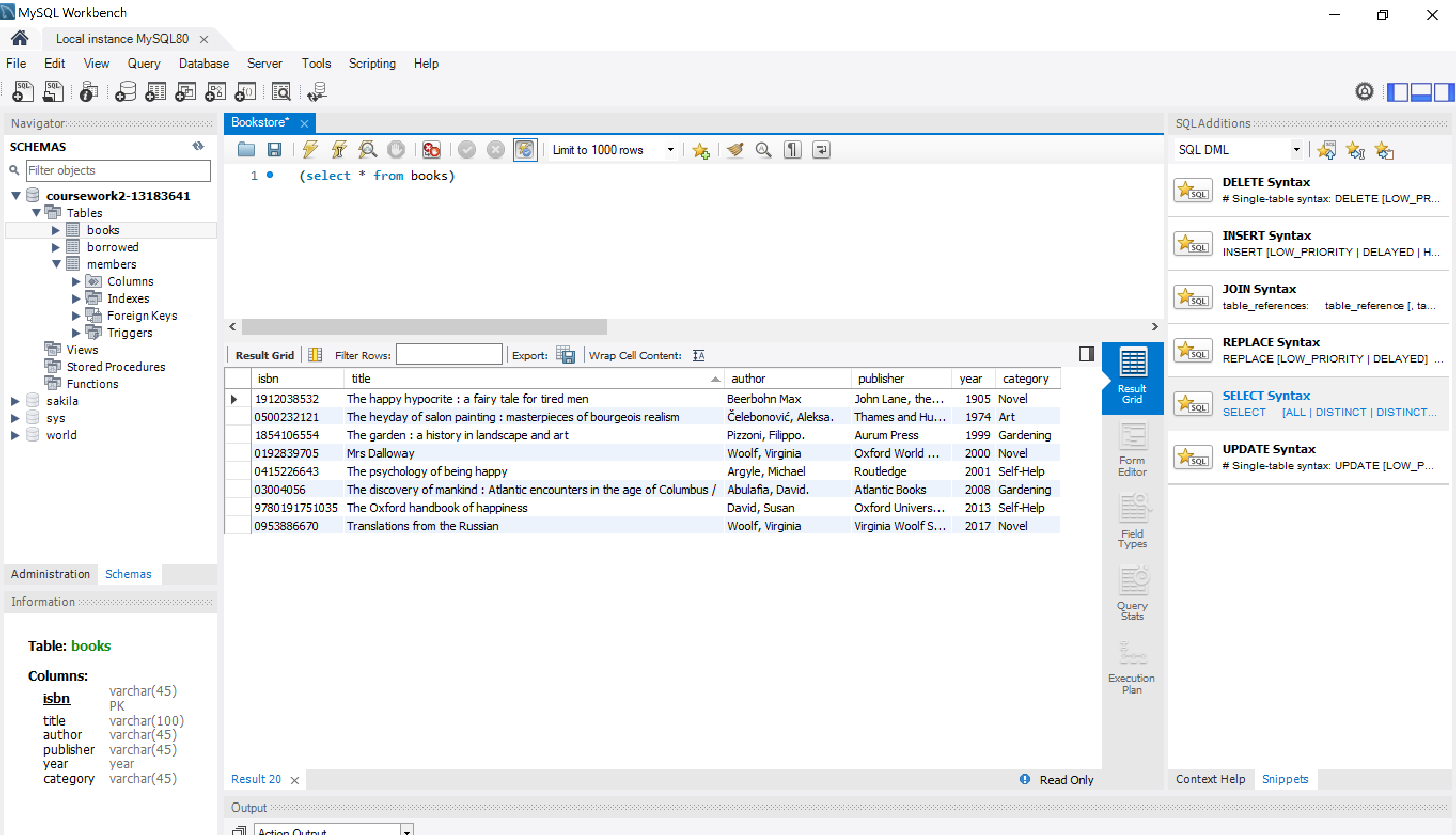
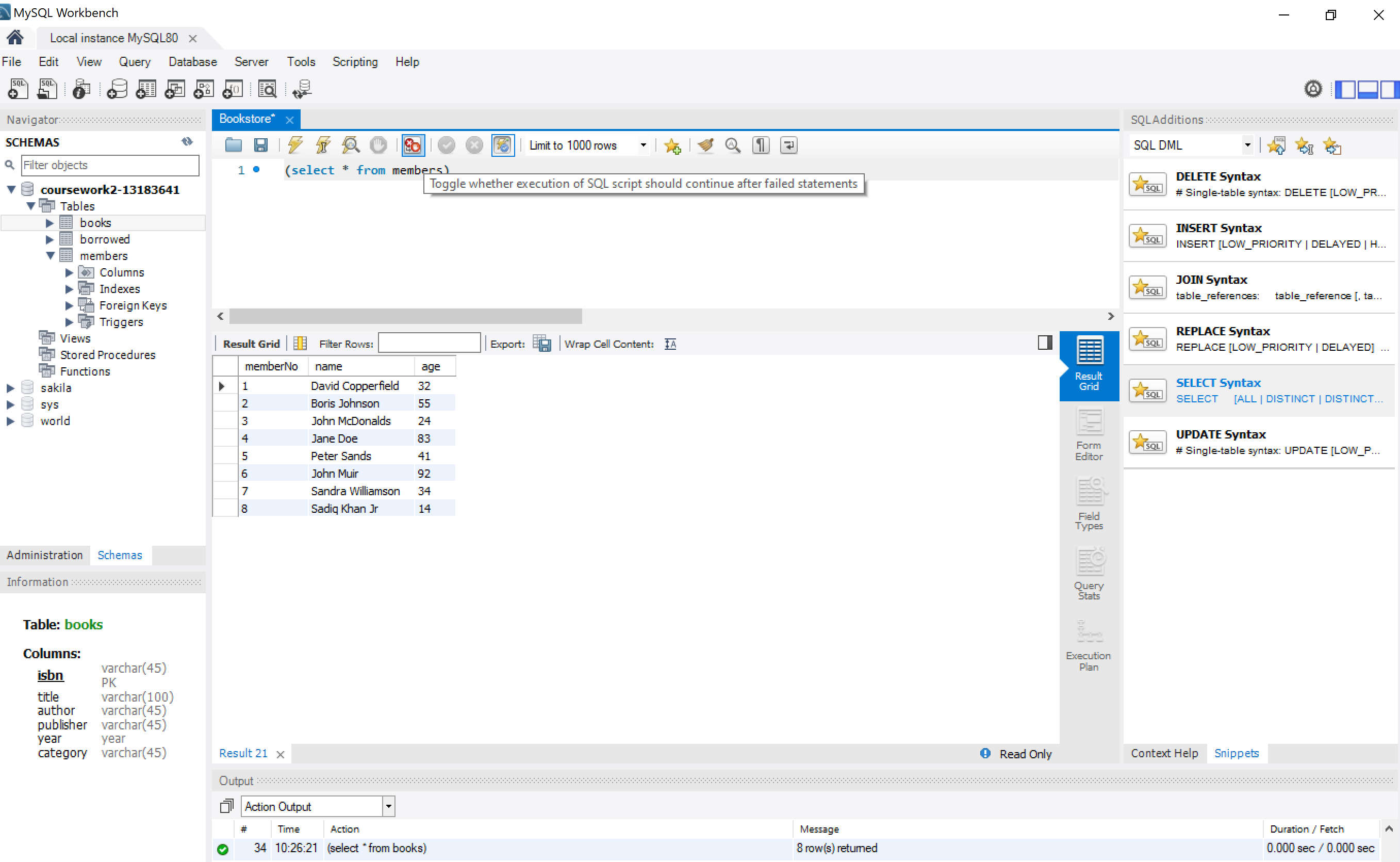
**Database Management - 2020**

