LAB EXPERIMENTS

PART A: SQL PROGRAMMING

A. Consider the following schema for a Library Database:

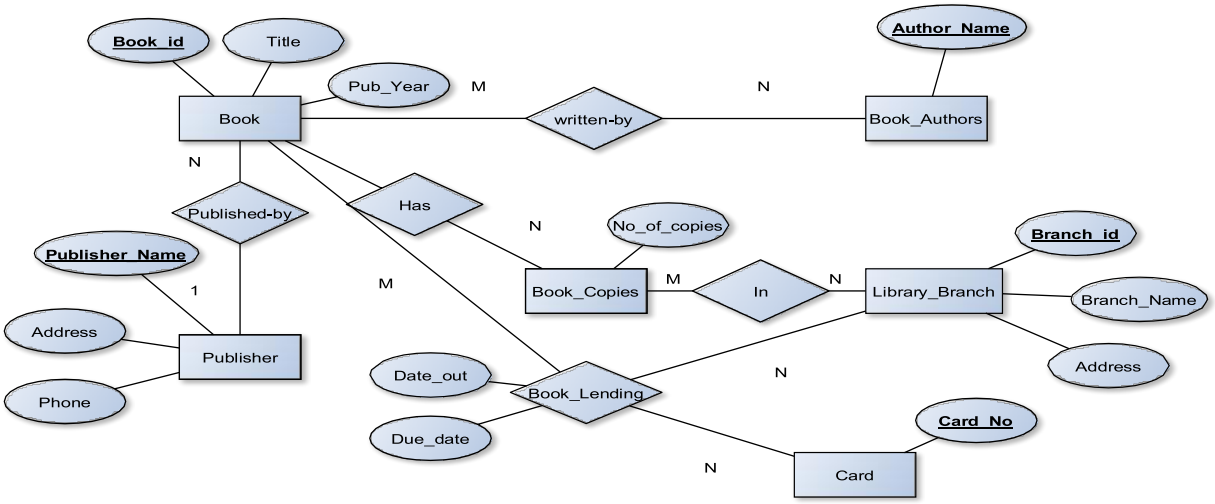
BOOK (*Book_id, Title, Publisher_Name, Pub_Year*)
BOOK_AUTHORS (*Book_id, Author_Name*)
PUBLISHER (*Name, Address, Phone*)
BOOK_COPIES (*Book_id, Branch_id, No-of_Copies*)
BOOK_LENDING (*Book_id, Branch_id, Card_No, Date_Out, Due_Date*)
LIBRARY_BRANCH (*Branch_id, Branch_Name, Address*)

Write SQL queries to

1. **Retrieve details of all books in the library – id, title, name of publisher, authors, number of copies in each branch, etc.**
2. **Get the particulars of borrowers who have borrowed more than 3 books, but from Jan 2017 to Jun 2017**
3. **Delete a book in BOOK table. Update the contents of other tables to reflect this data manipulation operation.**
4. **Partition the BOOK table based on year of publication. Demonstrate its working with a simple query.**
5. **Create a view of all books and its number of copies that are currently available in the Library.**

Solution:

