

## Prac #8

### SQL Practice II:

### Writing/Executing advanced SQL queries in MySQL Workbench to extract information from a database

Same as you did in the last prac (Prac #7), you will continue to practice SQL queries to extract useful information using MySQL Workbench. You will use the library database you created in previous pracs, and compose a number of SQL queries to extract information from the database.

While you practiced basic SQL query techniques to extract existing data from the database (from one single table) in the previous prac (Prac #7), in this prac you are going to practice advanced SQL query techniques to extract manipulated information using aggregating functions and to use JOIN operator syntax.

- **Learning outcomes and objectives**

Student will be able to

- compose various SQL queries in particular using SELECT command by adding restrictions to the search criteria.
- use special SQL commands to restrict or adjust the way of displaying output of the query
- use aggregate functions in SQL queries to carry out useful functions like counting, finding maximum/minimum etc.
- compose various SQL queries in particular using JOIN operators to extract information from two or more tables

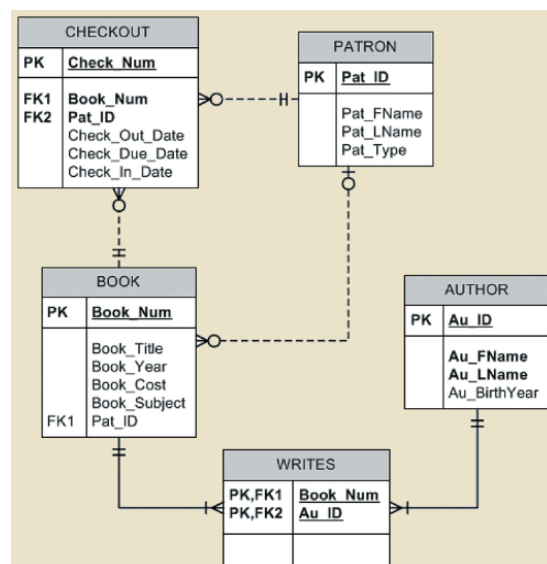
- **Task Overview**

Open the library database you created in the previous practical on MySQL Workbench. The conceptual model of the database is presented in the ERD as shown here.

Use this ERD as a quick reference when you compose SQL queries for this task questions.

Solutions are provided for you for some exercises. For each of these exercises, you will need to write/save an SQL query (though some exercises already show solutions, you are always recommended to write and run the code yourself).

For each of these exercises, a figure of the correct output is provided. If the output of the query is very large, only the first several rows of the output are shown. The head section of each query result must look exactly same as the output provided.



For your own back-up purpose, save a query for each question as the name of "Q1.sql", "Q2.sql", ...  
To submit this prac, you are required to copy-and-paste all queries to one WORD or Text document  
and submit it to be marked off. Please notate clearly each query number (e.g. Q1, Q2, ...).

[The following ten queries (Q1 ~ Q10) are required for you to use basic aggregating functions using a single table records. Please refer to contents covered in the lecture ([Lecture 07 SQL Part 2](#))]

1. Write a query to display the number of books stored in the library database system (See the figure below for the output).

Number of Books
20

Answer provided:

```
SELECT Count(BOOK_NUM) AS "Number of Books"  
FROM BOOK;
```

2. Write a query to display the number of different book subjects in this database system (See the figure below for the output).

Number of Subjects
4

Answer provided:

```
SELECT Count(DISTINCT BOOK_SUBJECT) AS "Number of Subjects"  
FROM BOOK;
```

3. Write a query to display the number of books that are available (not currently checked out) (See the figure below for the output).

Available Books
14

4. Write a query to display the highest book cost in the system (See the figure below for the output).

Most Expensive
129.95

Answer provided:

```
SELECT Max(BOOK_COST) AS "Most Expensive"  
FROM BOOK;
```

5. Write a query to display the lowest book cost in the system (See the figure below for the output).

Least Expensive
49.95

6. Write a query to display the total value of all books in the library (See the figure below for the output).

Library Value
1499

7. Write a query to display the number of different patrons who have ever checked out a book (See the figure below for the output).

DIFFERENT PATRONS
33

8. Write a query to display the subject and the number of books in each subject. Sort the results by the number of books in descending order, then by subject name in ascending order (See the figure below for the output).