**Part II: SQL Queries (50 marks)**

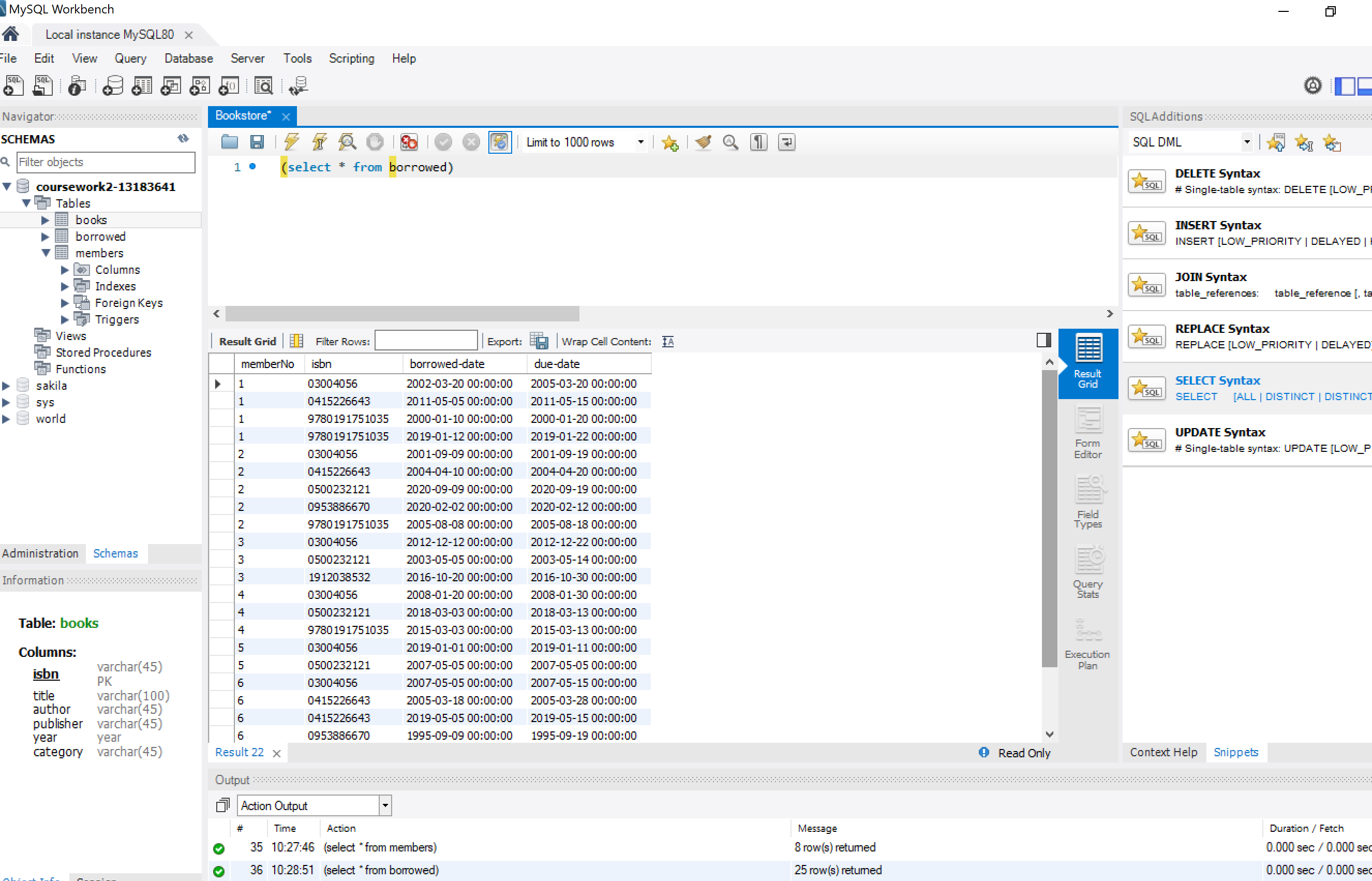
Books



Members

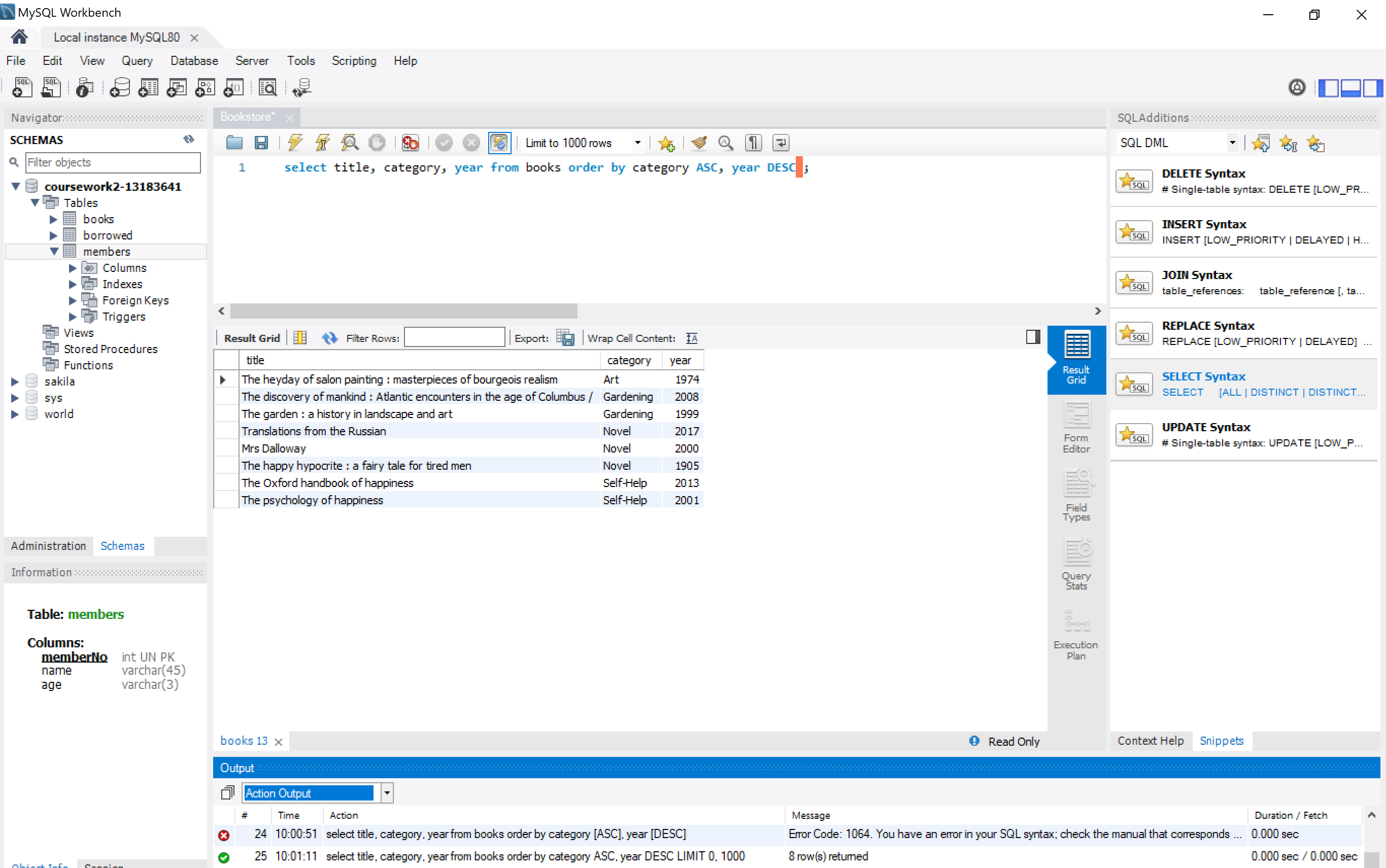


Borrowed



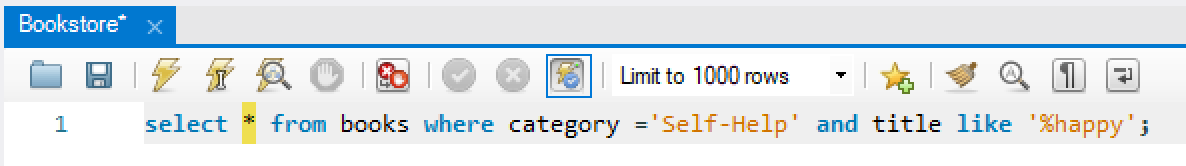
1. **List the title, category and year of publication of each book held in the library. The list should be ordered by ascending category, and within that by descending year of publication.**

