# LAB EXERCISE - 1

## LIBRARY DATABASE

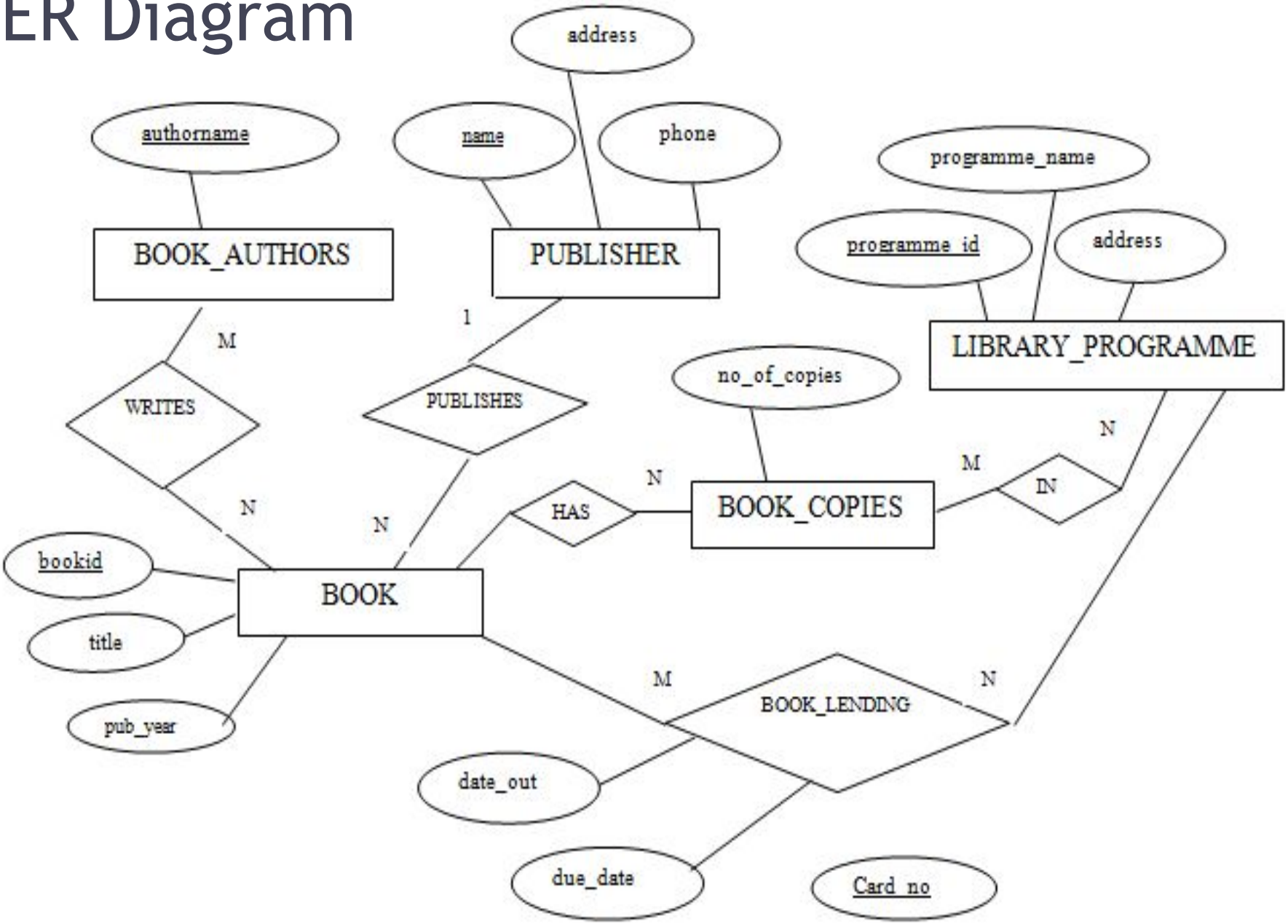
1. Consider the following schema for a Library Database:

