CP5804 Database Systems

Week 6: Lab activities

SQL Practice II:

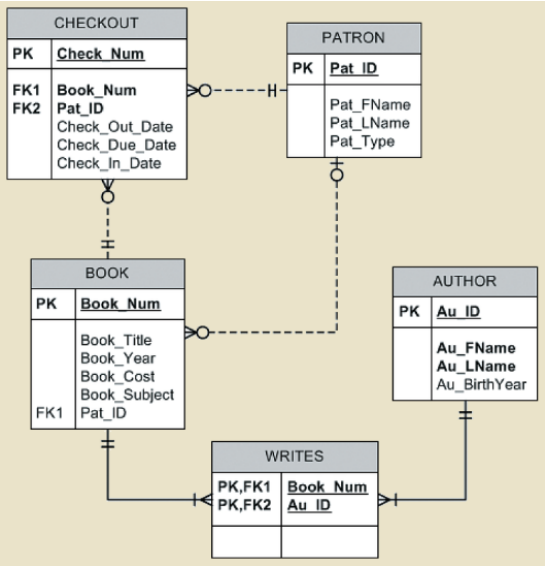
Writing/Executing advanced SQL operators in MySQL Workbench to extract information from a database

In this lab, you are going to practice advanced SQL query techniques to extract useful information using MySQL Workbench. In particular, you will practice to use JOIN operator syntax and to use subqueries for various purposes.

To be continuous from the previous lab, you will use the same library database you created and used in previous labs, and write a number of SQL queries to extract information from the database.

- **Learning outcomes and objectives**
  Student will be able to
  - compose various SQL queries in particular using JOIN operators to extract information from two or more tables
  - write subqueries when it is necessary to process data based on other processed data

- **Task Overview**
  Open the library database you created in the previous lab on MySQL Workbench. The conceptual model of the database is presented in the ERD as shown here. You will have to refer to this ERD when you compose SQL queries for this lab, to understand further details about this database including table structure, columns included in each table, PKs and FKs, relationships between tables etc.



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

  Do not forget to save each query as soon as after writing/running it. For example, save the query as 'Q1.sql' for the first question. You are required to submit a zipped directory containing all query files you completed as Q1~Q29 to be marked off for this lab activity.

Exercises:

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.

Save a query for each question as the name of "Q1.sql", "Q2.sql", … You are required to submit these files (as a zipped folder) to be marked off.

Note that some questions are provided with the accompanied solution to help your learning.

1. Write a query to display the patron ID, book number, and days kept for each checkout. "Days Kept" is the difference from the date on which the book is returned to the date it was checked out. (See the figure below for the output)

PATRON	BOOK	Days Kept
1165	5235	9
1209	5238	5
1160	5240	9
1160	5237	3
1202	5236	8
1203	5235	8
1174	5244	3
1181	5248	1
1170	5242	4
1161	5235	0

Answer provided:

```
SELECT PAT_ID AS PATRON,
       BOOK_NUM AS BOOK,
       DATEDIFF(CHECK_IN_DATE,
                CHECK_OUT_DATE)
       AS "Days Kept"
FROM CHECKOUT;
```

2. Write a query to display the patron ID, patron full name, and patron type for each patron. (See the figure below for the output)

PAT_ID	Patron Name	PAT_TYPE
1160	robert carter	Faculty
1161	Kelsey Koch	Faculty
1165	Cedric Baldwin	Faculty
1166	Vera Alvarado	Student
1167	Alan Martin	FACULTY
1170	Cory Barry	faculty
1171	Peggy Marsh	STUDENT

Answer provided:

```
SELECT PAT_ID,
       CONCAT(PAT_FNAME, ' ', PAT_LNAME) AS "Patron Name",
       PAT_TYPE
FROM PATRON;
```

Note: CONCAT is a special function compatible with MySQL which can be used to concatenate multiple columns data into one. If you use any other DBMS, you need to refer to the manual of the DBMS for further information about the specific functions they provide for concatenating two columns data into one.

3. **Write a query to display the book number, title with year, and subject for each book. (See the figure below for the output)**

BOOK_NUM	BOOK	BOOK_SUBJECT
5235	Beginner's Guide to JAVA (2012)	Programming
5236	Database in the Cloud (2012)	Cloud
5237	Mastering the database environment (2013)	Database
5238	Conceptual Programming (2013)	Programming
5239	J++ in Mobile Apps (2013)	Programming
5240	iOS Programming (2013)	Programming
5241	JAVA First Steps (2013)	Programming
5242	C# in Middleware Deployment (2013)	Middleware
5243	DATABASES in Theory (2013)	Database

4. **Write a query to display the author last name, author first name, and book number for each book written by that author. (See the figure below for the output)**

AU_LNAME	AU_FNAME	BOOK_NUM
Reeves	Benson	5237
Reeves	Benson	5253
Beatney	Rachel	5240
Salvadore	Carmine	5239
Salvadore	Carmine	5248
Bruer	Hugo	5243
Bruer	Hugo	5246
Chiang	Xia	5244
Chiang	Xia	5249
Chiang	Xia	5252
Durante	Reba	5235
Tankersly	Trina	5244

Answer provided:

```
SELECT AU_LNAME,
       AU_FNAME,
       BOOK_NUM
FROM AUTHOR
JOIN WRITES USING (AU_ID);
```

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.