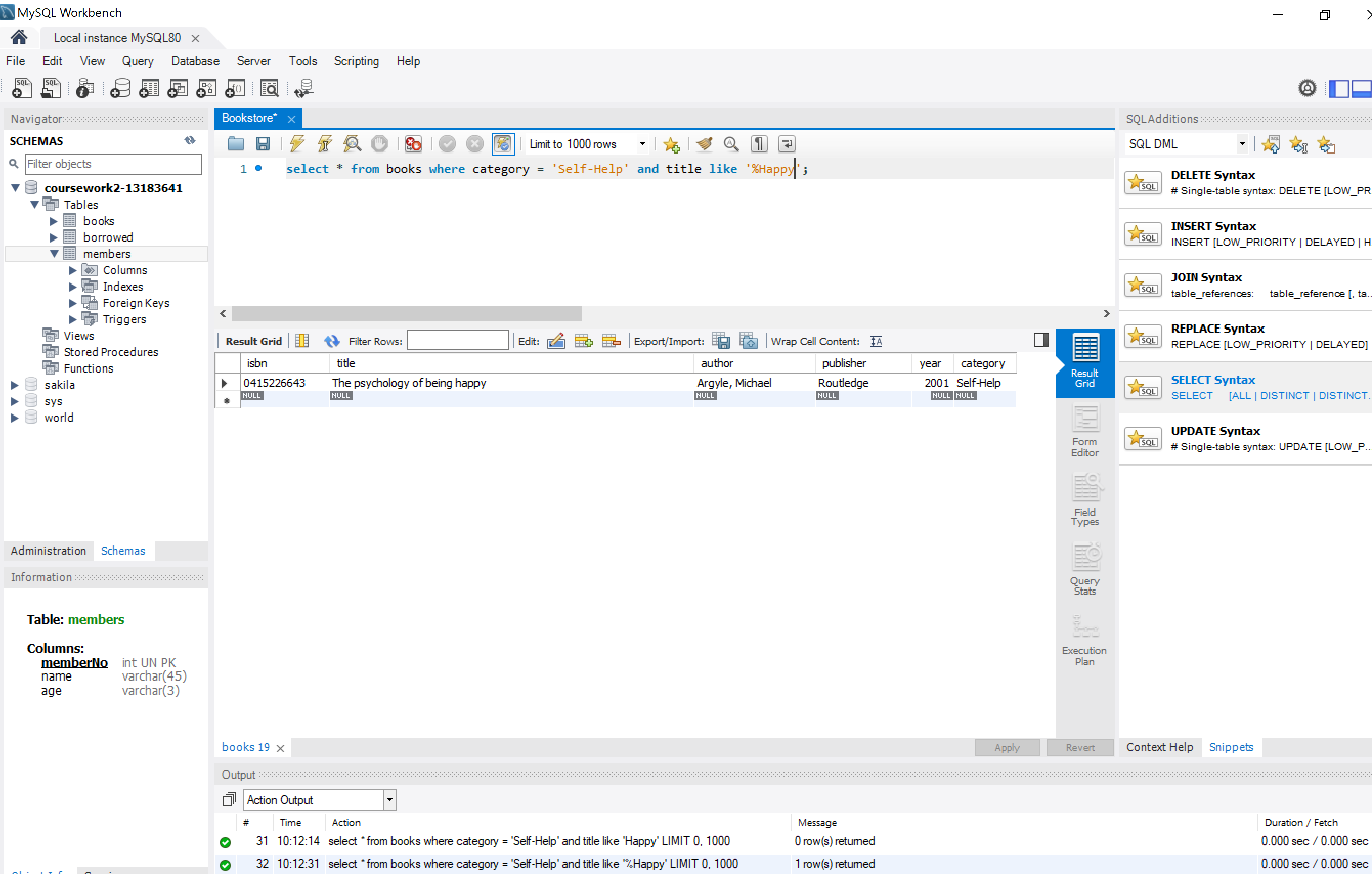
Simple selection of 2 elements from one table (books) ordered firstly by category and year as second criterion.



1. **Find the titles of books in the "Self Help" category that have "happy" somewhere in their title.**

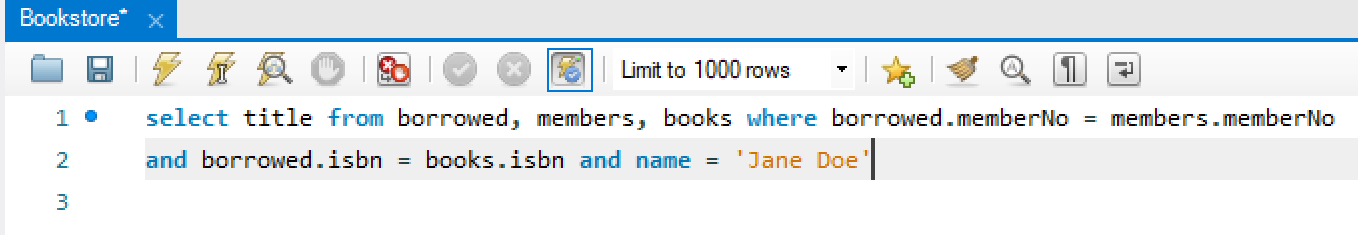
I select title (in the screenshot I have included \* all but it is exactly the same) from one single table (books) and put 2 conditions linked by the AND operator and I introduce like to find strings that include the word happy.

