

Database Management - 2020

Part II: SQL Queries (50 marks)

Books

The screenshot shows the MySQL Workbench interface. The 'Navigator' pane on the left displays the 'coursework2-13183641' database with tables 'books', 'borrowed', 'members', 'sakila', 'sys', and 'world'. The 'SQL Editor' pane contains the query: `(select * from books)`. The 'Result Grid' shows the following data:

isbn	title	author	publisher	year	category
1913038532	The happy hypocrite : a fairy tale for tired men	Beerbohm Max	John Lane, the...	1905	Novel
0500232121	The heyday of salon painting : masterpieces of bourgeois realism	Čelebonović, Aleksa	Thames and Hu...	1974	Art
1854106554	The garden : a history in landscape and art	Pizzoni, Filippo	Aurum Press	1999	Gardening
0192839705	Mrs Dalloway	Woolf, Virginia	Oxford World ...	2000	Novel
041526643	The psychology of being happy	Argyle, Michael	Routledge	2001	Self-Help
03004056	The discovery of mankind : Atlantic encounters in the age of Columbus /	Abulafia, David	Atlantic Books	2008	Gardening
9780191751035	The Oxford handbook of happiness	David, Susan	Oxford Univers...	2013	Self-Help
0953886670	Translations from the Russian	Woolf, Virginia	Virginia Woolf S...	2017	Novel

The 'Administration' pane on the left shows the 'Table: books' with columns: isbn (varchar(45), PK), title (varchar(100)), author (varchar(45)), publisher (varchar(45)), year (year), and category (varchar(45)).

Members

The screenshot shows the MySQL Workbench interface. The 'Navigator' pane on the left displays the 'coursework2-13183641' database with tables 'books', 'borrowed', 'members', 'sakila', 'sys', and 'world'. The 'SQL Editor' pane contains the query: `(select * from members)`. The 'Result Grid' shows the following data:

memberNo	name	age
1	David Copperfield	32
2	Boris Johnson	55
3	John McDonalds	24
4	Jane Doe	83
5	Peter Sands	41
6	John Muir	92
7	Sandra Williamson	34
8	Sadiq Khan Jr	14

The 'Administration' pane on the left shows the 'Table: books' with columns: isbn (varchar(45), PK), title (varchar(100)), author (varchar(45)), publisher (varchar(45)), year (year), and category (varchar(45)).

The 'Output' pane at the bottom shows the execution results for the query `(select * from books)`:

#	Time	Action	Message	Duration / Fetch
34	10:26:21	(select * from books)	8 row(s) returned	0.000 sec / 0.000 sec

Borrowed

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'coursework2-13183641' database schema with tables 'books', 'borrowed', 'members', 'sakila', 'sys', and 'world'. The 'books' table is selected, showing its columns: isbn (PK), title, author, publisher, year, and category. The main window shows a query editor with the SQL statement: `select * from borrowed`. The 'Result Grid' displays 22 rows of data with columns: memberNo, isbn, borrowed-date, and due-date. The 'Output' pane at the bottom shows the execution of the query, indicating 25 rows returned.

memberNo	isbn	borrowed-date	due-date
1	03004056	2002-03-20 00:00:00	2005-03-20 00:00:00
1	0415226643	2011-05-05 00:00:00	2011-05-15 00:00:00
1	9780191751035	2000-01-10 00:00:00	2000-01-20 00:00:00
1	9780191751035	2019-01-12 00:00:00	2019-01-22 00:00:00
2	03004056	2001-09-09 00:00:00	2001-09-19 00:00:00
2	0415226643	2004-04-10 00:00:00	2004-04-20 00:00:00
2	0500232121	2020-09-09 00:00:00	2020-09-19 00:00:00
2	0953886670	2020-02-02 00:00:00	2020-02-12 00:00:00
2	9780191751035	2005-08-08 00:00:00	2005-08-18 00:00:00
3	03004056	2012-12-12 00:00:00	2012-12-22 00:00:00
3	0500232121	2003-05-05 00:00:00	2003-05-14 00:00:00
3	1912038532	2016-10-20 00:00:00	2016-10-30 00:00:00
4	03004056	2008-01-20 00:00:00	2008-01-30 00:00:00
4	0500232121	2018-03-03 00:00:00	2018-03-13 00:00:00
4	9780191751035	2015-03-03 00:00:00	2015-03-13 00:00:00
5	03004056	2019-01-01 00:00:00	2019-01-11 00:00:00
5	0500232121	2007-05-05 00:00:00	2007-05-05 00:00:00
6	03004056	2007-05-05 00:00:00	2007-05-15 00:00:00
6	0415226643	2005-03-18 00:00:00	2005-03-28 00:00:00
6	0415226643	2019-05-05 00:00:00	2019-05-15 00:00:00
6	0953886670	1995-09-09 00:00:00	1995-09-19 00:00:00