- BOOK(**Book_id**, Title, Publisher_Name, Pub_Year)
- BOOK_AUTHORS(Book_id, Author_Name)
- PUBLISHER(Name, Address, Phone)
- BOOK_COPIES(Book_id, Programme_id, No-of_Copies)
- BOOK_LENDING(Book_id, Programme_id, Card_No, Date_Out, Due_Date)
- LIBRARY_PROGRAMME(Programme_id, Programme_Name, Address)

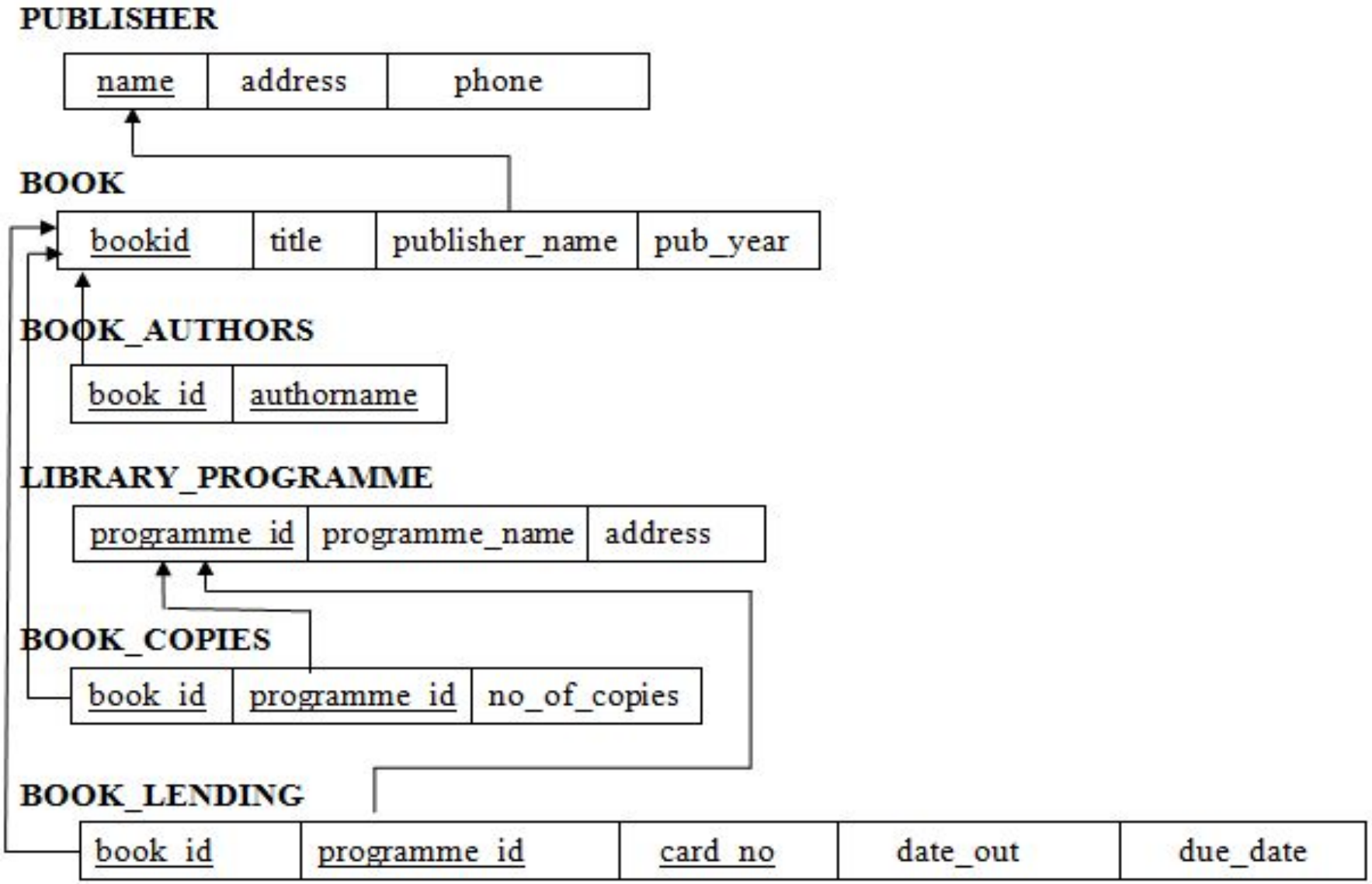
**Write SQL queries to**

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

# ER Diagram


address
authorname
name
phone
programme_name
BOOK_AUTHORS
PUBLISHER
programme_id
address
LIBRARY_PROGRAMME
M
1
WRITES
PUBLISHES
no_of_copies
N
M
N
IN
N
HAS
N
BOOK_COPIES
N
bookid
BOOK
title
M
N
BOOK_LENDING
pub_year
date_out
due_date
Card_no

# Schema Diagram



```
CREATE TABLE PUBLISHER(
   name VARCHAR2(20) PRIMARY KEY,
   address VARCHAR2(20),
   phone NUMBER(10));
```

```
INSERT INTO PUBLISHER VALUES
   ('Pearson','London',9874522224);
INSERT INTO PUBLISHER VALUES
   ('TataMcGraw','NewYork',9858523565);
INSERT INTO PUBLISHER VALUES
   ('Oxford','UK',9885121112);
INSERT INTO PUBLISHER VALUES
   ('Cambridge','UK',9785634615);
INSERT INTO PUBLISHER VALUES
   ('OReilly','California',9994125455);
```

```
SQL> select *from publisher;
```

| NAME | ADDRESS | PHONE |
|---|---|---|
| Pearson | London | 9874522224 |
| TataMcGraw | NewYork | 9858523565 |
| Oxford | UK | 9885121112 |
| Cambridge | UK | 9785634615 |
| OReilly | California | 9994125455 |

```
CREATE TABLE BOOK (
    bookid INT PRIMARY KEY,
        title  VARCHAR2(40),
    publisher_name VARCHAR2(20) REFERENCES
    publisher ON DELETE CASCADE,
    pub_year INT);
```

**OR**

```
CREATE TABLE BOOK (
    bookid INT PRIMARY KEY,
    title  VARCHAR2(40),
    publisher_name VARCHAR2(20) REFERENCES
    publisher(name) ON DELETE CASCADE,
    pub_year INT);
```

```