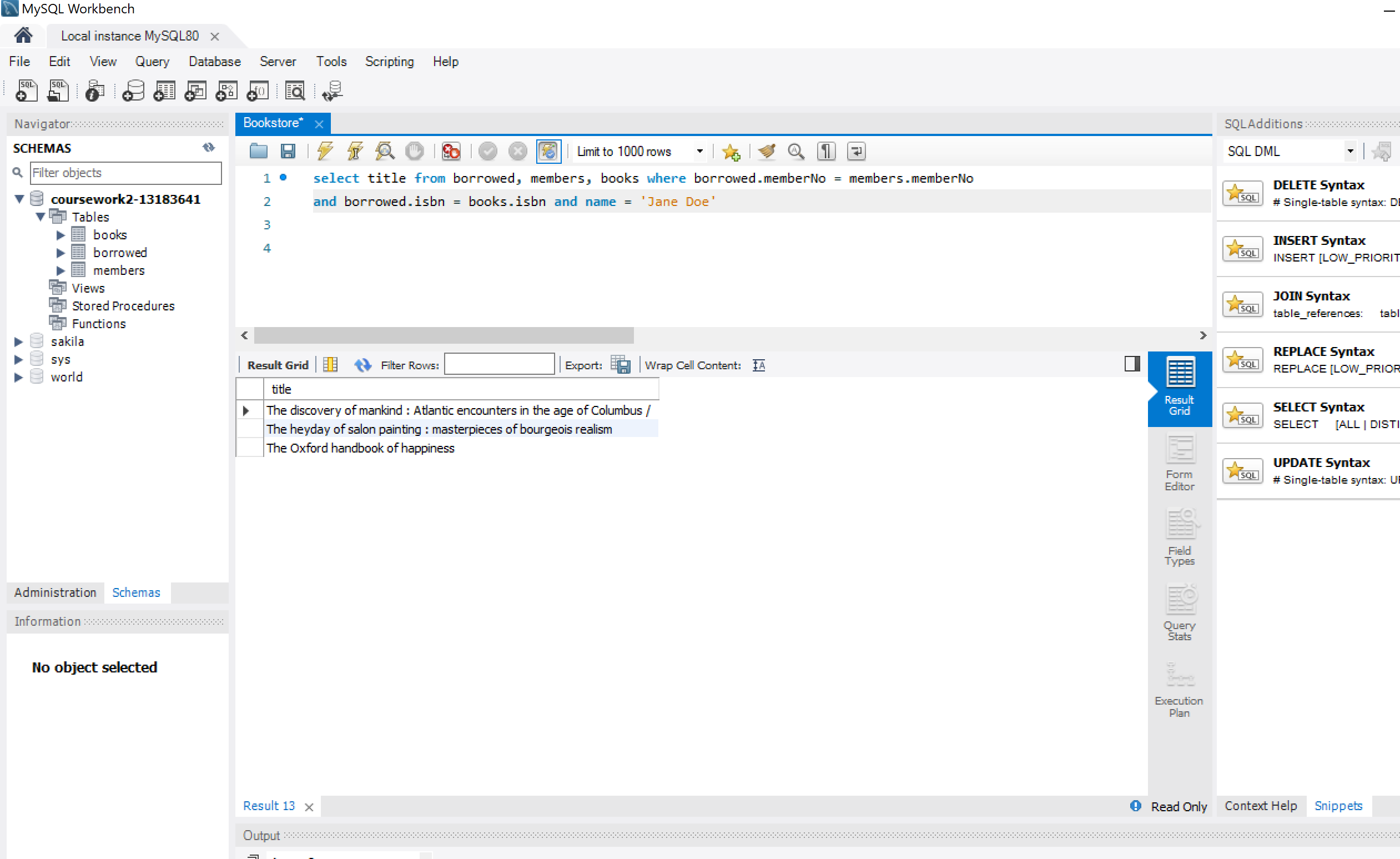




1. **Find the titles of books borrowed by "Jane Doe".**

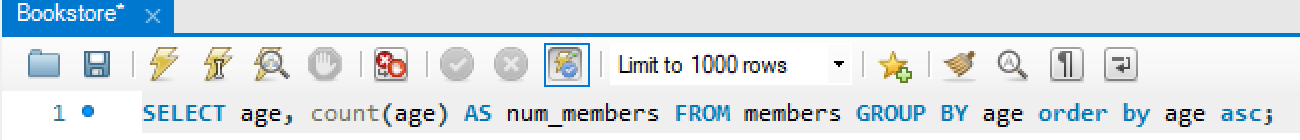
In this case I have to link the 3 main tables (books, members and operators) using members and isbn to discard duplicates and adding the additional condition that the name of the borrower has to equal the string ‘Jane Doe’.

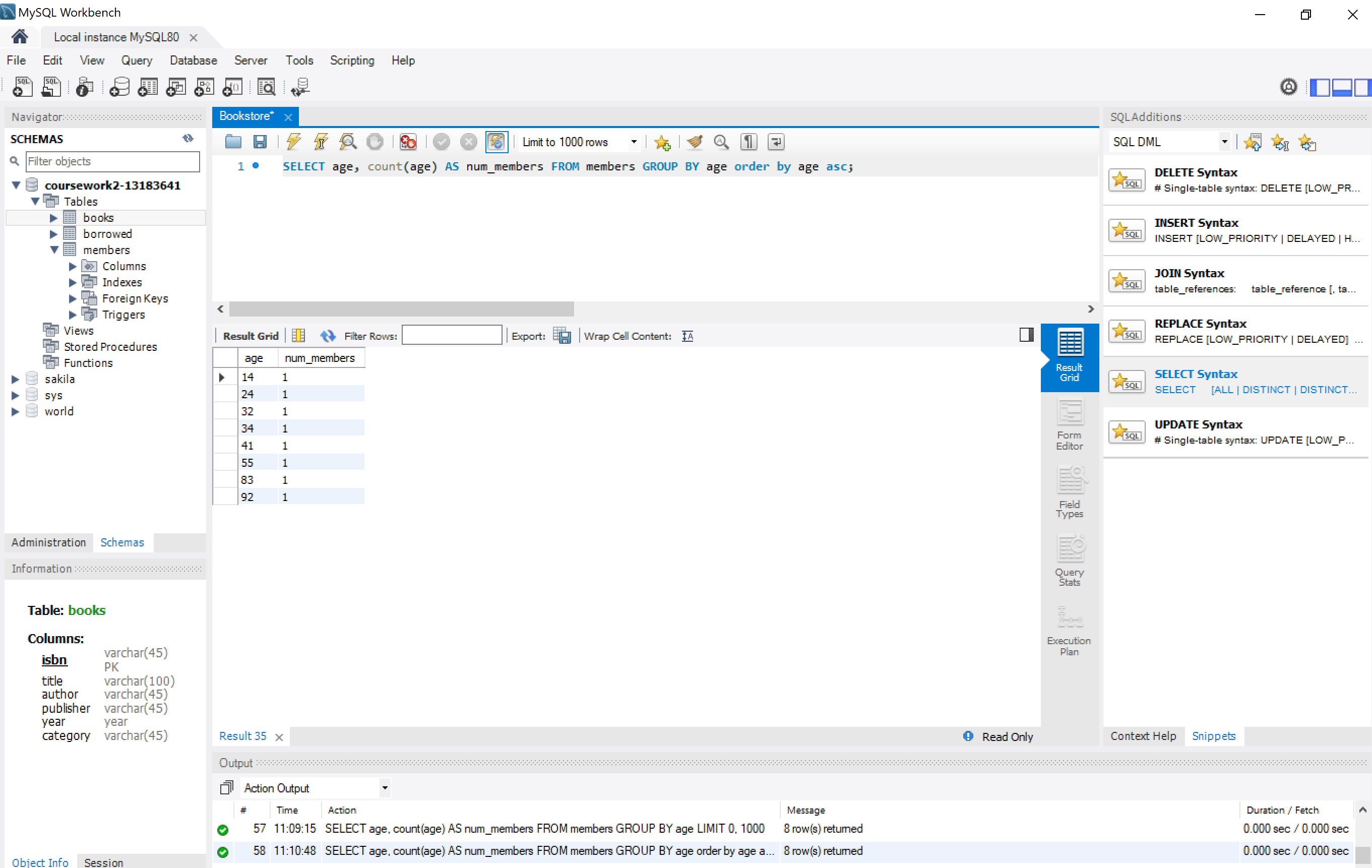




1. **Find the age profile of members, i.e., for each age, find the number of members of that age.**

I use count () to consolidate data from members adding them by age. I rename the column AS num\_members and finally, accordingly to our age consolidation I group the different ages and sort them in ascending order.