Entity-Relationship Diagram



Schema Diagram

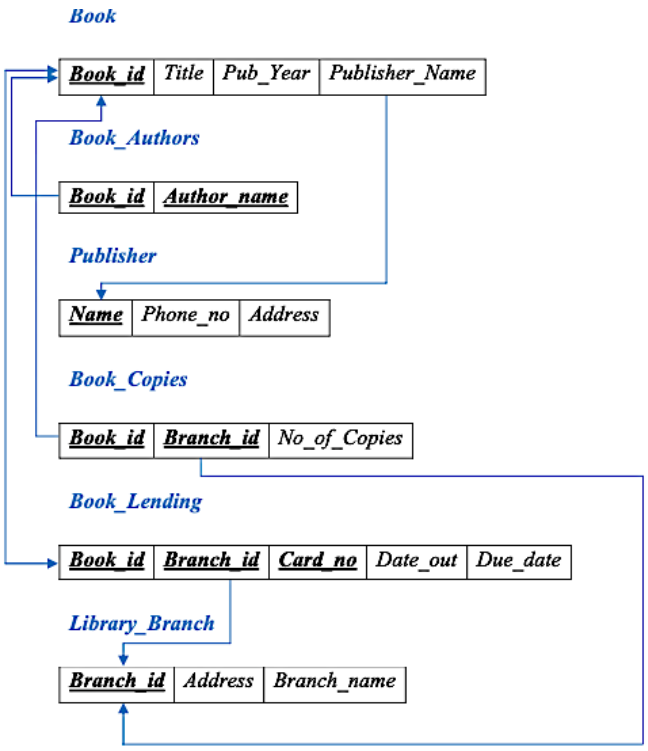


Table Creation

```
CREATE TABLE PUBLISHER
(
NAME VARCHAR(20) PRIMARY KEY,
PHONE VARCHAR(10),
ADDRESS VARCHAR(20)
);

CREATE TABLE BOOK
(
BOOK_ID INTEGER PRIMARY KEY,
TITLE VARCHAR(20),
PUB_YEAR VARCHAR(20),
PUBLISHER_NAME varchar(20),
FOREIGN KEY(PUBLISHER_NAME)  REFERENCES PUBLISHER(NAME) ON DELETE
CASCADE
);

CREATE TABLE BOOK_AUTHORS
(
```

```
AUTHOR_NAME VARCHAR(20),
BOOK_ID INTEGER ,
FOREIGN KEY(BOOK_ID) REFERENCES BOOK(BOOK_ID) ON DELETE CASCADE,
PRIMARY KEY(BOOK_ID, AUTHOR_NAME)
);

CREATE TABLE LIBRARY_BRANCH
(
BRANCH_ID INTEGER PRIMARY KEY,
BRANCH_NAME VARCHAR(50),
ADDRESS VARCHAR(50)
);

CREATE TABLE BOOK_COPIES
(
NO_OF_COPIES INTEGER,
BOOK_ID INTEGER,
BRANCH_ID INTEGER,
PRIMARY KEY(BOOK_ID, BRANCH_ID),
FOREIGN KEY(BOOK_ID) REFERENCES BOOK (BOOK_ID) ON DELETE CASCADE,
FOREIGN KEY(BRANCH_ID) REFERENCES LIBRARY_BRANCH(BRANCH_ID) ON
DELETE  CASCADE
);

CREATE TABLE CARD
(
CARD_NO INTEGER PRIMARY KEY
);

CREATE TABLE BOOK_LENDING
(
DATE_OUT DATE,
DUE_DATE DATE,
BOOK_ID INTEGER,
BRANCH_ID INTEGER,
CARD_NO INTEGER,
PRIMARY KEY (BOOK_ID, BRANCH_ID, CARD_NO),
FOREIGN KEY(BOOK_ID) REFERENCES BOOK (BOOK_ID) ON DELETE CASCADE,
FOREIGN KEY(BRANCH_ID) REFERENCES LIBRARY_BRANCH(BRANCH_ID) ON
DELETE CASCADE,
FOREIGN KEY(CARD_NO) REFERENCES CARD (CARD_NO) ON DELETE CASCADE
```

);

Table Descriptions

DESC PUBLISHER;

```
SQL> desc publisher;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 NAME                                      NOT NULL VARCHAR2(20)
 PHONE                                              NUMBER(38)
 ADDRESS                                            VARCHAR2(20)
```

DESC BOOK;

```
SQL> DESC BOOK;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 BOOK_ID                                   NOT NULL NUMBER(38)
 TITLE                                              VARCHAR2(20)
 PUB_YEAR                                           VARCHAR2(20)
 PUBLISHER_NAME                                     VARCHAR2(20)
```

DESC BOOK_AUTHORS;

```
SQL> DESC BOOK_AUTHORS;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 AUTHOR_NAME                               NOT NULL VARCHAR2(20)
 BOOK_ID                                   NOT NULL NUMBER(38)
```

DESC LIBRARY_BRANCH;

```
SQL> DESC LIBRARY_BRANCH;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 BRANCH_ID                                 NOT NULL NUMBER(38)
 BRANCH_NAME                                        VARCHAR2(50)
 ADDRESS                                            VARCHAR2(50)
```

DESC BOOK_COPIES;

```
SQL> DESC BOOK_COPIES;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 NO_OF_COPIES                                       NUMBER(38)
 BOOK_ID                                   NOT NULL NUMBER(38)
 BRANCH_ID                                 NOT NULL NUMBER(38)
```

DESC CARD;

```
SQL> DESC CARD;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CARD_NO                                   NOT NULL NUMBER(38)
```

