```
#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, Branch_id, No-of_Copies)
#BOOK_LENDING(Book_id, Branch_id, Card_No, Date_Out, Due_Date)
#LIBRARY_BRANCH(Branch_id, Branch_Name, Address)



create database Library;
use Library;
CREATE TABLE PUBLISHER(
 NAME VARCHAR(18) PRIMARY KEY,
 ADDRESS VARCHAR(10),
 PHONE VARCHAR(10));

CREATE TABLE BOOK(
 BOOK_ID INTEGER PRIMARY KEY,
 TITLE VARCHAR(20),
 PUBLISHER_NAME VARCHAR(20)REFERENCES PUBLISHER(NAME)ON DELETE cascade,
 PUB_YEAR year(4));

  CREATE TABLE BOOK_AUTHORS(
 BOOK_ID INTEGER REFERENCES BOOK(BOOK_ID) ON DELETE CASCADE,
 AUTHOR_NAME VARCHAR(20),
 PRIMARY KEY(BOOK_ID));

 CREATE TABLE LIBRARY_BRANCH(
 BRANCH_ID INTEGER PRIMARY KEY,
 BRANCH_NAME VARCHAR(18),
 ADDRESS VARCHAR(15));

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

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

 INSERT INTO PUBLISHER VALUES('PEARSON','BANGALORE','9875462530');
 INSERT INTO PUBLISHER VALUES('MCGRAW','NEWDELHI','7845691234');
INSERT INTO PUBLISHER VALUES('SAPNA','BANGALORE','7845963210');

 select * from publisher;

  INSERT INTO BOOK VALUES(1111,'SE','PEARSON',2005);
 INSERT INTO BOOK VALUES(2222,'DBMS','MCGRAW',2004);
 INSERT INTO BOOK VALUES(3333,'ANOTOMY','PEARSON',2010);
INSERT INTO BOOK VALUES(4444,'ENCYCLOPEDIA','SAPNA',2010);

 select * from book;

INSERT INTO BOOK_AUTHORS VALUES(1111,'SOMMERVILLE');
INSERT INTO BOOK_AUTHORS VALUES(2222,'NAVATHE');
INSERT INTO BOOK_AUTHORS VALUES(3333,'HENRY GRAY');
INSERT INTO BOOK_AUTHORS VALUES(4444,'THOMAS');

INSERT INTO LIBRARY_BRANCH VALUES(11,'CENTRAL TECHNICAL','MG ROAD');
INSERT INTO LIBRARY_BRANCH VALUES(22,'MEDICAL','BH ROAD');
INSERT INTO LIBRARY_BRANCH VALUES(33,'CHILDREN','SS PURAM');
 INSERT INTO LIBRARY_BRANCH VALUES(44,'SECRETARIAT','SIRAGATE');
 INSERT INTO LIBRARY_BRANCH VALUES(55,'GENERAL','JAYANAGAR');

INSERT INTO BOOK_COPIES VALUES(1111,11,5);
INSERT INTO BOOK_COPIES VALUES(3333,22,6);
 INSERT INTO BOOK_COPIES VALUES(4444,33,10);
INSERT INTO BOOK_COPIES VALUES(2222,11,12);
 INSERT INTO BOOK_COPIES VALUES(4444,55,3);

 INSERT INTO BOOK_LENDING VALUES(2222,11,1,'2017-01-10','2017-08-20');
INSERT INTO BOOK_LENDING VALUES(3333,22,2,'2017-07-06','2017-08-12');
 INSERT INTO BOOK_LENDING VALUES(4444,55,1,'2017-4-11-','2017-08-09');
 INSERT INTO BOOK_LENDING VALUES(2222,11,5,'2017-08-09','2017-08-19');
INSERT INTO BOOK_LENDING VALUES(4444,33,1,'2017-06-10','2017-08-15');
INSERT INTO BOOK_LENDING VALUES(1111,11,1,'2017-05-12','2017-06-10');
INSERT INTO BOOK_LENDING VALUES(3333,22,1,'2017-07-10','2017-07-15');
```

#1 Retrieve details of all books in the library â€“ id, title, name of publisher, authors, number of copies in each branch, etc.

```
SELECT LB.BRANCH_NAME, B.BOOK_ID,TITLE, PUBLISHER_NAME,AUTHOR_NAME,
NO_OF_COPIES
 FROM BOOK B, BOOK_AUTHORS BA, BOOK_COPIES BC, LIBRARY_BRANCH LB
 WHERE B.BOOK_ID = BA.BOOK_ID AND
 BA.BOOK_ID = BC.BOOK_ID AND
 BC.BRANCH_ID = LB.BRANCH_ID
 GROUP BY LB.BRANCH_NAME, B.BOOK_ID, TITLE, PUBLISHER_NAME,AUTHOR_NAME,
NO_OF_COPIES;
```

#2Get 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 '2017-01-01' AND '2017-06-30'
 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

```
select *from book;
delete from book
where book_id='3333';
select * from book;
```