INSERT INTO BOOK VALUES
   (1,'DBMS','Pearson',2017);
INSERT INTO BOOK VALUES
   (2,'Hadoop','Pearson',2000);
INSERT INTO BOOK VALUES
   (3,'AIML','TataMcGraw',2009);
INSERT INTO BOOK VALUES
   (4,'Python','Pearson',2017);
INSERT INTO BOOK VALUES
   (5,'CloudCmpt','OReilly',2014);
```

```
SQL> select *from book;

BOOKID TITLE                PUBLISHER_NAME                  PUB_YEAR
------ -------------------- ------------------------------ ---------
     1 DBMS                 Pearson                        2017
     2 Hadoop               Pearson                        2000
     3 AIML                 TataMcGraw                     2009
     4 Python               Pearson                        2017
     5 CloudCmpt            OReilly                        2014
```

```
CREATE TABLE LIBRARY_PROGRAMME(
    programme_id INT PRIMARY KEY,
    programme_name VARCHAR(10),
    address VARCHAR(20));
```

```
INSERT INTO LIBRARY_PROGRAMME VALUES
  (1,'CSE','Mangalore');
INSERT INTO LIBRARY_PROGRAMME VALUES
  (2,'EC','Mangalore');
INSERT INTO LIBRARY_PROGRAMME VALUES
  (3,'EE','Bangalore');
INSERT INTO LIBRARY_PROGRAMME VALUES
  (4,'IT','Chennai');
INSERT INTO LIBRARY_PROGRAMME VALUES
  (5,'IP','Chennai');
```

```
SQL> select *from library_programme
  2  ;

PROGRAMME_ID PROGRAMME_ ADDRESS
------------ ---------- --------------------
           1 CSE        Mangalore
           2 EC         Mangalore
           3 EE         Bangalore
           4 IT         Chennai
           5 IP         Chennai
```

```
CREATE TABLE BOOK_COPIES(
    book_id NUMBER(2)
    REFERENCES book
    ON DELETE CASCADE,
    programme_id NUMBER(2)
    REFERENCES
    library_programme(programme_id)
    ON DELETE CASCADE,
    no_of_copies NUMBER(2),
    PRIMARY KEY(book_id, programme_id));
```