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.

The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' pane shows the 'coursework2-13183641' database with tables 'books', 'borrowed', 'members', 'sakila', 'sys', and 'world'. The 'members' table is selected, showing its columns: 'memberid' (int UN PK), 'name' (varchar(45)), and 'age' (varchar(3)).

The main query editor shows the following SQL query:

```
1 select title, category, year from books order by category ASC, year DESC;
```

The 'Result Grid' shows the following data:

title	category	year
The heyday of salon painting : masterpieces of bourgeois realism	Art	1974
The discovery of mankind : Atlantic encounters in the age of Columbus /	Gardening	2008
The garden : a history in landscape and art	Gardening	1999
Translations from the Russian	Novel	2017
Mrs Dalloway	Novel	2000
The happy hypocrite : a fairy tale for tired men	Novel	1905
The Oxford handbook of happiness	Self-Help	2013
The psychology of happiness	Self-Help	2001

The 'Output' pane at the bottom shows the execution of the query, with a message indicating an error: 'Error Code: 1064: You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '' at line 1'.

2. 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.

The screenshot shows the MySQL Workbench interface with the following SQL query in the query editor:

```
1 select * from books where category = 'Self-Help' and title like '%happy';
```

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

Filter objects

SCHEMAS

coursework2-13183641

Tables

books

borrowed

members

Columns

Indexes

ForeignKeys

Triggers

Views

Stored Procedures

Functions

sakila

sys

world

Administration Schemas

Information

Table: members

Columns:

memberNo int UN PK

name varchar(45)

age varchar(3)

SQL Editor

1 • select * from books where category = 'Self-Help' and title like '%Happy';

Result Grid

isbn	title	author	publisher	year	category
0415226643	The psychology of being happy	Argyle, Michael	Routledge	2001	Self-Help

books 19 x

Output

#	Time	Action	Message	Duration / Fetch
31	10:12:14	select * from books where category = 'Self-Help' and title like '%Happy' LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
32	10:12:31	select * from books where category = 'Self-Help' and title like '%Happy' LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

SQLAdditions

SQL DML

DELETE Syntax

INSERT Syntax

JOIN Syntax

REPLACE Syntax

SELECT Syntax

UPDATE Syntax

3. 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'.