DESC BOOK_LENDING;

```
SQL> desc book_lending;
 Name
 -----------------------------------------------------------------------------------------------------------
 DATE_OUT
 DUE_DATE
 BOOK_ID
 BRANCH_ID
 CARD_NO
```

Insertion of Values to Tables

INSERT INTO PUBLISHER VALUES (‘MCGRAW-HILL’, 9989076587, ‘BANGALORE’);
INSERT INTO PUBLISHER VALUES (‘PEARSON’, 9889076565, ‘NEWDELHI’);
INSERT INTO PUBLISHER VALUES (‘RANDOM HOUSE’, 7455679345, ‘HYDRABAD’);
INSERT INTO PUBLISHER VALUES (‘HACHETTE LIVRE’, 8970862340, ‘CHENAI’);
INSERT INTO PUBLISHER VALUES (‘GRUPO PLANETA’, 7756120238, ‘BANGALORE’);

INSERT INTO BOOK VALUES (1,’DBMS’,’JAN-2017’, ‘MCGRAW-HILL’);
INSERT INTO BOOK VALUES (2,’ADBMS’,’JUN-2016’, ‘MCGRAW-HILL’);
INSERT INTO BOOK VALUES (3,’CN’,’SEP-2016’, ‘PEARSON’);
INSERT INTO BOOK VALUES (4,’CG’,’SEP-2015’, ‘GRUPO PLANETA’);
INSERT INTO BOOK VALUES (5,’OS’,’MAY-2016’, ‘PEARSON’);

INSERT INTO BOOK_AUTHORS VALUES (’NAVATHE’, 1);
INSERT INTO BOOK_AUTHORS VALUES (’NAVATHE’, 2);
INSERT INTO BOOK_AUTHORS VALUES (’TANENBAUM’, 3);
INSERT INTO BOOK_AUTHORS VALUES (’EDWARD ANGEL’, 4);
INSERT INTO BOOK_AUTHORS VALUES (’GALVIN’, 5);

INSERT INTO LIBRARY_BRANCH VALUES (10,’RR NAGAR’,’BANGALORE’);
INSERT INTO LIBRARY_BRANCH VALUES (11,’RNSIT’,’BANGALORE’);
INSERT INTO LIBRARY_BRANCH VALUES (12,’RAJAJI NAGAR’, ’BANGALORE’);
INSERT INTO LIBRARY_BRANCH VALUES (13,’NITTE’,’MANGALORE’);
INSERT INTO LIBRARY_BRANCH VALUES (14,’MANIPAL’,’UDUPI’);

INSERT INTO BOOK_COPIES VALUES (10, 1, 10);
INSERT INTO BOOK_COPIES VALUES (5, 1, 11);
INSERT INTO BOOK_COPIES VALUES (2, 2, 12);
INSERT INTO BOOK_COPIES VALUES (5, 2, 13);
INSERT INTO BOOK_COPIES VALUES (7, 3, 14);
INSERT INTO BOOK_COPIES VALUES (1, 5, 10);
INSERT INTO BOOK_COPIES VALUES (3, 4, 11);

```
INSERT INTO CARD VALUES (100);
INSERT INTO CARD VALUES (101);
INSERT INTO CARD VALUES (102);
INSERT INTO CARD VALUES (103);
INSERT INTO CARD VALUES (104);

INSERT INTO BOOK_LENDING VALUES ('2017-01-01','2017-06-07', 1, 10, 101);
INSERT INTO BOOK_LENDING VALUES ('2017-01-11','2017-03-11', 3, 14, 101);
INSERT INTO BOOK_LENDING VALUES ('2017-02-21','2017-04-21', 2, 13, 101);
INSERT INTO BOOK_LENDING VALUES ('2017-03-15','2017-07-15', 4, 11, 101);
INSERT INTO BOOK_LENDING VALUES ('2017-04-12','2017-05-12', 1, 11, 104);
```

SELECT * FROM PUBLISHER;

```
SQL> select * from publisher;

NAME                      PHONE ADDRESS
-------------------- ---------- --------------------
MCGRAW-HILL          9989076587 BANGALORE
PEARSON              9889076565 NEWDELHI
RANDOM HOUSE         7455679345 HYDRABAD
HACHETTE LIVRE       8970862340 CHENAI
GRUPO PLANETA        7756120238 BANGALORE
```

