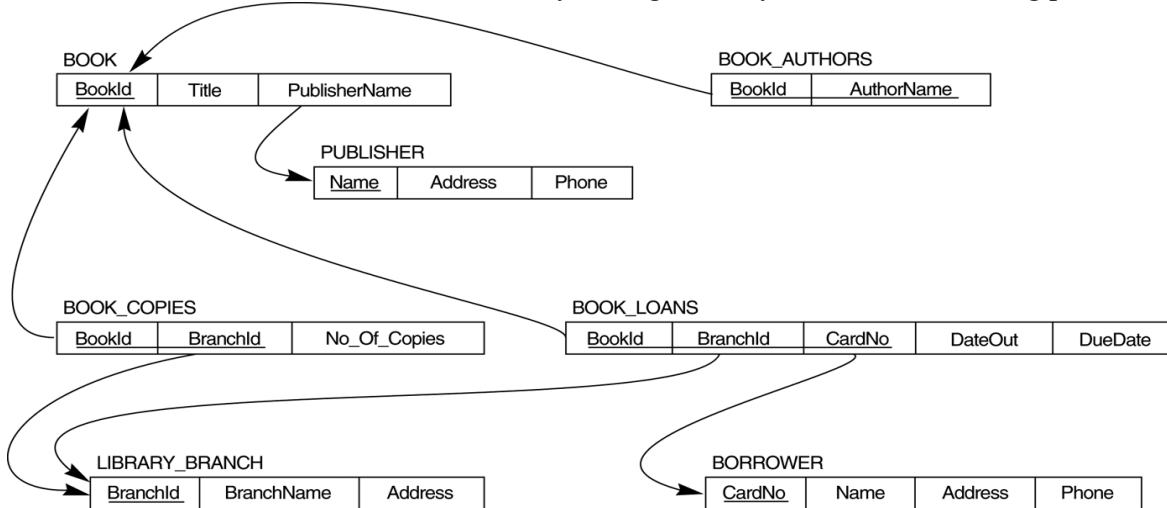


SQL EXERCISE

Practice and practice we can improve our skills.

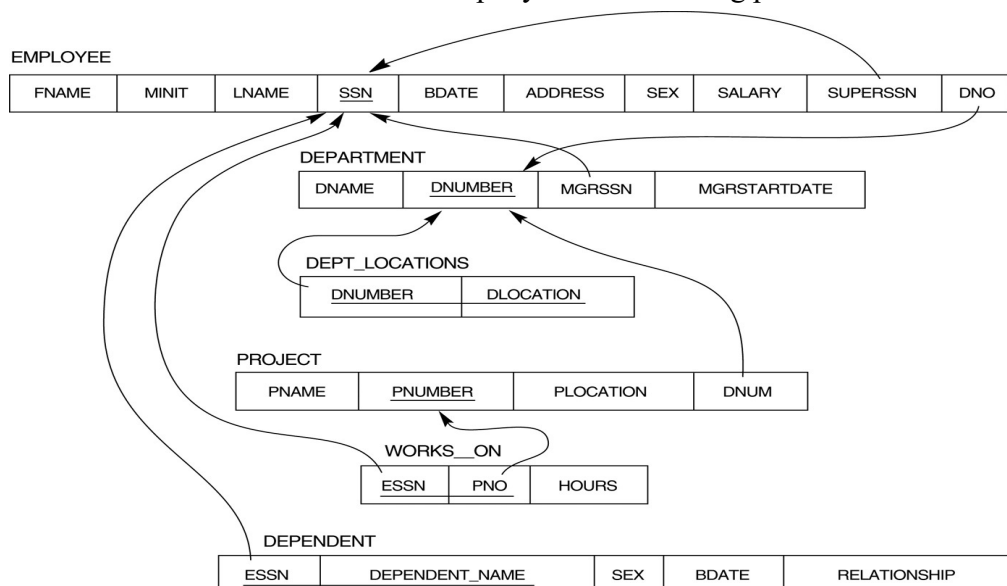
Retrieve the required information using SQL language.