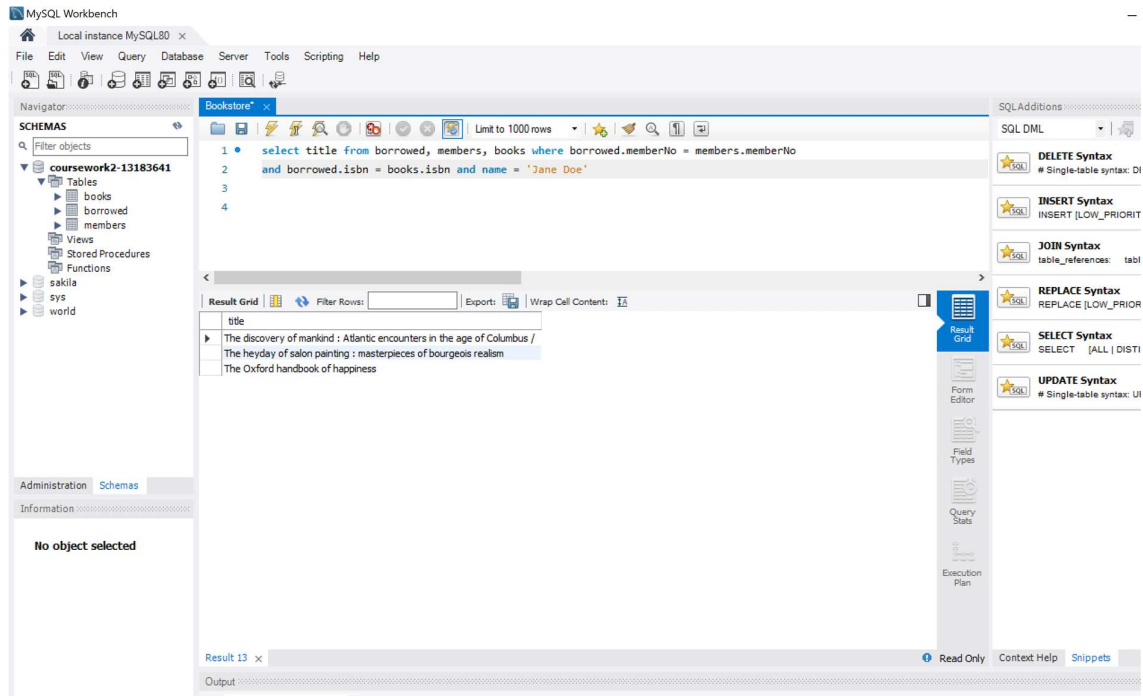
Bookstore x

Limit to 1000 rows

1 • select title from borrowed, members, books where borrowed.memberNo = members.memberNo

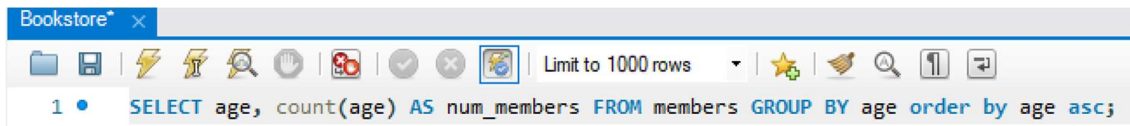
2 and borrowed.isbn = books.isbn and name = 'Jane Doe'

3



4. 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.



The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'Schemas' tree with 'coursework2-13183641' selected. The main window shows a query: `SELECT age, count(age) AS num_members FROM members GROUP BY age order by age asc;`. The 'Result Grid' shows the following data:

age	num_members
14	1
24	1
32	1
34	1
41	1
55	1
83	1
92	1

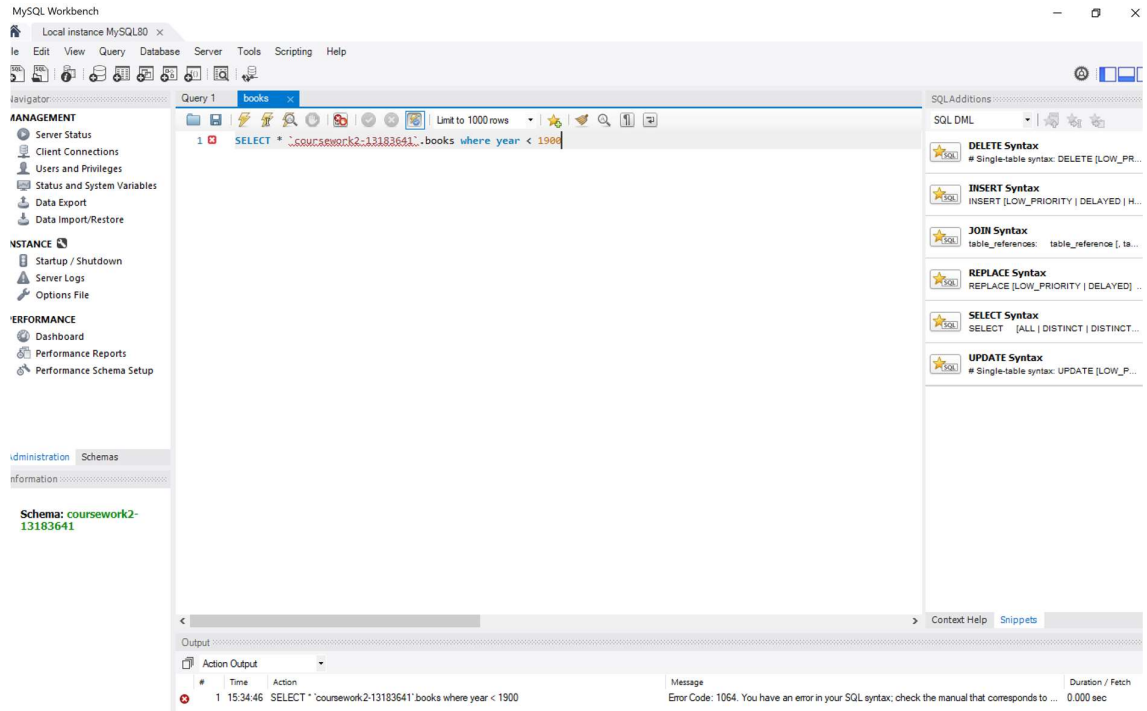
The right sidebar shows 'SQL DML' syntax examples for DELETE, INSERT, JOIN, REPLACE, SELECT, and UPDATE. The bottom status bar shows the query execution time and row count.