BOOK_SUBJECT	Books In Subject
Programming	9
Cloud	4
Database	4
Middleware	3

**Answer provided:**

```
SELECT BOOK_SUBJECT, Count(*) AS "Books In Subject"
FROM BOOK
GROUP BY BOOK_SUBJECT;
```

9. Write a query to display the author ID and the number of books written by that author. Sort the results in descending order by number of books, then in ascending order by author ID (See the figure for the output).

AU_ID	Books Written
262	3
460	3
185	2
229	2
251	2
383	2
394	2
559	2
218	1
273	1
284	1
438	1
581	1
592	1
603	1

**Answer provided:**

```
SELECT AU_ID, Count(*) AS "Books
Written"
FROM WRITES
GROUP BY AU_ID
ORDER BY COUNT(*) DESC, AU_ID;
```

10. Write a query to display the book number and the number of times each book has been checked out. Do not include books that have never been checked out. Sort the results by the number of times checked out in descending order and then by book number in descending order (See the figure below for the output)

BOOK_NUM	Times Checked Out
5236	12
5235	9
5240	7
5238	6
5237	5
5254	4
5252	4
5249	4
5246	4
5244	4
5242	4
5248	3
5243	2

[The following queries (Q11 ~ Q23) are required for you to use aggregating functions and/or JOIN operations to extract information from multiple tables . Please refer to contents covered in the lecture ([Lecture 07 SQL Part 2](#))]

11. Write a query to display the author last name, author first name, and book number for each book written by that author. Sort the results by author last name, first name, and then book number (See the figure for the output. The actual result will have 25 rows)

**Answer provided:**

```
SELECT AU_LNAME, AU_FNAME, BOOK_NUM
FROM AUTHOR JOIN WRITES ON
AUTHOR.AU_ID = WRITES.AU_ID
ORDER BY AU_LNAME, AU_FNAME,
BOOK_NUM;
```

AU_LNAME	AU_FNAME	BOOK_NUM
Aggerwal	Manish	5242
Beatney	Rachel	5240
Bruer	Hugo	5243
Bruer	Hugo	5246
Chiang	Xia	5244
Chiang	Xia	5249
Chiang	Xia	5252
Durante	Reba	5235
Lake	Robert	5245
Lake	Robert	5247
McGill	Rachel	5241
McGill	Rachel	5254
Palca	Julia	5238

**Note:** There are various ways to implement join operation in SQL, thus there may be various SQL solutions for this task. The sample solution provided is just one of them.

12. Write a query to display the author ID, book number, title, and subject for each book. Sort the results by book number and then author ID (See the figure below for the output. The actual result will have 25 rows)

AU_ID	BOOK_NUM	BOOK_TITLE	BOOK_SUBJECT
273	5235	Beginner's Guide to JAVA	Programming
383	5236	Database in the Cloud	Cloud
185	5237	Mastering the database environment	Database
603	5238	Conceptual Programming	Programming
229	5239	J++ in Mobile Apps	Programming
460	5239	J++ in Mobile Apps	Programming
592	5239	J++ in Mobile Apps	Programming
218	5240	iOS Programming	Programming
460	5241	JAVA First Steps	Programming
559	5241	JAVA First Steps	Programming

13. Write a query to display the author last name, first name, book title, and replacement cost for each book. Sort the results by book number and then author ID (See the figure below for the output. The actual result will have 25 rows)

AU_LNAME	AU_FNAME	BOOK_TITLE	BOOK_COST
Durante	Reba	Beginner's Guide to JAVA	59.95
Walsh	Neal	Database in the Cloud	79.95
Reeves	Benson	Mastering the database environment	89.95
Palca	Julia	Conceptual Programming	59.95
Salvadore	Carmine	J++ in Mobile Apps	49.95
Paulsen	Connie	J++ in Mobile Apps	49.95
Sheel	Lawrence	J++ in Mobile Apps	49.95
Beatney	Rachel	iOS Programming	79.95
Paulsen	Connie	JAVA First Steps	49.95
McGill	Rachel	JAVA First Steps	49.95
Aggerwal	Manish	C# in Middleware Deployment	59.95