INSERT INTO **BOOK_COPIES** VALUES (1,1,99);

INSERT INTO **BOOK_COPIES** VALUES (2,1,99);

INSERT INTO **BOOK_COPIES** VALUES (3,2,99);

INSERT INTO **BOOK_COPIES** VALUES (3,1,99);

```
SQL> select *from book_copies
  2  ;
```

| BOOK_ID | PROGRAMME_ID | NO_OF_COPIES |
|---|---|---|
| 1 | 1 | 99 |
| 2 | 1 | 99 |
| 3 | 2 | 99 |
| 3 | 1 | 99 |

```
CREATE TABLE  BOOK_LENDING(
   book_id NUMBER(5),
   programme_id NUMBER(3),
   card_no NUMBER(3),
   date_out DATE,
   due_date DATE,
   FOREIGN KEY(book_id,programme_id)
  REFERENCES
   book_copies ON DELETE CASCADE,
   PRIMARY KEY(programme_id, book_id,
  card_no),
   CONSTRAINT CK1 CHECK (due_date >
  date_out) );
```

- INSERT INTO **BOOK_LENDING** VALUES (1,1,1,'02-JAN-17','09-JAN-17');
- INSERT INTO **BOOK_LENDING** VALUES (1,1,2,'02-MAR-17','09-MAR-17');
- INSERT INTO **BOOK_LENDING** VALUES (2,1,2,'02-MAR-17','09-MAR-17');
- INSERT INTO **BOOK_LENDING** VALUES (1,1,3,'04-APR-17','30-JUN-17');

```
SQL> select *from book_lending;

   BOOK_ID PROGRAMME_ID    CARD_NO DATE_OUT  DUE_DATE
---------- ------------ ---------- --------- ---------
         1            1          1 02-JAN-17 09-JAN-17
         1            1          2 02-MAR-17 09-MAR-17
         2            1          2 02-MAR-17 09-MAR-17
         1            1          3 04-APR-17 30-JUN-17
```

```
CREATE  TABLE  BOOK_AUTHORS(
    bookid INT REFERENCES book
    ON DELETE CASCADE,
    authorname VARCHAR2(20),
    PRIMARY KEY(bookid, authorname));
```

- INSERT INTO **BOOK_AUTHORS** VALUES (1, 'Elmarsi');
- INSERT INTO **BOOK_AUTHORS** VALUES (1, 'Navathe');
- INSERT INTO **BOOK_AUTHORS** VALUES (2, 'Douglas');
- INSERT INTO **BOOK_AUTHORS** VALUES (3, 'Elaine');
- INSERT INTO **BOOK_AUTHORS** VALUES (5, 'Srinivasan');

```
SQL> select *from book_authors;

    BOOKID AUTHORNAME
---------- --------------------
         1 Elmarsi
         1 Navathe
         2 Douglas
         3 Elaine
         5 Srinivasan
```

```
SQL> select *from publisher;

NAME                 ADDRESS              PHONE
-------------------- -------------------- ----------
Pearson              London               9874522224
TataMcGraw           NewYork              9858523565
Oxford               UK                   9885121112
Cambridge            UK                   9785634615
OReilly              California           9994125455
```

```
SQL> select *from book_copies
  2  ;

   BOOK_ID PROGRAMME_ID NO_OF_COPIES
---------- ------------ ------------
         1            1           99
         2            1           99
         3            2           99
         3            1           99
```

```
SQL> select *from book;

BOOKID TITLE                PUBLISHER_NAME       PUB_YEAR
------ -------------------- -- -------------------- ---------

     1 DBMS                   Pearson              2017
     2 Hadoop                 Pearson              2000
     3 AIML                   TataMcGraw           2009
     4 Python                 Pearson              2017
     5 CloudCmpt              OReilly              2014
```

```
SQL> select *from book_lending;

   BOOK_ID PROGRAMME_ID    CARD_NO DATE_OUT  DUE_DATE
---------- ------------ ---------- --------- ---------
         1            1          1 02-JAN-17 09-JAN-17
         1            1          2 02-MAR-17 09-MAR-17
         2            1          2 02-MAR-17 09-MAR-17
         1            1          3 04-APR-17 30-JUN-17
```

```
SQL> select *from library_programme
  2  ;

PROGRAMME_ID PROGRAMME_ ADDRESS
------------ ---------- ----------
           1 CSE        Mangalore
           2 EC         Mangalore
           3 EE         Bangalore
           4 IT         Chennai
           5 IP         Chennai
```

```
SQL> select *from book_authors;

    BOOKID AUTHORNAME
---------- --------------------
         1 Elmarsi
         1 Navathe
         2 Douglas
         3 Elaine
         5 Srinivasan
```