5. 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.

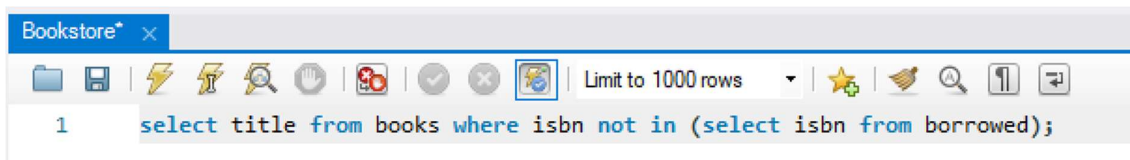
The query itself is not difficult at all: just select books in the DB and include the condition where year is less than 1900.

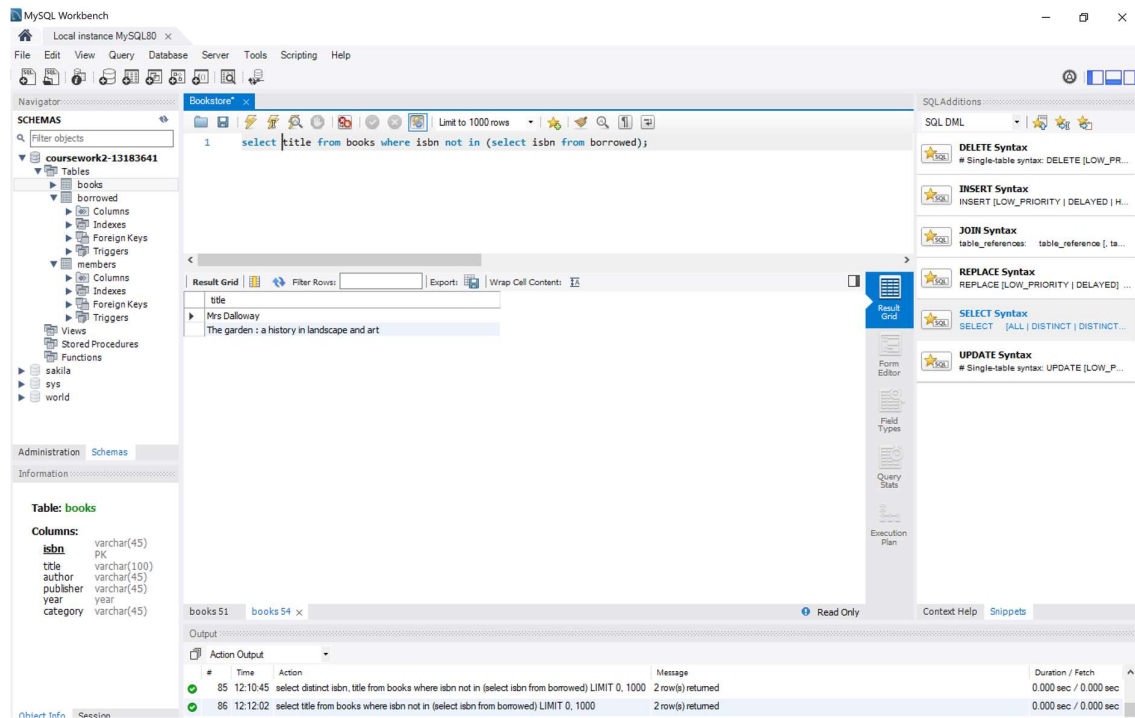
The screenshot shows the MySQL Workbench interface with a query editor. The query is: `SELECT * FROM coursework2-13183641.books where year < 1900`. The query is highlighted in blue.