Answer provided:

```
SELECT AU_LNAME, AU_FNAME, BOOK_TITLE, BOOK_COST
FROM AUTHOR JOIN WRITES ON AUTHOR.AU_ID = WRITES.AU_ID JOIN
BOOK ON WRITES.BOOK_NUM = BOOK.BOOK_NUM
ORDER BY BOOK.BOOK_NUM, AUTHOR.AU_ID;
```

14. Write a query to display the patron ID, book number, patron first name and last name, and book title for all currently checked out books. (Remember to use the redundant relationship made between Patron and Book for current checkouts.) Sort the output by patron last name and book title. (See the figure below for the output)

PAT_ID	BOOK_NUM	PAT_FNAME	PAT_LNAME	BOOK_TITLE
1229	5252	Gerald	Burke	Beyond the Database Veil
1229	5238	Gerald	Burke	Conceptual Programming
1228	5242	Homer	Goodman	C# in Middleware Deployment
1212	5240	Iva	McClain	iOS Programming
1172	5246	Tony	Miles	Capture the Cloud
1207	5249	Iva	Ramos	Starlight Applications

15. Write a query to display the author ID, first and last name, book number, and book title of all books in the subject "Cloud". Sort the results by book title and then by author last name. (See the figure below for the output)

AU_ID	AU_FNAME	AU_LNAME	BOOK_NUM	BOOK_TITLE
251	Hugo	Bruer	5246	Capture the Cloud
262	Xia	Chiang	5244	Cloud-based Mobile Applications
284	Trina	Tankersly	5244	Cloud-based Mobile Applications
383	Neal	Walsh	5236	Database in the Cloud
262	Xia	Chiang	5249	Starlight Applications

16. Write a query to display the book number, title, author last name, author first name, patron ID, last name, and patron type for all books currently checked out to a patron. Sort the results by book title. (See the figure below for the output)

BOOK_NUM	BOOK_TITLE	AU_LNAME	AU_FNAME	PAT_ID	PAT_LNAME	PAT_TYPE
5252	Beyond the Database Veil	Chiang	Xia	1229	Burke	Student
5242	C# in Middleware Deployment	Aggerwal	Manish	1228	Goodman	Student
5246	Capture the Cloud	Bruer	Hugo	1172	Miles	STUDENT
5238	Conceptual Programming	Palca	Julia	1229	Burke	Student
5240	iOS Programming	Beatney	Rachel	1212	McClain	Student
5249	Starlight Applications	Chiang	Xia	1207	Ramos	Student

17. Write a query to display the book number, title, and number of times each book has been checked out. Include books that have never been checked out. Sort the results in descending order by the number of times checked out and then by title. (See the figure below for the output)

BOOK_NUM	BOOK_TITLE	Times Checked Out
5236	Database in the Cloud	12
5235	Beginner's Guide to JAVA	9
5240	iOS Programming	7
5238	Conceptual Programming	6
5237	Mastering the database environment	5
5252	Beyond the Database Veil	4
5242	C# in Middleware Deployment	4
5246	Capture the Cloud	4
5244	Cloud-based Mobile Applications	4
5254	Coding Style for Maintenance	4
5249	Starlight Applications	4
5248	What You Always Wanted to Know About Database, But Were Afraid to Ask	3
5243	DATABASES in Theory	2
5239	J++ in Mobile Apps	0
5241	JAVA First Steps	0
5250	Reengineering the Middle Tier	0
5247	Shining Through the Cloud: Sun Programming	0
5245	The Golden Road to Platform independence	0
5251	Thoughts on Revitalizing Ruby	0
5253	Virtual Programming for Virtual Environments	0

Answer provided:

```
SELECT BOOK.BOOK_NUM, BOOK_TITLE, Count(CHECK_NUM) AS "Times
Checked Out"
FROM BOOK LEFT JOIN CHECKOUT ON BOOK.BOOK_NUM =
CHECKOUT.BOOK_NUM
GROUP BY BOOK.BOOK_NUM, BOOK_TITLE
ORDER BY COUNT(CHECK_NUM) DESC, BOOK_TITLE;
```

18. Write a query to display the author ID, author last name, book title, checkout date, and patron last name for all the books written by authors with the last name "Bruer" that have ever been checked out by patrons with the last name "Miles". Sort the results by check out date (See the figure below for the output)