Part I. Give a database schema for a library management system as the following picture.



1. How many copies of the book titled The Lost Tribe are owned by the library branch whose name is "Sharpstown"?
2. How many copies of the book titled The Lost Tribe are owned by each library branch?
3. Retrieve the names of all borrowers who do not have any books checked out .
4. For each book that is loaned out from the "Sharpstown" branch and whose DueDate is today, retrieve the book title, the borrower's name, and the borrower's address.
5. For each library branch, retrieve the branch name and the total number of books loaned out from that branch.
6. Retrieve the names, addresses, and number of books checked out for all borrowers who have more than five books checked out.
7. For each book authored (or co-authored) by "Stephen King", retrieve the title and the number of copies owned by the library branch whose name is "Central"

Part II Give a database schema of a company as the following picture.



8. Retrieve the names of employees in department 5 who work more than 10 hours per week on the 'ProductX' project.
9. For each project, list the project name and the total hours per week (by all employees) spent on that project.
10. Retrieve the names of employees who work on every project.
11. Retrieve the names of employees who do not work on any project.
12. Find the names and addresses of employees who work on at least one project located in Houston but whose department has no location in Houston.
13. List the last names of department managers who have no dependents.
14. Find details of those employees whose salary is > the average salary for all employees. Output salary in descending order.
15. Find details of those employees whose salary is > the average salary for all employees in his/her department. Output salary in ascending order.

Answer:

- *Should not refer to the answer before trying to write down your solutions*
- *The answer for each question is only one (or two) of the many other solutions*

1.

Solution 1:

```
SELECT bc.No_Of_Copies
FROM BOOK b, BOOK_COPIES bc, LIBRARY_BRANCH bl
WHERE      b.BookId = bc.BookId AND
           bc.BranchId = bl.BranchId AND
           Title='The Lost Tribe' AND BranchName='Sharpstown';
```

Solution 2:

```
SELECT No_Of_Copies
FROM ((BOOK NATURAL JOIN BOOK_COPIES ) NATURAL JOIN
LIBRARY_BRANCH )
WHERE Title='The Lost Tribe' AND BranchName='Sharpstown';
```

2. **SELECT BranchName, No_Of_Copies**
FROM ((BOOK NATURAL JOIN BOOK_COPIES) NATURAL JOIN
LIBRARY_BRANCH)
WHERE Title='The Lost Tribe';

3.

Solution 1:

```
SELECT      Name
FROM        BORROWER B
WHERE       CardNo NOT IN (SELECT CardNo
                           FROM BOOK_LOANS );
```

Solution 2:

```
SELECT      Name
FROM        BORROWER B
WHERE       NOT EXISTS (SELECT *
                       FROM BOOK_LOANS L
                       WHERE B.CardNo = L.CardNo );
```

4. **SELECT B.Title, R.Name, R.Address
FROM BOOK B, BORROWER R, BOOK_LOANS BL, LIBRARY_BRANCH LB
WHERE LB.BranchName='Sharpstown' AND LB.BranchId=BL.BranchId AND
BL.DueDate='today' AND BL.CardNo=R.CardNo AND BL.BookId=B.BookId**

5. **SELECT L.BranchName, COUNT(*)
FROM LIBRARY_BRANCH L, BOOK_LOANS BL
WHERE BL.BranchId = L.BranchId
GROUP BY L.BranchName;**

6. **SELECT B.Name, B.Address, COUNT(*)
FROM BORROWER B, BOOK_LOANS L
WHERE B.CardNo = L.CardNo
GROUP BY B.CardNo, B.Name, B.Address
HAVING COUNT(*) > 5;**

7.

Solution 1:

**SELECT Title, No_Of_Copies
FROM (((BOOK_AUTHORS NATURAL JOIN BOOK) NATURAL JOIN BOOK_COPIES)
NATURAL JOIN LIBRARY_BRANCH)WHERE Author_Name='Stephen King' AND
BranchName='Central';**

Solution 2: Student should write the another solution not using the natural join.