6. Find the titles of books that have never been borrowed.

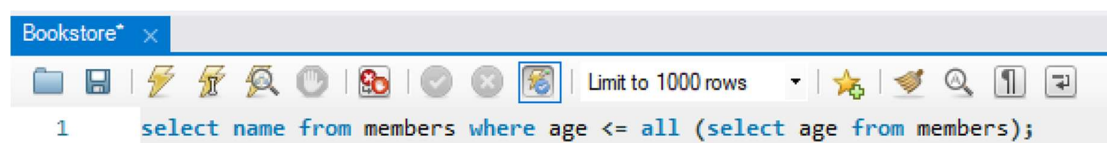
Books are identified by ISBN. I select the title from books table and add the condition that its isbn does not match any included in the borrowed tables using NOT IN.

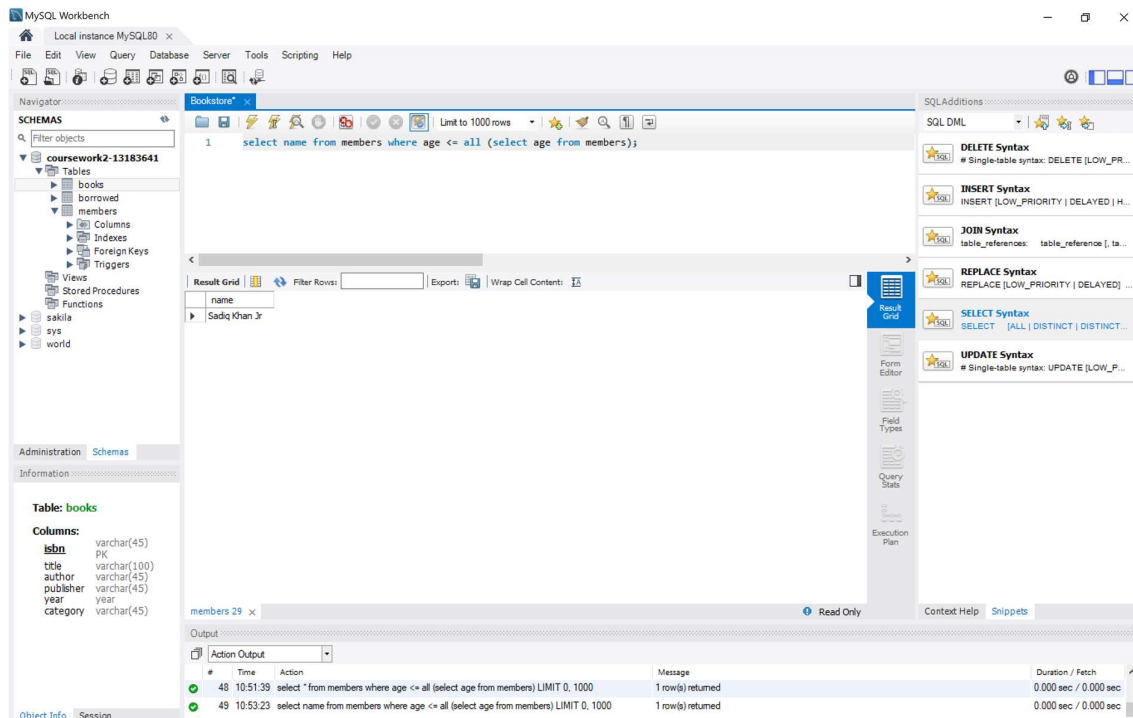




7. Find the name of the youngest member of the library.

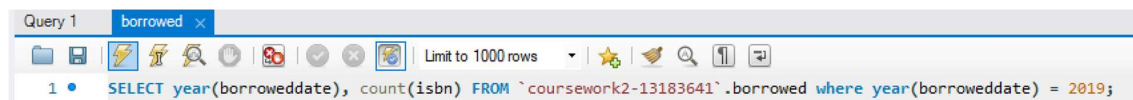
In this case I select column name from members table and add the condition that its age is the minor value of all ages from members using all and comparing to another query that includes ages from all members. As it is a very simple table there is only one member with that age.





8. 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.