SELECT * FROM BOOK;

```
SQL> SELECT * FROM BOOK;

   BOOK_ID TITLE                PUB_YEAR             PUBLISHER_NAME
---------- -------------------- -------------------- --------------------
         1 DBMS                 JAN-2017             MCGRAW-HILL
         2 ADBMS                JUN-2016             MCGRAW-HILL
         3 CN                   SEP-2016             PEARSON
         4 CG                   SEP-2015             GRUPO PLANETA
         5 OS                   MAY-2016             PEARSON
```

SELECT * FROM BOOK_AUTHORS;

```
SQL> SELECT * FROM BOOK_AUTHORS;

AUTHOR_NAME             BOOK_ID
-------------------- ----------
NAVATHE                      1
NAVATHE                      2
TANENBAUM                    3
EDWARD ANGEL                 4
GALVIN                       5
```

SELECT * FROM LIBRARY_BRANCH;

```
SQL> SELECT * FROM LIBRARY_BRANCH;

 BRANCH_ID BRANCH_NAME                                        ADDRESS
---------- -------------------------------------------------- --------------------------------------
        10 RR NAGAR                                           BANGALORE
        11 RNSIT                                              BANGALORE
        12 RAJAJI NAGAR                                       BANGALORE
        13 NITTE                                              MANGALORE
        14 MANIPAL                                            UDUPI
```

SELECT * FROM BOOK_COPIES;

```
SQL> SELECT * FROM BOOK_COPIES;

NO_OF_COPIES    BOOK_ID  BRANCH_ID
------------ ---------- ----------
          10          1         10
           5          1         11
           2          2         12
           5          2         13
           7          3         14
           1          5         10
           3          4         11
```

SELECT * FROM CARD;

```
SQL> SELECT * FROM CARD;

   CARD_NO
----------
       100
       101
       102
       103
       104
```

SELECT * FROM BOOK_LENDING;

```
SQL> select * from book_lending;

DATE_OUT  DUE_DATE     BOOK_ID  BRANCH_ID    CARD_NO
--------- --------- ---------- ---------- ----------
01-JAN-17 01-JUN-17          1         10        101
11-JAN-17 11-MAR-17          3         14        101
21-FEB-17 21-APR-17          2         13        101
15-MAR-17 15-JUL-17          4         11        101
12-APR-17 12-MAY-17          1         11        104
```

Queries:

1. **Retrieve details of all books in the library – id, title, name of publisher, authors, number of copies in each branch, etc.**

```
SELECT B.BOOK_ID, B.TITLE, B.PUBLISHER_NAME, A.AUTHOR_NAME,
C.NO_OF_COPIES, L.BRANCH_ID
FROM BOOK B, BOOK_AUTHORS A, BOOK_COPIES C, LIBRARY_BRANCH L
WHERE B.BOOK_ID=A.BOOK_ID AND B.BOOK_ID=C.BOOK_ID AND
L.BRANCH_ID =C.BRANCH_ID;
```

BOOK_ID	TITLE	PUBLISHER_NAME	AUTHOR_NAME	NO_OF_COPIES	BRANCH_ID
1	DBMS	MCGRAW-HILL	NAVATHE	10	10
1	DBMS	MCGRAW-HILL	NAVATHE	5	11
2	ADBMS	MCGRAW-HILL	NAVATHE	2	12
2	ADBMS	MCGRAW-HILL	NAVATHE	5	13
3	CN	PEARSON	TANENBAUM	7	14
5	OS	PEARSON	GALVIN	1	10
4	CG	GRUPO PLANETA	EDWARD ANGEL	3	11

2. **Get the particulars of borrowers who have borrowed more than 3 books, but from Jan 2017 to Jun 2017.**

```
SELECT CARD_NO
FROM BOOK_LENDING
WHERE DATE_OUT BETWEEN '2017-JAN-01' AND '2017-JUN-30'
GROUP BY CARD_NO
HAVING COUNT (*)>3;
```

CARD_NO
101

3. **Delete a book in BOOK table. Update the contents of other tables to reflect this data manipulation operation.**

```
DELETE FROM BOOK
WHERE BOOK_ID=3;
```

```
SQL> DELETE FROM BOOK
  2  WHERE BOOK_ID=3;

1 row deleted.

SQL> SELECT * FROM BOOK;

   BOOK_ID TITLE                PUB_YEAR             PUBLISHER_NAME
---------- -------------------- -------------------- --------------------
         1 DBMS                 JAN-2017             MCGRAW-HILL
         2 ADBMS                JUN-2016             MCGRAW-HILL
         4 CG                   SEP-2015             GRUPO PLANETA
         5 OS                   MAY-2016             PEARSON
```

.