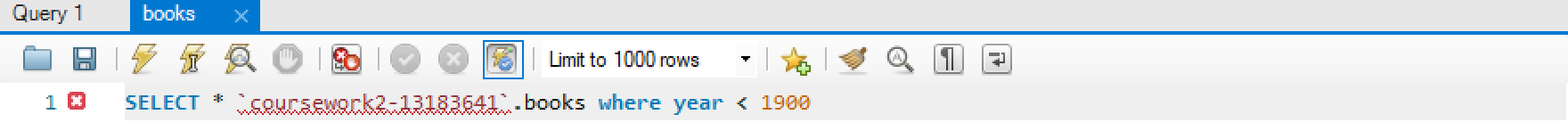


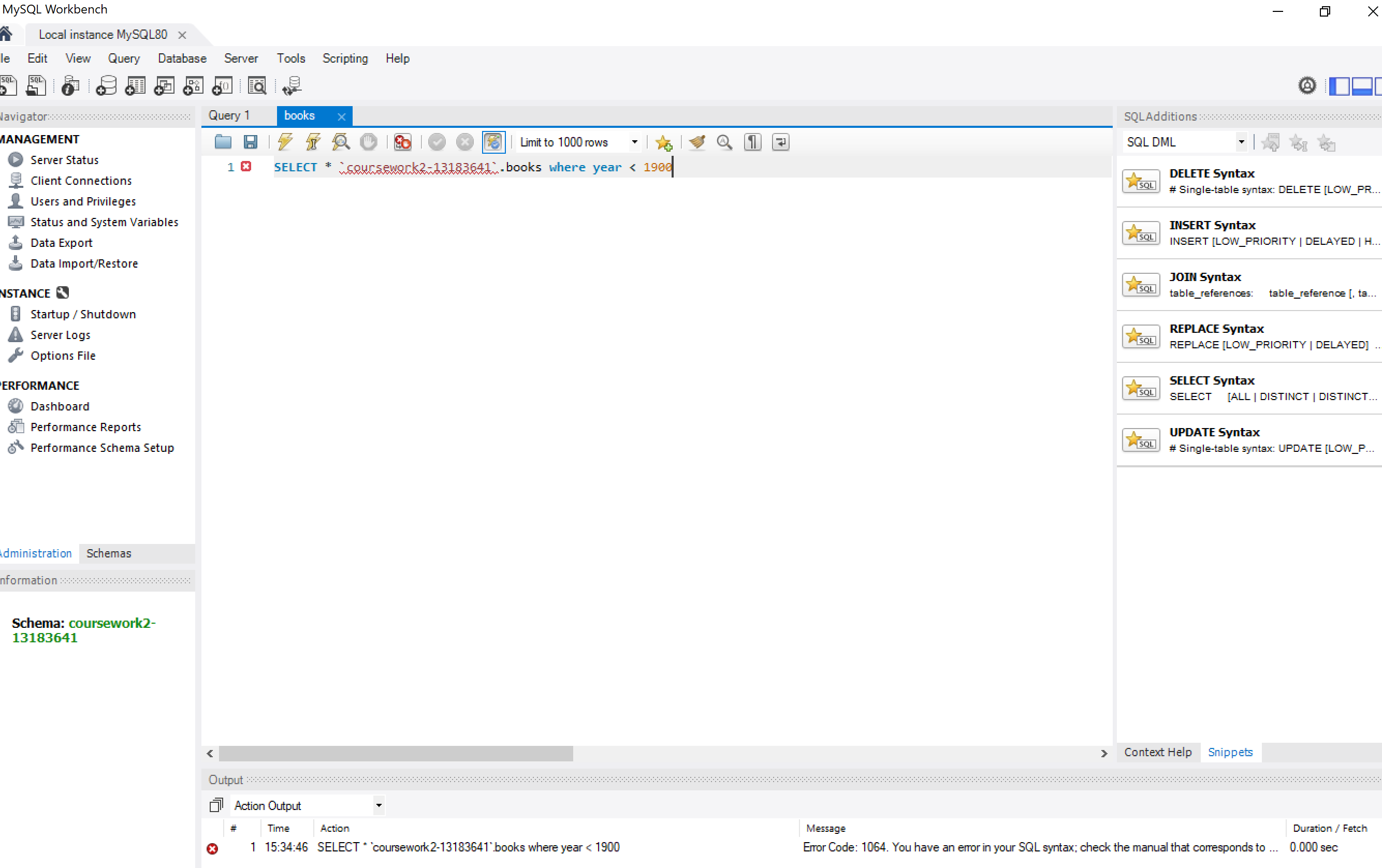


1. **Find the titles of books published before 1900.**

MySQL configuration does not allow selecting dates before 1900. I do not want to change Year configuration into varchar (it could have been a possibility). I should have changed the overall configuration of MySQL using date time instead of smalldate time.

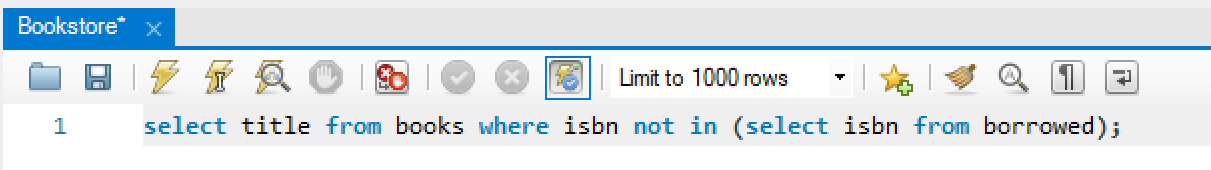
The query itself is not difficult at all: just select books in the DB and include the condition where year is less that 1900.

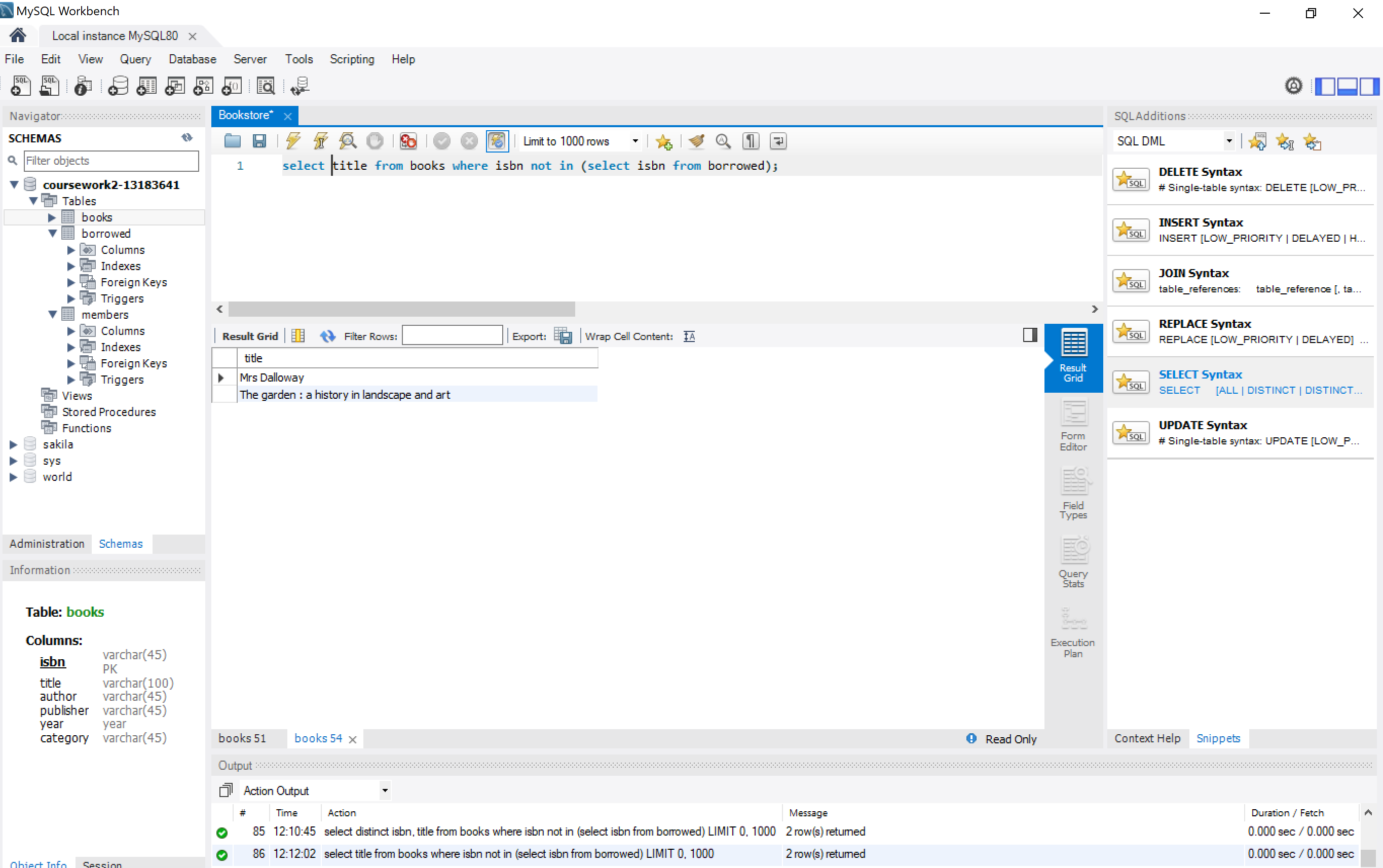




1. **Find the titles of books that have never been borrowed.**

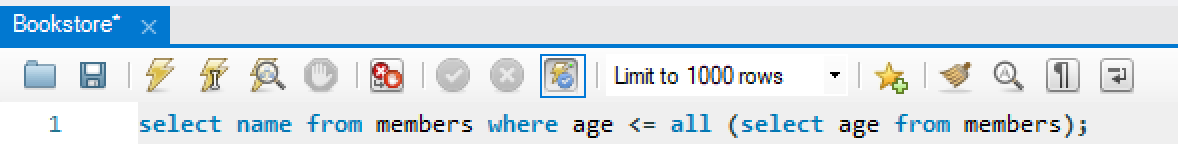
Books are identified by ISBN. I select the title from books table and add the condition that its isbn does not match any included in the borrowed tables using NOT IN.