The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' pane shows a tree view for 'coursework2-13183641' with tables 'books', 'borrowed', 'members', 'sakila', 'sys', and 'world'. The 'borrowed' table is selected. The 'Table: borrowed' details pane shows columns: 'memberNo' (int PK), 'isbn' (varchar PK), 'borroweddate' (date PK), and 'duedate' (date PK). The 'Query' pane contains the following SQL query:

```
1 • SELECT year(borroweddate), count(isbn) FROM `coursework2-13183641`.borrowed where year(borroweddate) = 2019;
```

The 'Result Grid' shows the following data:

year(borroweddate)	count(isbn)
2019	4

The 'Output' pane shows the execution log:

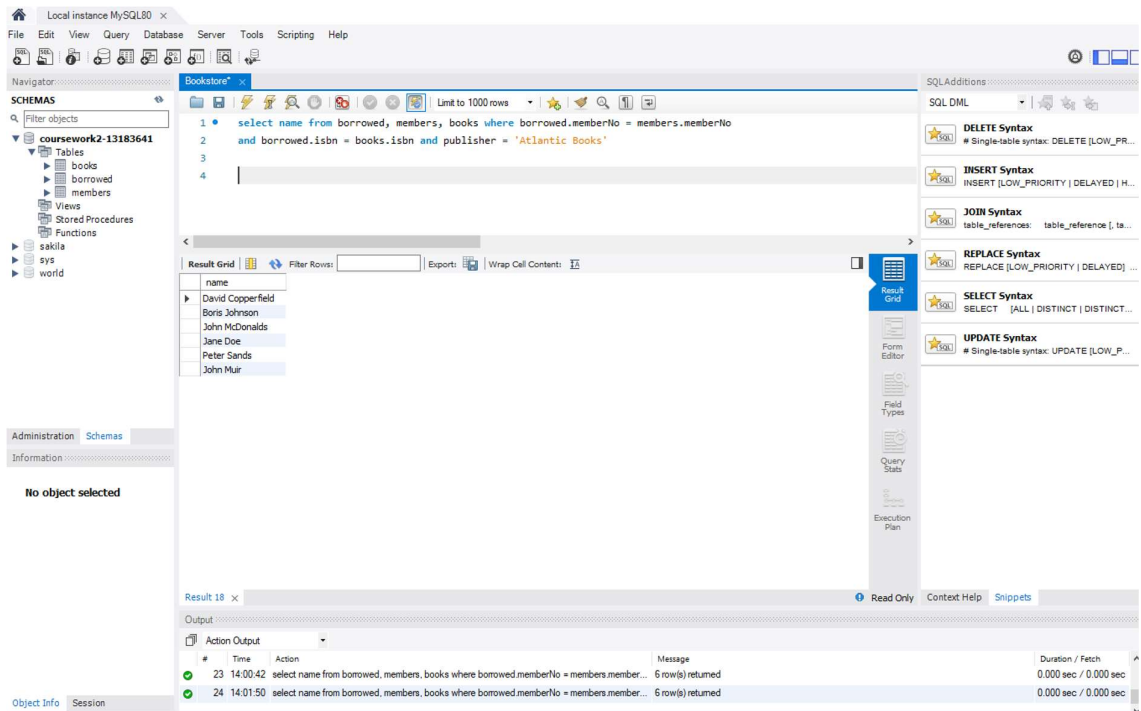
#	Time	Action	Message	Duration / Fetch
109	13:24:56	SELECT year(borroweddate), isbn FROM `coursework2-13183641`.borrowed where year(bor...	4 row(s) returned	0.000 sec / 0.000 s
110	13:25:56	SELECT year(borroweddate), count(isbn) FROM `coursework2-13183641`.borrowed where y...	1 row(s) returned	0.000 sec / 0.000 s