5. **Write a query to display the author ID, book number, title, and year for each book. (See the figure below for the output)**

AU_ID	BOOK_NUM	BOOK_TITLE	BOOK_YEAR
273	5235	Beginner's Guide to JAVA	2012
383	5236	Database in the Cloud	2012
185	5237	Mastering the database environment	2013
603	5238	Conceptual Programming	2013
229	5239	J++ in Mobile Apps	2013
460	5239	J++ in Mobile Apps	2013
592	5239	J++ in Mobile Apps	2013
218	5240	iOS Programming	2013
460	5241	JAVA First Steps	2013
559	5241	JAVA First Steps	2013
581	5242	C# in Middleware Deployment	2013
251	5243	DATABASES in Theory	2013

6. **Write a query to display the author last name, first name, book title, and year for each book. (See the figure below for the output)**

AU_LNAME	AU_FNAME	BOOK_TITLE	BOOK_YEAR
Durante	Reba	Beginner's Guide to JAVA	2012
Walsh	Neal	Database in the Cloud	2012
Reeves	Benson	Mastering the database environment	2013
Palca	Julia	Conceptual Programming	2013
Salvadore	Carmine	J++ in Mobile Apps	2013
Paulsen	Connie	J++ in Mobile Apps	2013
Sheel	Lawrence	J++ in Mobile Apps	2013
Beatney	Rachel	iOS Programming	2013
Paulsen	Connie	JAVA First Steps	2013
McGill	Rachel	JAVA First Steps	2013

Answer provided:

```
SELECT AU_LNAME,
       AU_FNAME,
       BOOK_TITLE,
       BOOK_YEAR
FROM AUTHOR
JOIN WRITES USING (AU_ID)
JOIN BOOK USING (BOOK_NUM);
```

7. **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 described in the assignment instructions 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

8. **Write a query to display the patron ID, full name (first and last), and patron type for all patrons. Sort the results by patron type, then by last name and first name. Ensure that all sorting is case insensitive. (See the figure below for the output)**

PAT_ID	NAME	PAT_TYPE
1165	Cedric Baldwin	Faculty
1170	Cory Barry	faculty
1160	robert carter	Faculty
1183	Helena Hughes	Faculty
1161	Kelsey Koch	Faculty
1167	Alan Martin	FACULTY
1166	Vera Alvarado	Student
1202	Holly Anthony	Student
1180	Nadine Blair	STUDENT

9. **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. (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

Answer provided:

```
SELECT BOOK_NUM,
       Count(*) AS "Times Checked Out"
FROM CHECKOUT
GROUP BY BOOK_NUM
ORDER BY "Times Checked Out" DESC,
         BOOK_NUM;
```

10. **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

11. 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

12. 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 times checked out, 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_NUM,
       BOOK_TITLE,
       Count(CHECK_NUM) AS `Times Checked Out`
FROM BOOK
LEFT JOIN CHECKOUT USING (BOOK_NUM)
GROUP BY BOOK_NUM,
```

```
          BOOK_TITLE
ORDER BY `Times Checked Out` DESC,
          BOOK_TITLE;
```

13. **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_NUM,
       BOOK_TITLE,
       Count(CHECK_NUM) AS `Times Checked Out`
FROM BOOK
JOIN CHECKOUT USING (BOOK_NUM)
GROUP BY BOOK_NUM,
          BOOK_TITLE HAVING Count(CHECK_NUM) > 5
ORDER BY `Times Checked Out` DESC,
          BOOK_TITLE;
```

14. **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". (See the figure below for the output)**

AU_ID	AU_LNAME	BOOK_TITLE	CHECK_OUT_DATE	PAT_LNAME
251	Bruer	Capture the Cloud	4/21/2015	Miles
251	Bruer	Capture the Cloud	5/15/2015	Miles

15. **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 then first name. (See the figure below for the output) – Hint: use LEFT JOIN operator**

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

16. **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 2 checkouts but only 1 book. Limit the results to only patrons that have made at least 3 checkouts. Sort the results in descending order by number of books, then in descending order by number of checkouts, 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	Horne	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

Answer provided:

```
SELECT PAT_ID,
       PAT_LNAME,
       Count(CHECK_NUM) AS `NUM CHECKOUTS`,
       Count(DISTINCT BOOK_NUM) AS `NUM DIFFERENT BOOKS`
FROM CHECKOUT
JOIN PATRON USING (PAT_ID)
GROUP BY PAT_ID,
         PAT_LNAME
HAVING Count(CHECK_NUM) > 2
```

```
ORDER BY `NUM DIFFERENT BOOKS` DESC,
         `NUM CHECKOUTS` DESC,
         PAT_ID;
```