AU_ID	AU_LNAME	BOOK_TITLE	CHECK_OUT_DATE	PAT_LNAME
251	Bruer	Capture the Cloud	2017-04-21	Miles
251	Bruer	Capture the Cloud	2017-05-15	Miles

19. Write a query to display the patron ID, first and last name of all patrons that have never checked out any book. Sort the result by patron last name and then first name. (See the figure below for the output)

PAT_ID	PAT_FNAME	PAT_LNAME
1166	Vera	Alvarado
1180	Nadine	Blair
1238	Erika	Bowen
1208	Ollie	Cantrell
1227	Alicia	Dickson
1205	Claire	Gomez
1239	Elton	Irwin
1240	Jan	Joyce
1243	Roberto	Kennedy
1242	Mario	King
1237	Brandi	Larson
1167	Alan	Martin
1182	Jamal	Melendez
1201	Shelby	Noble
1244	Leon	Richmond
1200	Lorenzo	Torres
1241	Irene	West

20. Write a query to display the book number, title, and number of times each book has been checked out. Limit the results to books that have been checked out more than 5 times. Sort the results in descending order by the number of times checked out and then by title. (See the figure below for the output)

BOOK_NUM	BOOK_TITLE	Times Checked Out
5236	Database in the Cloud	12
5235	Beginner's Guide to JAVA	9
5240	iOS Programming	7
5238	Conceptual Programming	6

**Answer provided:**

```
SELECT BOOK.BOOK_NUM, BOOK_TITLE, Count(CHECK_NUM) AS "Times
Checked Out"
FROM BOOK JOIN CHECKOUT ON BOOK.BOOK_NUM = CHECKOUT.BOOK_NUM
GROUP BY BOOK.BOOK_NUM, BOOK_TITLE
HAVING Count(CHECK_NUM) > 5
ORDER BY Count(CHECK_NUM) DESC, BOOK_TITLE;
```

21. Write a query to display the patron ID, last name, number of times that patron has ever checked out a book, and the number of different books the patron has ever checked out. For example, if a given patron has checked out the same book twice, that would count as two checkouts but only one book. Limit the results to only patrons that have made at least three checkouts. Sort the results in descending order by number of books, then in descending order by number of checkouts, and then in ascending order by patron ID. (See the figure below for the output)

PAT_ID	PAT_LNAME	NUM CHECKOUTS	NUM DIFFERENT BOOKS
1161	Koch	3	3
1165	Baldwin	3	3
1181	Home	3	3
1185	Yang	3	3
1210	Cooley	3	3
1229	Burke	3	3
1160	carter	3	2
1171	Marsh	3	2
1172	Miles	3	2
1207	Ramos	3	2
1209	Mathis	3	2
1183	Hughes	3	1

22. Write a query to display the average number of days a book is kept during a checkout. (See the figure below for the output)

Average Days Kept
4.44

**Answer provided:**

```
SELECT Round(Avg(datediff(CHECK_IN_DATE, CHECK_OUT_DATE)), 2)
AS "Average Days Kept"
FROM CHECKOUT;
```

23. Write a query to display the patron ID and the average number of days that patron keeps books during a checkout. Limit the results to only patrons who have at least three checkouts. Sort the results in descending order by the average days the book is kept. (See the figure below for the output)

PAT_ID	Average Days Kept
1160	7.00
1185	6.67
1165	5.67
1207	5.50
1209	5.33
1172	4.50
1183	4.33
1171	3.67
1181	3.67
1161	3.33
1210	2.33
1229	2.00

---

This is the end of Prac 8.

You are required to submit via LearnJCU one WORD document containing all SQL queries you composed for this Prac.

**Marking Criteria:**

**You will be given 0 to 3 marks depending on your completeness/correctness of your work.**

- 0 or 0.5 – not attempted or mostly wrong or not reasonable logic used in most queries
- 1, 1.5 or 2 – attempted with good efforts but not fully correct. 40%~70% queries are fully correct
- 2.5 or 3 – fully correct and reasonable or very minor errors caused by simple mistake, missing, unnecessary complication etc. 80%~100% queries are fully correct

Please note that the individual feedback for each answer you submit will not be provided but once all markings are done the sample solution will be provided so that you can refer to them for your self-review.

---