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

```
SELECT BOOK_ID, TITLE, PUBLISHER_NAME, PUB_YEAR

 FROM BOOK

 GROUP BY PUB_YEAR, BOOK_ID, TITLE, PUBLISHER_NAME;
```

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

```
CREATE VIEW BOOKS_AVAILABLE AS
 SELECT b.BOOK_ID, b.TITLE, c.NO_OF_COPIES
 FROM LIBRARY_BRANCH l , BOOK b , BOOK_COPIES c
 WHERE B.BOOK_ID = C.BOOK_ID AND
 L.BRANCH_ID=C.BRANCH_ID;
 select * from books_available
```

#2 Consider the following schema for Order Database: SALESMAN(Salesman_id, Name, City, Commission) CUSTOMER(Customer_id, Cust_Name, City, Grade, Salesman_id) ORDERS(Ord_No,Purchase_Amt, Ord_Date, Customer_id, Salesman_id) Write SQL queries to

#1. Count the customers with grades above Bangalore's average.
#2. Find the name and numbers of all salesman who had more than one customer.
#3. List all the salesman and indicate those who have and donâ€Ÿt have customers in their cities (Use UNION operation.)
#4. Create a view that finds the salesman who has the customer with the highest order of a day.
#5. Demonstrate the DELETE operation by removing salesman with id 1000. All his orders must also be deleted

```
CREATE DATABASE ORDER_db;
USE ORDER_db;

CREATE TABLE SALESMAN(
SALESMAN_ID int PRIMARY KEY,Salesman_name VARCHAR(20) not null,CITY varchar(20)
not null,COMMISSION INTEGER );

CREATE TABLE CUSTOMER (CUSTOMER_ID int PRIMARY KEY,CUSTOMER_NAME
VARCHAR(20)not null,CITY VARCHAR(20)not null,GRADE INT not null,
SALESMAN_ID integer references salesman(salesman_id) on delete set null
 );

CREATE TABLE ORDERS (
ORD_NO INT PRIMARY KEY,Purchase_Amt integer not null,ORDER_DATE DATE not
null,CUSTOMER_ID INTEGER references customer(customer_id) on delete cascade
,SALESMAN_ID INTEGER references salesman(salesman_id) on delete cascade);

INSERT INTO SALESMAN VALUES (1000,"RAJ","BENGALURU",50);
INSERT INTO SALESMAN VALUES (2000,"ASHWIN","TUMKUR",30);
INSERT INTO SALESMAN VALUES (3000,"BINDU","MUMBAI",40);
INSERT INTO SALESMAN VALUES (4000,"LAVANYA","BENGALURU",40);
INSERT INTO SALESMAN VALUES (5000,"ROHIT","MYSORE",60);
INSERT INTO CUSTOMER VALUES(11,"INFOSYS","BENGALURU",5,1000);
INSERT INTO CUSTOMER VALUES(22,"TCS","BENGALURU",4,2000);
INSERT INTO CUSTOMER VALUES(33,"WIPRO","MYSORE",7,1000);
INSERT INTO CUSTOMER VALUES(44,"TCS","MYSORE",6,2000);
INSERT INTO CUSTOMER VALUES(55,"ORACLE","TUMKUR",3,3000);



INSERT INTO ORDERS VALUES(1,200000,'2021-06-12',11,1000);
INSERT INTO ORDERS VALUES(2,300000,'2021-05-15',11,2000);
INSERT INTO ORDERS VALUES(3,400000,'2021-09-18',22,1000);

SELECT * FROM ORDERS;


#1
SELECT COUNT(CUSTOMER_ID)
FROM CUSTOMER
WHERE GRADE>(SELECT AVG(GRADE)
FROM CUSTOMER
WHERE CITY LIKE 'BENGALURU');


#2

SELECT salesman_name, COUNT(CUSTOMER_ID)
FROM SALESMAN S, CUSTOMER C
WHERE S.SALESMAN_ID=C.SALESMAN_ID
GROUP BY customer_NAME
HAVING COUNT(CUSTOMER_ID)>1;


#3
(SELECT salesman_name
FROM SALESMAN S, CUSTOMER C
WHERE S.SALESMAN_ID=C.SALESMAN_ID AND
S.CITY=C.CITY)
UNION
(SELECT SALESMAN_name
FROM SALESMAN
WHERE SALESMAN_ID NOT IN(SELECT S1.SALESMAN_ID
FROM SALESMAN S1, CUSTOMER C1
WHERE S1.SALESMAN_ID=C1.SALESMAN_ID AND
S1.CITY=C1.CITY));
```

#4

```
CREATE VIEW SALES_HIGHERODER AS
SELECT SALESMAN_ID, PURCHASE_AMT
FROM ORDERS
WHERE PURCHASE_AMT=(SELECT MAX(PURCHASE_AMT)
FROM ORDERS

WHERE ORDER_DATE='2021-09-18');

SELECT * FROM SALES_HIGHERODER;
```

#5

```
select *from salesman;
DELETE from salesman
WHERE salesman_id = 1000;
select * from salesman
```