8. **SELECT LNAME, FNAME
FROM EMPLOYEE, WORKS_ON, PROJECT
WHERE DNO=5 AND SSN=ESSN AND PNO=PNUMBER AND PNAME='ProductX'
AND HOURS>10;**

9. **SELECT PNAME, SUM (HOURS)
FROM PROJECT, WORKS_ON
WHERE PNUMBER=PNO
GROUP BY PNAME;**

Note: The Group By clause should be replaced as GROUP BY (PNUMBER, PNAME) since there may be some projects have the same name.

10.

Solution 1:

**SELECT E.LNAME, E.FNAME
FROM EMPLOYEE E
WHERE NOT EXISTS (SELECT PNUMBER
FROM PROJECT
WHERE PNUMBER NOT IN (SELECT PNO
FROM WORKS_ON
WHERE ESSN=E.SSN));**

Solution 2:

```
SELECT      LNAME, FNAME
FROM        EMPLOYEE
WHERE       NOT EXISTS (SELECT PNUMBER
                        FROM      PROJECT
                        WHERE     NOT EXISTS (SELECT *
                                           FROM WORKS_ON
                                           WHERE PNUMBER=PNO AND ESSN=SSN ) );
```

11.

Solution 1:

```
SELECT      LNAME, FNAME
FROM        EMPLOYEE
WHERE       SSN NOT IN ( SELECT ESSN
                        FROM WORKS_ON);
```

Solution 2:

```
SELECT      LNAME, FNAME
FROM        EMPLOYEE
WHERE       NOT EXISTS ( SELECT *
                        FROM WORKS_ON
                        WHERE ESSN=SSN );
```

12.

Solution 1:

```
SELECT      LNAME, FNAME, ADDRESS
FROM        EMPLOYEE
WHERE       EXISTS ( SELECT *
                    FROM WORKS_ON W, PROJECT P, DEPT_LOCATIONS DL
                    WHERE     W.PNO = P.PNUMBER AND
                             P.DNUM = DL.DNUM  AND
                             DL.DLOCATION <> 'Houston');
```

Solution 2:

```
SELECT      LNAME, FNAME, ADDRESS
FROM        EMPLOYEE
WHERE       EXISTS ( SELECT *
                    FROM WORKS_ON, PROJECT
                    WHERE SSN=ESSN AND PNO=PNUMBER AND PLOCATION='Houston' )
AND
NOT EXISTS ( SELECT *
            FROM DEPT_LOCATIONS
            WHERE DNO=DNUMBER AND DLOCATION='Houston' );
```

13.

Solution 1:

```
SELECT      E.LNAME, E.FNAME
FROM        EMPLOYEE E, DEPARTMENT D
WHERE       E.SSN = D.MRGSSN AND
            NOT EXISTS ( SELECT DEPENDENT_NAME
                        FROM DEPENDENT
                        WHERE ESSN=E.SSN )
```

Solution 2:

```
SELECT      LNAME, FNAME
FROM        EMPLOYEE
WHERE       EXISTS ( SELECT *
                     FROM DEPARTMENT
                     WHERE SSN=MGRSSN )
AND
NOT EXISTS ( SELECT *
             FROM DEPENDENT
             WHERE SSN=ESSN );
```

```
14. SELECT      *
FROM            Employee
WHERE           Salary > (SELECT AVG (Salary)
                        FROM Employee )
ORDER BY Salary DESC;
```

15.

Solution 1:

```
SELECT      E.*
FROM        EMPLOYEE E, (SELECT DNO, AVERAGE(SALARY) AS LTB
                        FROM EMPLOYEE
                        GROUP BY DNO) AS A
WHERE       E.DNO = A.DNO AND
           E.SALARY > LTB;
```

Solution 2:

```
SELECT      E.*
FROM        EMPLOYEE E
WHERE       E.SALARY > (SELECT AVERAGE(SALARY)
                      FROM EMPLOYEE
                      WHERE DNO = E.DNO);
```

Note: if we want to display the average of the salary we should employ the solution 1 and put the “field name” LTB in the select clause.