9. 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'.

The screenshot shows the MySQL Workbench interface with a query in the 'Query' pane:

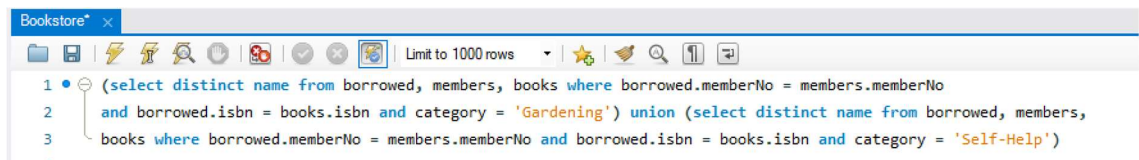
```
1 • select name from borrowed, members, books where borrowed.memberNo = members.memberNo
2 and borrowed.isbn = books.isbn and publisher = 'Atlantic Books'
3
```

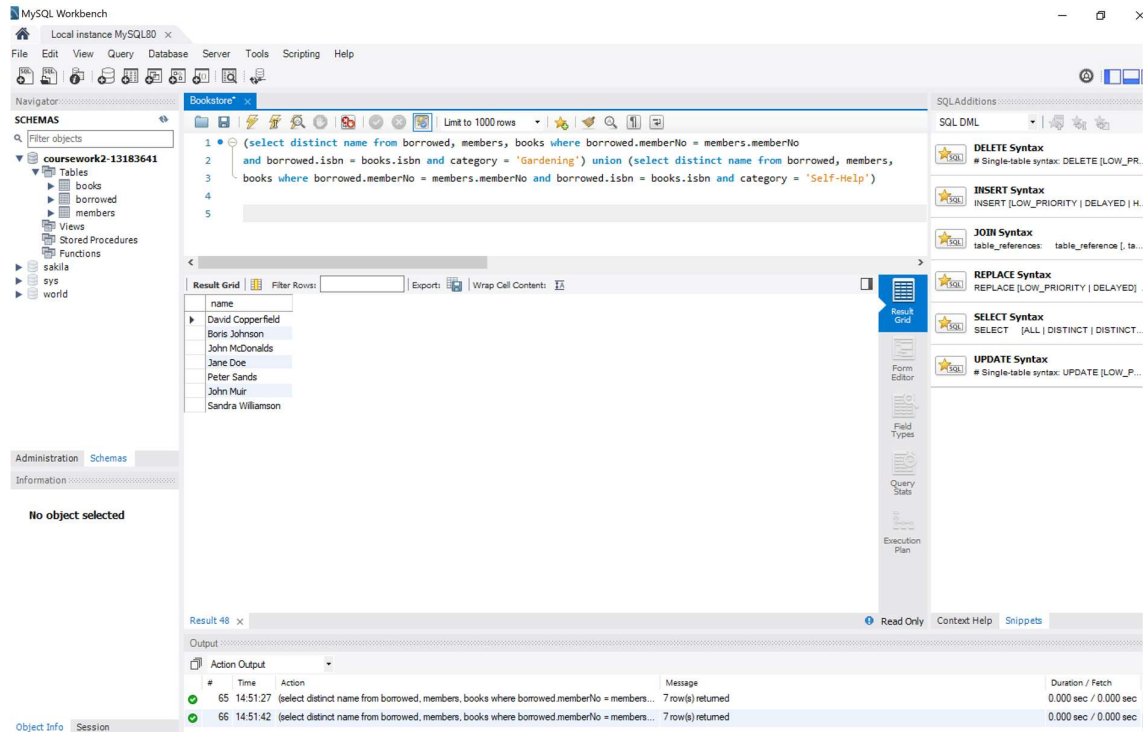


10. For each category, find the names of members who have borrowed more than five books in that category.

11. Find the names of members who have borrowed books from both of the categories "Gardening" and "Self Help".

