Q1: Specify the following queries on the database of Figure (1) bellow in SQL:

a. How many copies of the book titled The Lost Tribe are owned by the library branch whose name is 'Sharpstown'?

SELECT NoOfCopies
FROM ( (BOOK NATURAL JOIN BOOK_COPIES ) NATURAL JOIN LIBRARY_BRANCH )
WHERE Title='The Lost Tribe' AND BranchName='Sharpstown'
b. How many copies of the book titled The Lost Tribe are owned by each library branch?

SELECT BranchName, NoOfCopies
FROM ( (BOOK NATURAL JOIN BOOK_COPIES ) NATURAL JOIN LIBRARY_BRANCH )
WHERE Title='The Lost Tribe'

c. Retrieve the names of all borrowers who do not have any books checked out.

SELECT Name
FROM BORROWER B
WHERE NOT EXIST ( SELECT *
FROM BOOK_LOANS L
WHERE B.CardNo = L.CardNo ).

d. 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.

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

e. For each library branch, retrieve the branch name and the total number of books loaned out from that branch.

SELECT L.BranchName, COUNT(*)
FROM BOOK_LOANS B, LIBRARY_BRANCH L
WHERE B.BranchId = L.BranchId
GROUP BY L.BranchName
f. Retrieve the names, addresses, and number of books checked out for all borrowers who have more than five books checked out.

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

g. For each book authored (or coauthored) by 'Stephen King,' retrieve the title and the number of copies owned by the library branch whose name is 'Central.'

```
SELECT Title, NoOfCopies
FROM ((BOOK_AUTHORS NATURAL JOIN BOOK)
NATURAL JOIN BOOK_COPIES)
NATURAL JOIN LIBRARY_BRANCH)
WHERE Author_Name = 'Stephen King' and BranchName = 'Central'
```
Q2: Specify the following queries on the database of Figure (2) below in SQL:

<table>
<thead>
<tr>
<th>STUDENT</th>
<th>Name</th>
<th>StudentNumber</th>
<th>Class</th>
<th>Major</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smith</td>
<td>17</td>
<td>1</td>
<td>CS</td>
<td></td>
</tr>
<tr>
<td>Brown</td>
<td>8</td>
<td>2</td>
<td>CS</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>COURSE</th>
<th>CourseName</th>
<th>CourseNumber</th>
<th>CreditHours</th>
<th>Department</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intro to Computer Science</td>
<td>CS1310</td>
<td>4</td>
<td>CS</td>
<td></td>
</tr>
<tr>
<td>Data Structures</td>
<td>CS3320</td>
<td>4</td>
<td>CS</td>
<td></td>
</tr>
<tr>
<td>Discrete Mathematics</td>
<td>MATH2410</td>
<td>3</td>
<td>MATH</td>
<td></td>
</tr>
<tr>
<td>Database</td>
<td>CS3380</td>
<td>3</td>
<td>CS</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SECTION</th>
<th>SectionIdentifier</th>
<th>CourseNumber</th>
<th>Semester</th>
<th>Year</th>
<th>Instructor</th>
</tr>
</thead>
<tbody>
<tr>
<td>85</td>
<td>MATH2410</td>
<td>Fall</td>
<td>98</td>
<td>King</td>
<td></td>
</tr>
<tr>
<td>92</td>
<td>CS1310</td>
<td>Fall</td>
<td>98</td>
<td>Anderson</td>
<td></td>
</tr>
<tr>
<td>102</td>
<td>CS3320</td>
<td>Spring</td>
<td>99</td>
<td>Knuth</td>
<td></td>
</tr>
<tr>
<td>112</td>
<td>MATH2410</td>
<td>Fall</td>
<td>99</td>
<td>Chang</td>
<td></td>
</tr>
<tr>
<td>119</td>
<td>CS1310</td>
<td>Fall</td>
<td>99</td>
<td>Anderson</td>
<td></td>
</tr>
<tr>
<td>135</td>
<td>CS3380</td>
<td>Fall</td>
<td>99</td>
<td>Stone</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GRADE_REPORT</th>
<th>StudentNumber</th>
<th>SectionIdentifier</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>17</td>
<td>112</td>
<td></td>
<td>B</td>
</tr>
<tr>
<td>17</td>
<td>119</td>
<td></td>
<td>C</td>
</tr>
<tr>
<td>8</td>
<td>85</td>
<td></td>
<td>A</td>
</tr>
<tr>
<td>8</td>
<td>92</td>
<td></td>
<td>A</td>
</tr>
<tr>
<td>8</td>
<td>102</td>
<td></td>
<td>B</td>
</tr>
<tr>
<td>8</td>
<td>135</td>
<td></td>
<td>A</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PREREQUISITE</th>
<th>CourseNumber</th>
<th>PrerequisiteNumber</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS3380</td>
<td>CS3320</td>
<td></td>
</tr>
<tr>
<td>CS3380</td>
<td>MATH2410</td>
<td></td>
</tr>
<tr>
<td>CS3320</td>
<td>CS1310</td>
<td></td>
</tr>
</tbody>
</table>
a. Retrieve the names of all senior students majoring in 'COSC' (computer science).

```
SELECT Name
FROM STUDENT
WHERE Major='COSC'
```

b. Retrieve the names of all courses taught by professor King in 85 and 86.

```
SELECT CourseName
FROM COURSE, SECTION
WHERE COURSE.CourseNumber=SECTION.CourseNumber AND Instructor='King'
AND (Year='85' OR Year='86')

Another possible SQL query uses nesting as follows:

```
SELECT CourseName
FROM COURSE
WHERE CourseNumber IN ( SELECT CourseNumber
FROM SECTION
WHERE Instructor='King' AND (Year='85' OR Year='86') )
```

c. For each section taught by professor King, retrieve the course number, semester, year, and number of students who took the section.

```
SELECT CourseNumber, Semester, Year, COUNT(*)
FROM SECTION, GRADE_REPORT
WHERE Instructor='King' AND SECTION.SectionIdentifier=GRADE_REPORT.SectionIdentifier
GROUP BY CourseNumber, Semester, Year
```
d. Retrieve the name and transcript of each senior student (Class=5) majoring in COSC. Transcript includes course name, course number, credit hours, semester, year, and grade for each course completed by the student.

```
SELECT Name, CourseName, C.CourseNumber, CreditHours, Semester, Year, Grade
FROM STUDENT ST, COURSE C, SECTION S, GRADE_REPORT G
WHERE Class=5 AND Major='COSC' AND ST.StudentNumber=G.StudentNumber AND
G.SectionIdentifier=S.SectionIdentifier AND S.CourseNumber=C.CourseNumber
```

e. Retrieve the names and major departments of all straight A students (students who have a grade of A in all their courses).

```
SELECT Name, Major
FROM STUDENT
WHERE NOT EXISTS ( SELECT * 
FROM GRADE_REPORT
WHERE StudentNumber= STUDENT.StudentNumber AND NOT(Grade='A'))
```

f. Retrieve the names and major departments of all students who do not have any grade of A in any of their courses.

```
SELECT Name, Major
FROM STUDENT
WHERE NOT EXISTS ( SELECT * 
FROM GRADE_REPORT
WHERE StudentNumber= STUDENT.StudentNumber AND Grade='A' )
```