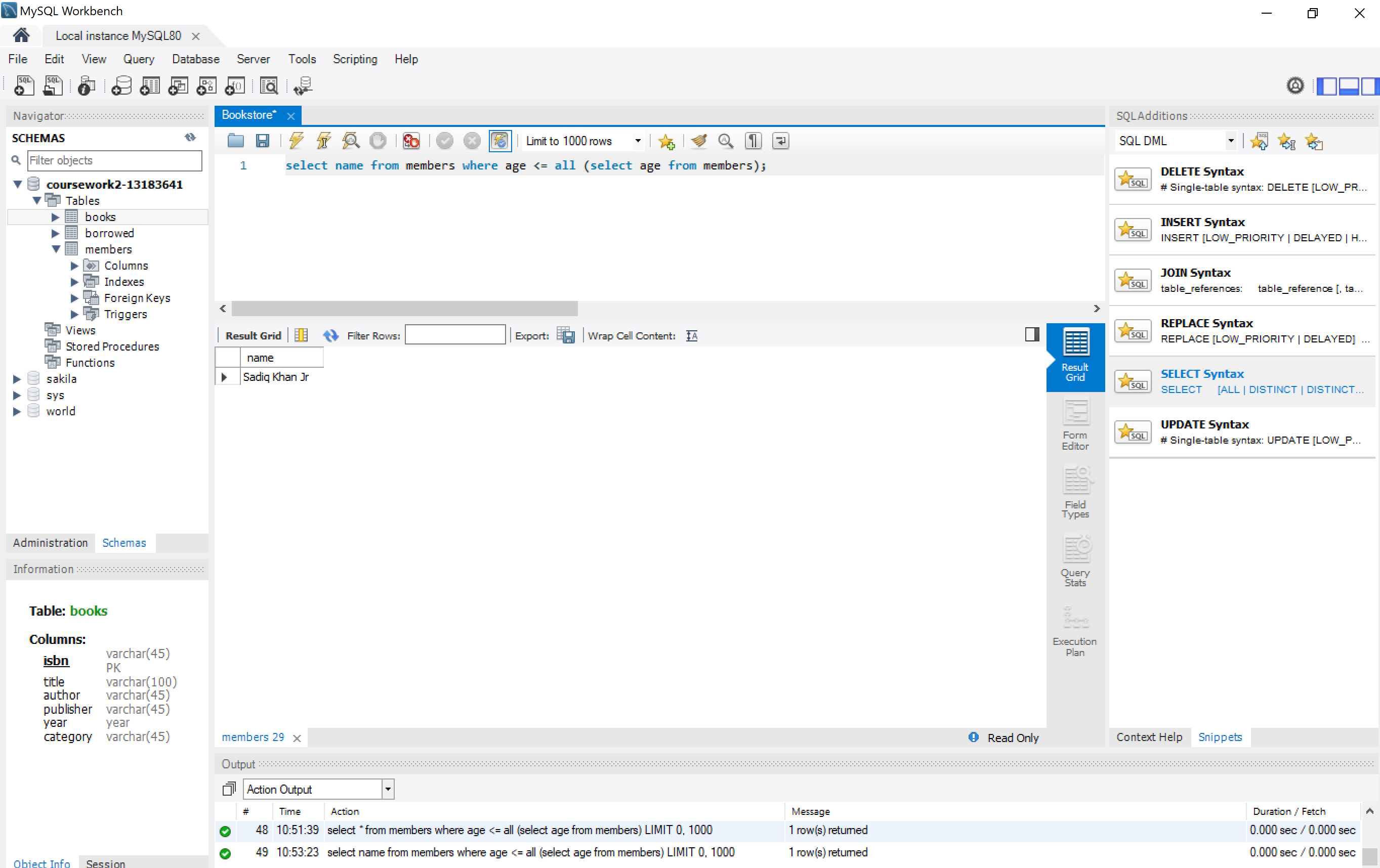




1. **Find the name of the youngest member of the library.**

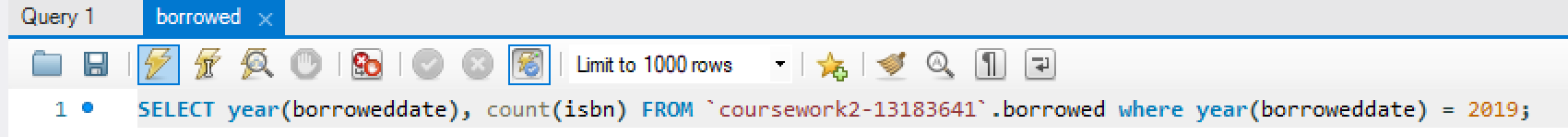
In this case I select column name from members table and add the condition that its age is the minor value of all ages from members using all and comparing to another query that includes ages from all members. As it is a very simple table there is only one member with that age.