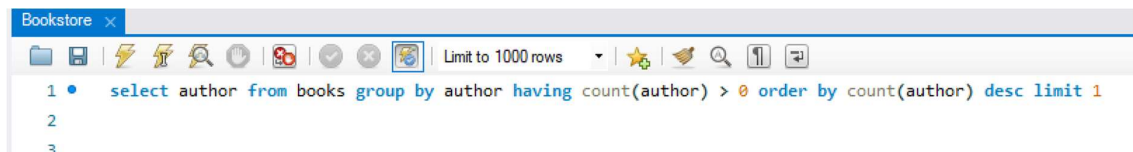
I use the Union to merge 2 different queries. Firstly, out of the 3 main tables linked I chose the category 'Gardening'. Secondly, Using UNION I add another similar query where the books have been selected by category 'Self-Help'. Union adds both queries.

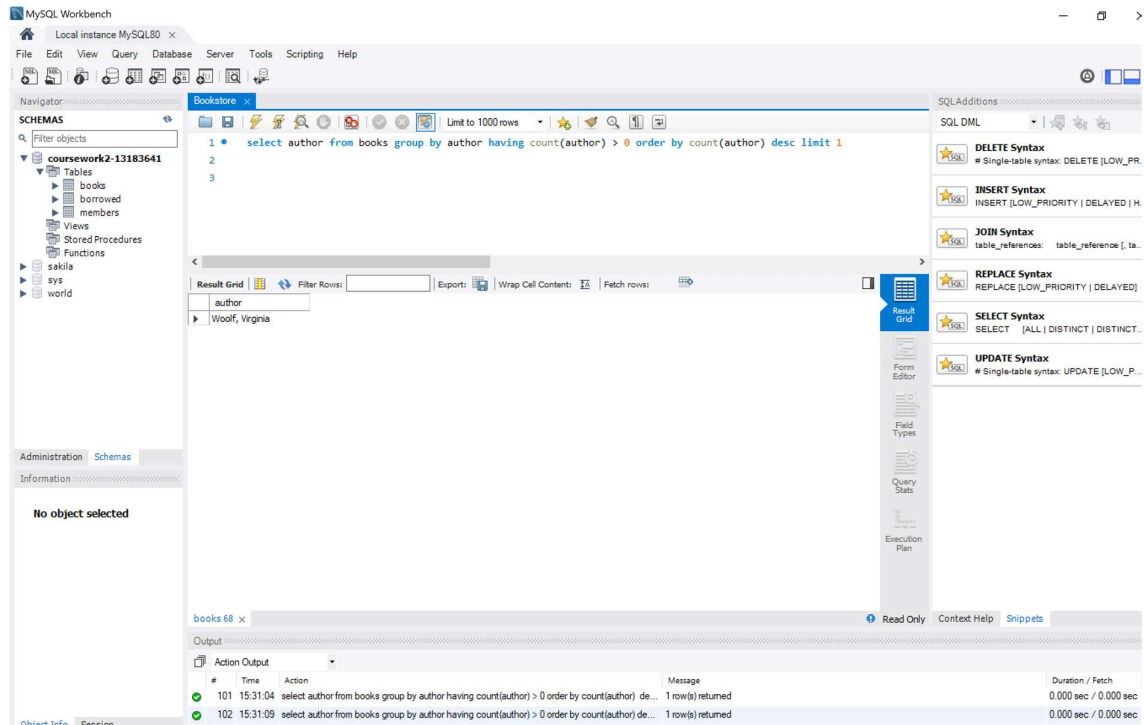




12. Find the most prolific author in the library, i.e., the author who has authored the most books.

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.





13. 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.