17. **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;
```

18. **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 that have at least 3 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
1185	6.67
1165	5.67
1207	5.5
1209	5.33
1172	4.5
1183	4.33
1181	3.67
1171	3.67
1161	3.33
1210	2.33
1229	2

19. **Write a query to display the book number, title, and cost of books that have the lowest cost of any books in the system. Sort the results by book number. (See the figure below for the output)**

BOOK_NUM	BOOK_TITLE	BOOK_COST
5239	J++ in Mobile Apps	49.95
5241	JAVA First Steps	49.95
5248	What You Always Wanted to Know About Database, But Were Afraid to Ask	49.95
5254	Coding Style for Maintenance	49.95

Answer provided:

```
SELECT BOOK_NUM,
       BOOK_TITLE,
       BOOK_COST
FROM BOOK
WHERE BOOK_COST =
    (SELECT Min(BOOK_COST)
     FROM BOOK)
ORDER BY BOOK_NUM;
```

20. Write a query to display the author ID, first and last name for all authors that have never written a book with the subject Programming. Sort the results by author last name. (See the figure below for the output)

AU_ID	AU_FNAME	AU_LNAME
581	Manish	Aggerwal
251	Hugo	Bruer
262	Xia	Chiang
438	Perry	Pearson
284	Trina	Tankersly
383	Neal	Walsh

Answer provided:

```
SELECT AU_ID,
       AU_FNAME,
       AU_LNAME
FROM AUTHOR
WHERE AU_ID NOT IN
    (SELECT AU_ID
     FROM BOOK
     JOIN WRITES USING (BOOK_NUM)
     WHERE BOOK_SUBJECT = 'Programming')
ORDER BY AU_LNAME;
```

21. Write a query to display the book number, title, subject, average cost of books within that subject, and the difference between each book's cost and the average cost of books in that subject. Sort the results by book title. (See the figure below for the output)

BOOK_NUM	BOOK_TITLE	BOOK_SUBJECT	AVGCOST	DIFFERENCE
5235	Beginner's Guide to JAVA	Programming	66.62	-6.67
5236	Database in the Cloud	Cloud	72.45	7.5
5237	Mastering the database environment	Database	84.95	5
5238	Conceptual Programming	Programming	66.62	-6.67
5239	J++ in Mobile Apps	Programming	66.62	-16.67
5240	iOS Programming	Programming	66.62	13.33
5241	JAVA First Steps	Programming	66.62	-16.67
5242	C# in Middleware Deployment	Middleware	89.95	-30
5243	DATABASES in Theory	Database	84.95	45
5244	Cloud-based Mobile Applications	Cloud	72.45	-2.5
5245	The Golden Road to Platform independence	Middleware	89.95	30
5246	Capture the Cloud	Cloud	72.45	-2.5
5247	Shining Through the Cloud: Sun Programming	Programming	66.62	43.33
5248	What You Always Wanted to Know About Database, But Were Afraid to Ask	Database	84.95	-35
5249	Starlight Applications	Cloud	72.45	-2.5
5250	Reengineering the Middle Tier	Middleware	89.95	0
5251	Thoughts on Revitalizing Ruby	Programming	66.62	-6.67
5252	Beyond the Database Veil	Database	84.95	-15
5253	Virtual Programming for Virtual Environments	Programming	66.62	13.33
5254	Coding Style for Maintenance	Programming	66.62	-16.67

Answer provided:

```
SELECT BOOK_NUM,
       BOOK_TITLE,
       BOOK_SUBJECT,
       Round(AVGCOST, 2) AS "Average Subject Cost",
       BOOK_COST - Round(AVGCOST, 2) AS DIFFERENCE
FROM BOOK
```

```
JOIN
   (SELECT BOOK_SUBJECT,
           Avg(BOOK_COST) AS AVGCOST
    FROM BOOK
    GROUP BY BOOK_SUBJECT) AS SUB USING (BOOK_SUBJECT)
ORDER BY BOOK_TITLE;
```

22. **Write a query to display the book number, title, subject, author last name, and the number of books written by that author. Limit the results to books in the Cloud subject. Sort the results by book title and then author last name. (See the figure below for the output)**

BOOK_NUM	BOOK_TITLE	BOOK_SUBJECT	AU_LNAME	Num Books by Author
5246	Capture the Cloud	Cloud	Bruer	2
5244	Cloud-based Mobile Applications	Cloud	Chiang	3
5244	Cloud-based Mobile Applications	Cloud	Tankersly	1
5236	Database in the Cloud	Cloud	Walsh	2
5249	Starlight Applications	Cloud	Chiang	3

23. **Write a query to display the lowest average cost of books within a subject and the highest average cost of books within a subject. (See the figure below for the output)**

Lowest Avg Cost	Highest Avg Cost
66.62	89.95

This is the end of Week 6 Lab.

You are required to submit a zipped folder containing all SQL query files you created.