4. **Partition the BOOK table based on year of publication. Demonstrate its working with asimple query.**

```
CREATE VIEW V_PUBLICATION AS
SELECT PUB_YEAR
FROM BOOK;
```

```
PUB_YEAR
-------------
JAN-2017
JUN-2016
SEP-2016
SEP-2015
MAY-2016
```

Or

Partitioning can be achieved without splitting tables by physically putting tables on individual disk drives. Partitioning allows tables, indexes, and index-organized tables to be subdivided into smaller pieces, therefore queries that access only a fraction of the data can run faster because there is fewer data to scan. There are two major forms of partitioning :

Horizontal Partitioning : Horizontal partitioning divides table rows into multiple partitions (based on a logic).

Vertical Partitioning : Vertical partitioning divides a table into multiple tables that contain fewer columns.

In MySQL you can partition a table using **CREATE TABLE** or **ALTER TABLE** command.

```
CREATE TABLE BOOKP
  (
  BOOK_ID  INT  NOT NULL ,
  TITLE  VARCHAR(20),
  PUBLISHER_NAME  VARCHAR(20),
  PUB_YEAR  INT
  )
 PARTITION BY RANGE (Pub_year)
( PARTITION  q0  VALUES  LESS  THAN  (2015),
 PARTITION  q1  VALUES  LESS  THAN  (2016),
 PARTITION    q2  VALUES LESS THAN (2017)
);
```

INSERT INTO BOOKP **VALUES** ('801' , 'DBMS','Willey', '2013');
INSERT INTO BOOKP **VALUES** ('802' , 'DS','Pearson', '2014');
INSERT INTO BOOKP **VALUES** ('803' , 'OS','Willey', '2015');
INSERT INTO BOOKP **VALUES** ('804' , 'CG','MC-GRAW-HILL', '2016');

```
SELECT * FROM BOOKP;
+---------+-------+----------------+----------+
| Book_iD | Title | Publisher_name | Pub_yeaR |
+---------+-------+----------------+----------+
|     801 | DBMS  | Willey         |     2013 |
|     802 | DS    | Pearson        |     2014 |
|     803 | OS    | Willey         |     2015 |
|     804 | CG    | MC-GRAW-HILL   |     2016 |
+---------+-------+----------------+----------+
 4 rows in set (0.00 sec)

SELECT * FROM BOOKP PARTITION(Q1);
+---------+-------+----------------+----------+
| Book_iD | Title | Publisher_name | Pub_yeaR |
+---------+-------+----------------+----------+
|     803 | OS    | Willey         |     2015 |
+---------+-------+----------------+----------+
 1 row in set (0.00 sec)

SELECT * FROM BOOKP PARTITION(Q0);
+---------+-------+----------------+----------+
| Book_iD | Title | Publisher_name | Pub_yeaR |
+---------+-------+----------------+----------+
|     801 | DBMS  | Willey         |     2013 |
|     802 | DS    | Pearson        |     2014 |
+---------+-------+----------------+----------+
 2 rows in set (0.00 sec)

SELECT * FROM BOOKP PARTITION(Q2);
+---------+-------+----------------+----------+
| Book_iD | Title | Publisher_name | Pub_yeaR |
+---------+-------+----------------+----------+
|     804 | CG    | MC-GRAW-HILL   |     2016 |
+---------+-------+----------------+----------+
 1 row in set (0.00 sec)
```

5. **Create a view of all books and its number of copies that are currently available in the Library.**

```
CREATE VIEW V_BOOKS AS
SELECT B.BOOK_ID, B.TITLE, C.NO_OF_COPIES
FROM BOOK B, BOOK_COPIES C, LIBRARY_BRANCH L
WHERE B.BOOK_ID=C.BOOK_ID AND C.BRANCH_ID=L.BRANCH_ID;
```

BOOK_ID	TITLE	NO_OF_COPIES
1	DBMS	10
1	DBMS	5
2	ADBMS	2
2	ADBMS	5
3	CN	7
5	OS	1
4	CG	3