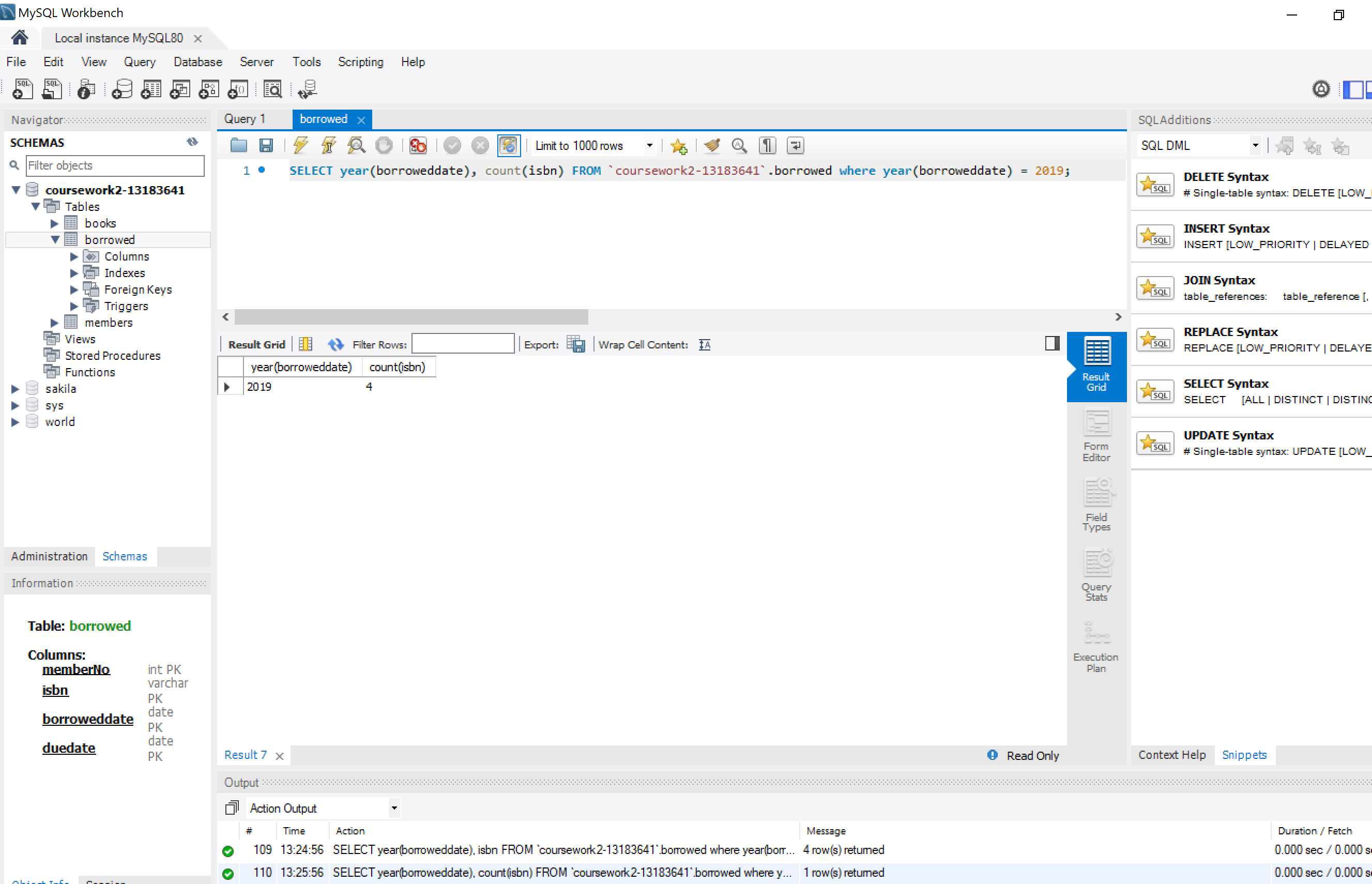




1. **Find the total number of books borrowed in 2019.**

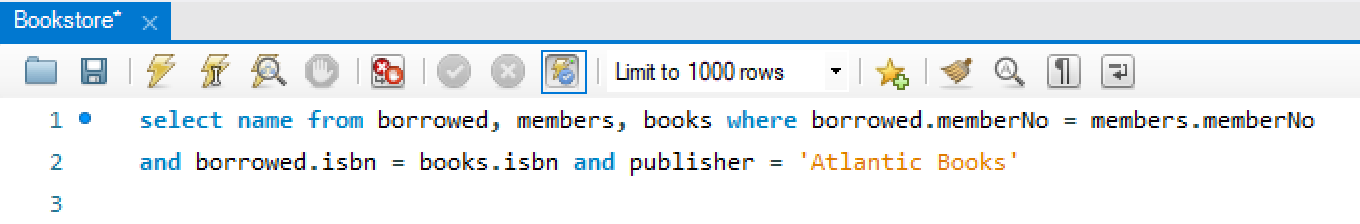
In this case I use the year() command to group all books based on borroweddate and count them by ISBN in the borrowed table. Finally I add the condition where the borrowed year equals 2019.

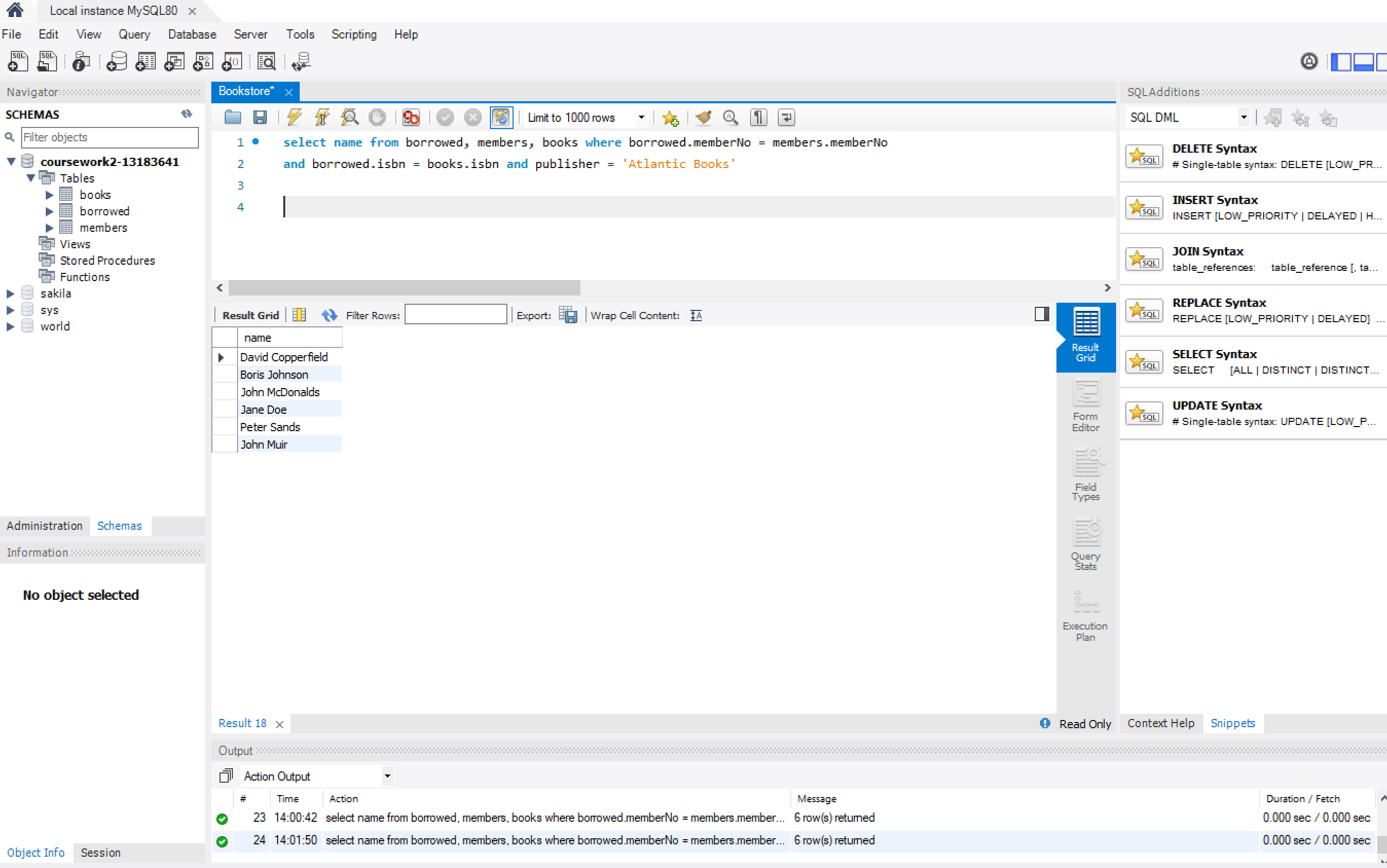




1. **Find the names of members who have borrowed any book published by "Atlantic Books".**

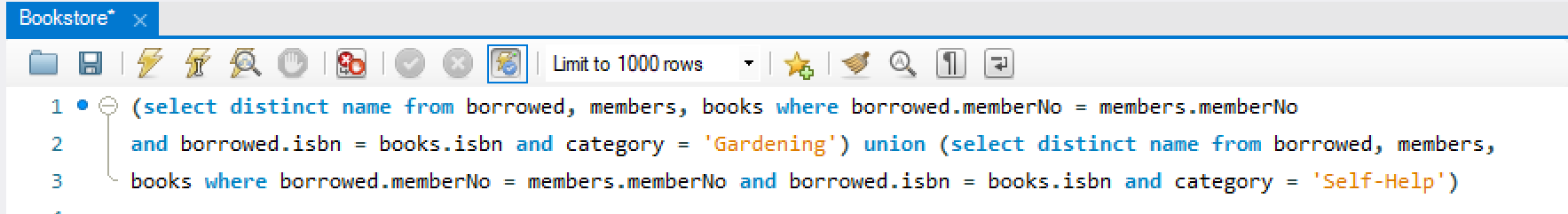
Here I am linking the 3 main tables using its main keys (memberNo and Isbn). Out of that data that does not have duplicates I chose the name and add the condition using AND that the publisher equals the string ‘Atlantic Books’.