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

```
SELECT b.bookid, lp.programme_id, title, publisher_name,
no_of_copies, authorname
FROM book b, book_authors a, book_copies bc,
library_programme lp
WHERE b.bookid = a.bookid AND b.bookid = bc.book_id
AND  bc.programme_id = lp.programme_id;
```

**OR**

```
SELECT b.bookid, title, publisher_name,authorname,
no_of_copies
FROM  book b, book_authors a, book_copies bc
WHERE b.bookid = bc.book_id AND b.bookid = a.bookid;
```

| BOOKID | PROGRAMME_ID | TITLE | PUBLISHER_NAME | NO_OF_COPIES | AUTHORNAME |
|---|---|---|---|---|---|
| 1 | 1 | DBMS | Pearson | 99 | Elmarsi |
| 1 | 1 | DBMS | Pearson | 99 | Navathe |
| 2 | 1 | Hadoop | Pearson | 99 | Douglas |
| 3 | 1 | AIML | TataMcGraw | 99 | Elaine |
| 3 | 2 | AIML | TataMcGraw | 99 | Elaine |

- INSERT INTO **BOOK_LENDING** VALUES (3,1,2,'05-JAN-17',’10-FEB-17');

```
SQL> SELECT * FROM BOOK_LENDING;

   BOOK_ID PROGRAMME_ID    CARD_NO DATE_OUT  DUE_DATE
---------- ------------ ---------- --------- ---------
         1            1          1 02-JAN-17 09-JAN-17
         1            1          2 02-MAR-17 09-MAR-17
         2            1          2 02-MAR-17 09-MAR-17
         1            1          3 04-APR-17 30-JUN-17
         3            1          2 05-JAN-17 10-FEB-17
```

**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 '01-JAN-2017' AND '30-JUN-2017'

GROUP BY card_no

HAVING COUNT(*) >= 3 ;
```

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

DELETE FROM book WHERE Bookid = &bid;

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

```
CREATE TABLE BOOK1 (
   book_id number PRIMARY KEY,
   title VARCHAR2(10),
   publisher_name VARCHAR2(20) REFERENCES
   publisher ON DELETE CASCADE,
   pub_year number(4))
   PARTITION BY RANGE(pub_year)
    (PARTITION p1 VALUES LESS THAN(2001),
     PARTITION p2 VALUES LESS THAN(2005),
     PARTITION P3 VALUES LESS THAN (2010),
     PARTITION P4 VALUES LESS
   THAN(MAXVALUE));

SELECT * FROM BOOK1 PARTITION (p1);
```

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

```
CREATE OR REPLACE VIEW available_book AS
  SELECT b.bookid, b.title, sum(bc.no_of_copies) -
  (SELECT  count(*)
  FROM book_lending  bl
  WHERE bl.book_id = b. bookid
  GROUP BY bl.book_id) AS books_available
  FROM book b, book_copies  bc
  WHERE b.bookid = bc.book_id
  GROUP BY b.bookid, b.title;

SELECT *FROM available_book;
```