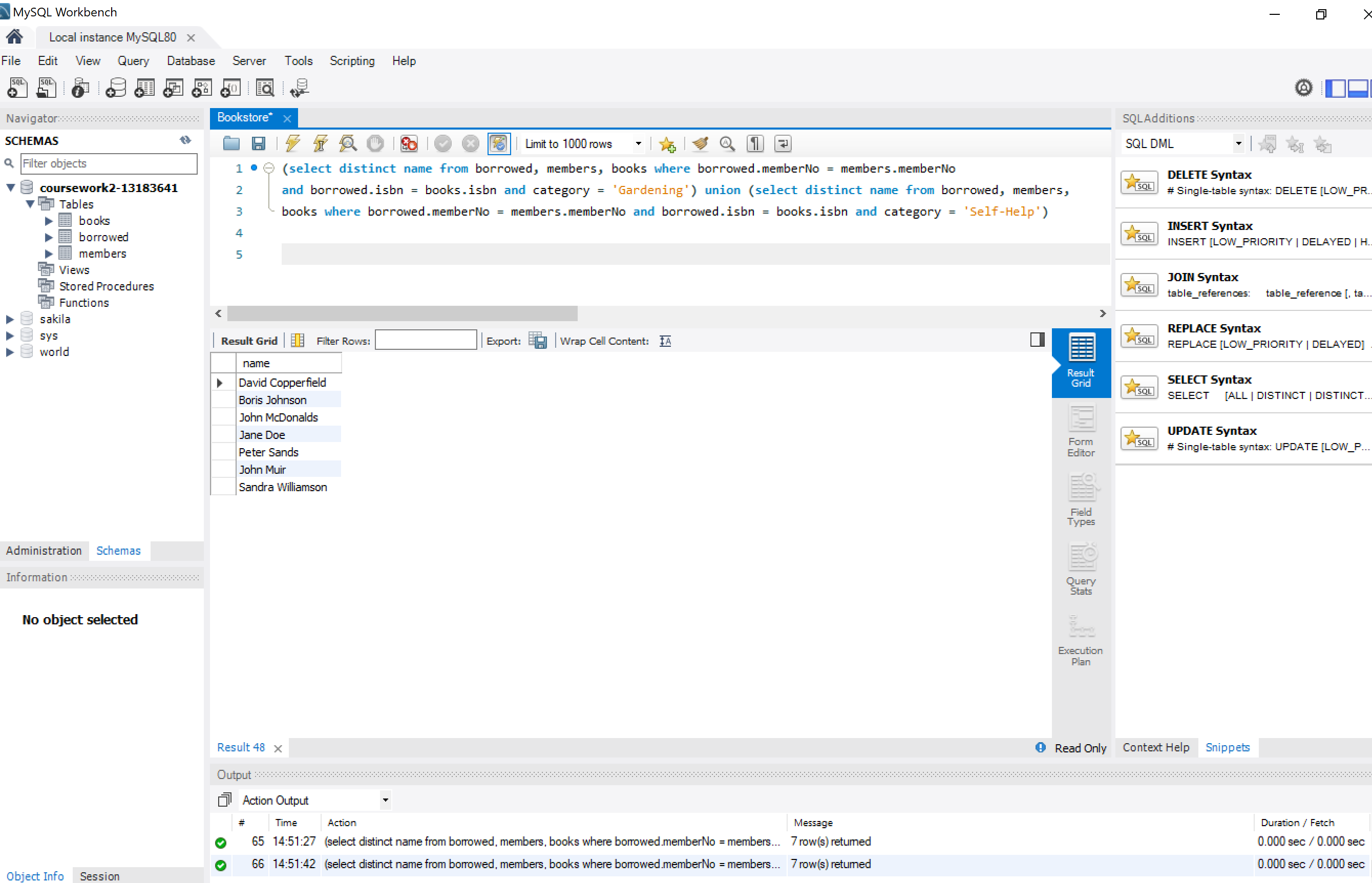




1. For each category, find the names of members who have borrowed more than five books in that category.
2. **Find the names of members who have borrowed books from both of the categories "Gardening" and "Self Help".**

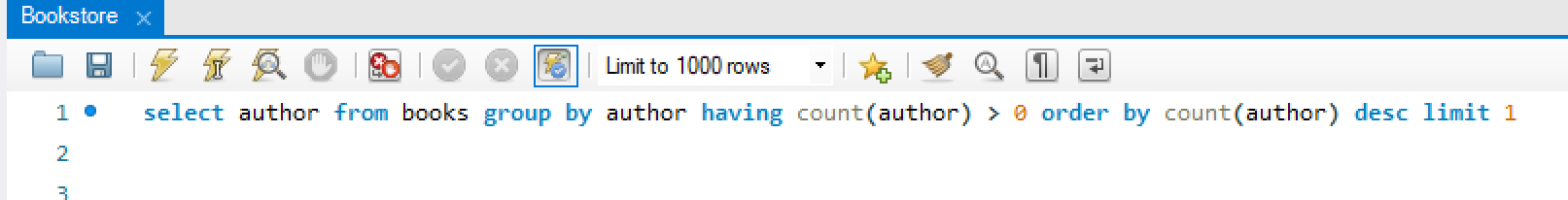
I use the Union to merge 2 different queries. Firstly, out of the 3 main tables linked I chose the category ‘Gardening’. Secondly, Using UNION I add another similar query where the books have been selected by category ‘Self-Help’. Union adds both querys.

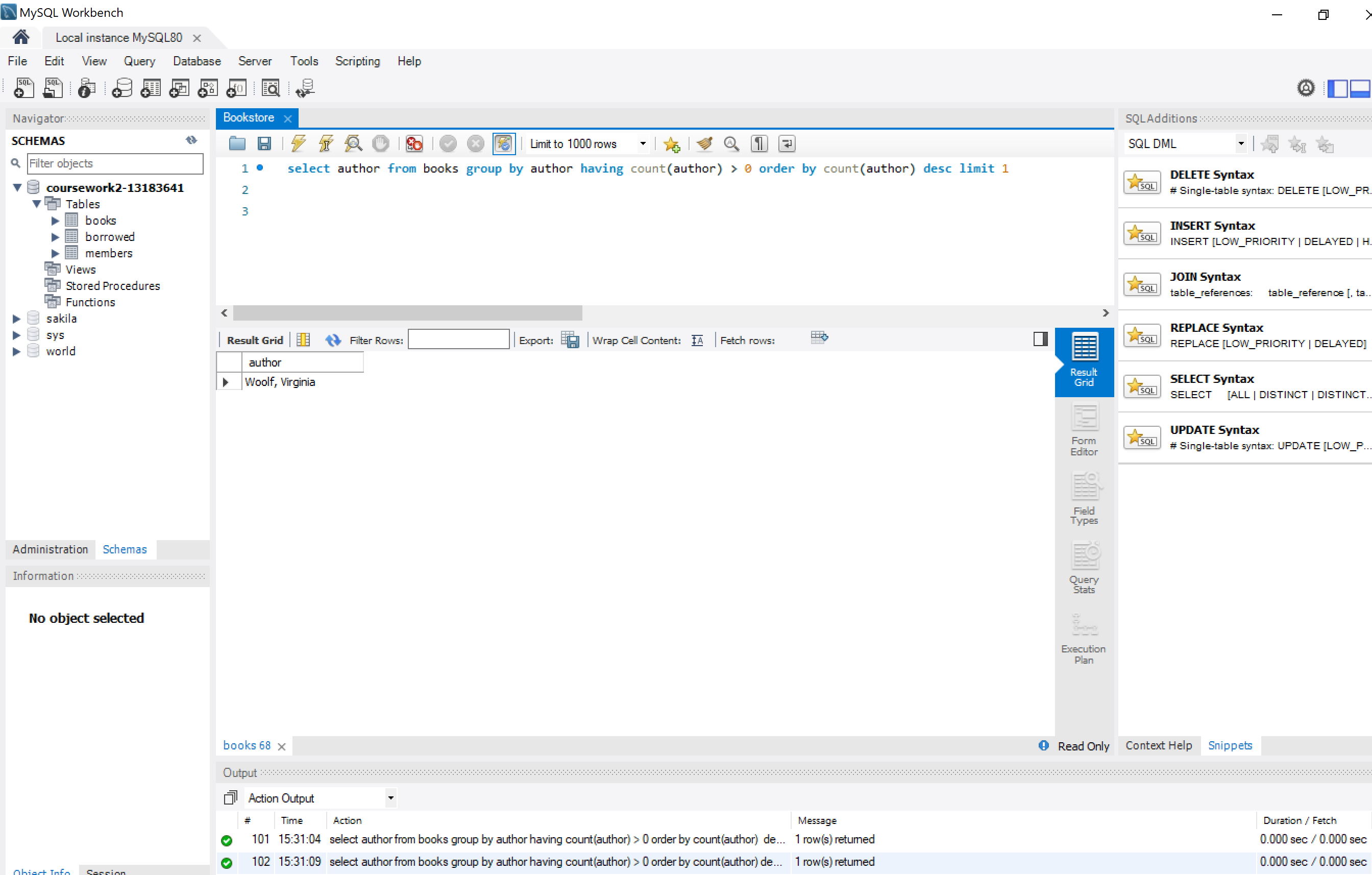




1. **Find the most prolific author in the library, i.e., the author who has authored the most books.**

Simple query extracting data from one single table. I group the books by author and add them using HAVING COUNT > 0. I sort the books in descending order and I use LIMIT 1 to pick the highest value.





1. Find the names of members who have borrowed the same book more than once.

Submit a pdf or text file containing a listing of the SQL queries and their results **(See attached files**). Write a sentence or two on each query describing what you did.