Chapter 7 Review Questions

1. Discuss how NULLs are treated in comparison operators in SQL. How are NULLs treated when aggregate functions are applied in an SQL query? How are NULLs treated if they exist in grouping attributes?
   Unknown/TRUE – AND – FALSE
   Uknown/TRUE – OR – TRUE
   NOT(unknown) – unknown
   Aggregated functions – ignore null value – MAX/MIN/COUNT/SUM/AVG
   Group – NULLs will. Be put in their own group – have their own result row
2. Specify the following queries on the database in Figure 5.5 in SQL.

   a. For each department whose average employee salary is more than $30,000, retrieve the department name and the number of employees working for that department.

   b. Suppose that we want the number of male employees in each department making more than $30,000, rather than all employees. Can we specify this query in SQL? Why or why not?

3. In SQL, specify the following queries on the database in Figure 5.5 using the concept of nested queries and other concepts described in this chapter.
   a. Retrieve the names of all employees who work in the department that has the employee with the highest salary among all employees
   ```
   SELECT FNAME, LNAME FROM EMPLOYEE WHERE DNO=(
   SELECT DNO FROM EMPLOYEE WHERE SALARY =
   (SELECT MAX(SALARY) FROM EMPLOYEE));
   ```

   b. Retrieve the names of employees who make at least $10,000 more than the employee who is paid the least in the company.

   ```
   SELECT LNAME FROM EMPLOYEE WHERE SALARY >=(SELECT
   MIN(SALARY)+ 10000 FROM EMPLOYEE);
   ```
   c. Retrieve the names of all employees whose supervisor's supervisor has '888665555' for Ssn.

   ```
   SELECT LNAME FROM EMPLOYEE WHERE SUPERSSN IN
   (SELECT SSN FROM EMPLOYEE WHERE SUPERSSN =
   ‘888665555’);
   ```

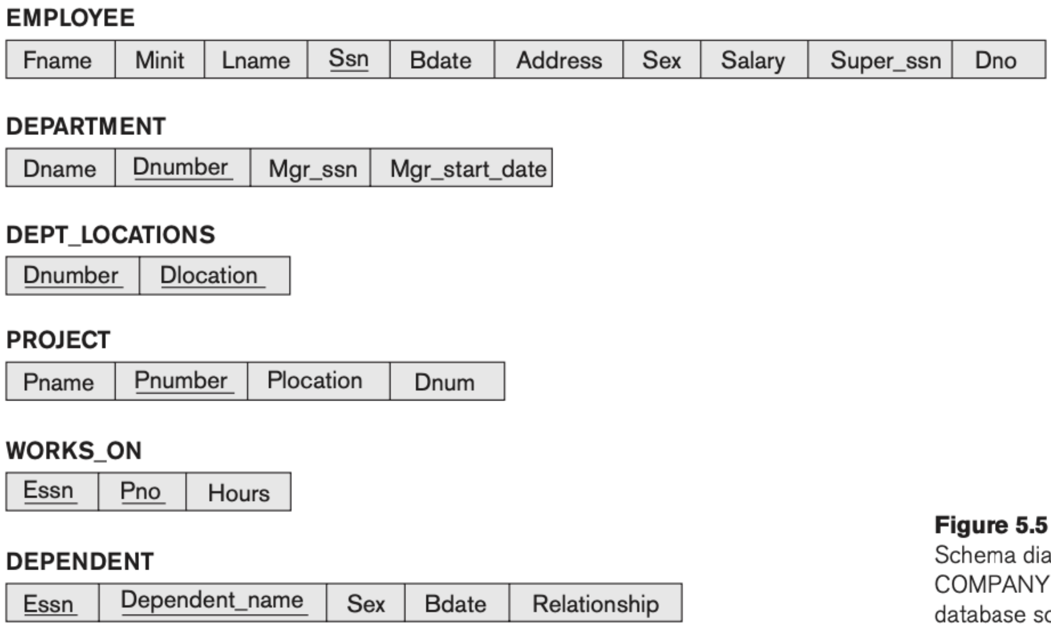


Figure 5.5
Schema diagram for the COMPANY relational database schema.

4. Specify the following queries in SQL on the database schema in Figure 1.2.
   a. 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 AS S
WHERE NOT EXISTS
(SELECT * FROM GRADE_REPORT WHERE STUDENT_NUMBER =
S.STUDENT_NUMBER AND NOT(GRADE = 'A')) ;
```

   ANY STUDENT THAT HAS AN A GRADE IN A COURSE:

```
SELECT S.NAME, S.MAJOR FROM STUDENT AS S JOIN
GRADE_REPORT AS G ON STUDENT_NUM WHERE G.GRADE ='A';
----
SELECT S.NAME, S.MAJOR FROM STUDENT AS S, GRADE_REPORT
AS G WHERE G.STUDENT_NUM = S.STUDENT_NUMBER AND
G.GRADE= 'A';
```

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

```
SELECT NAME, MAJOR FROM STUDET WHERE NOT EXIST (SELECT
* FROM GRADE_REPORT WHERE STUDENT_NUM =
STUDENT.STUDENT_NUMBER AND GRADE ='A');
```

STUDENT

Name	Student_number	Class	Major
Smith	17	1	CS
Brown	8	2	CS

COURSE

Course_name	Course_number	Credit_hours	Department
Intro to Computer Science	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Database	CS3380	3	CS

SECTION

Section_identifier	Course_number	Semester	Year	Instructor
85	MATH2410	Fall	07	King
92	CS1310	Fall	07	Anderson
102	CS3320	Spring	08	Knuth
112	MATH2410	Fall	08	Chang
119	CS1310	Fall	08	Anderson
135	CS3380	Fall	08	Stone

GRADE_REPORT

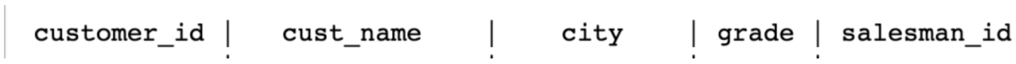
Student_number	Section_identifier	Grade
17	112	B
17	119	C
8	85	A
8	92	A
8	102	B
8	135	A

PREREQUISITE

Course_number	Prerequisite_number
CS3380	CS3320
CS3380	MATH2410
CS3320	CS1310

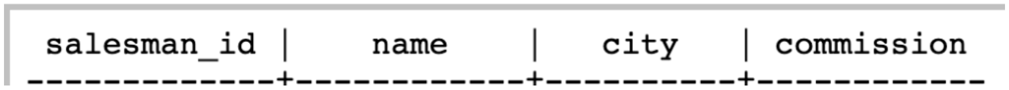
Figure 1.2
A database that stores student and course information.

customer



```
 customer_id |   cust_name    |    city    | grade | salesman_id
```

salesman



```
 salesman_id |    name    |   city    | commission
-------------+------------+-----------+------------
```

Write a SQL statement to know which salesman are working for which customer.

```
SELECT C.CUSTOMER_NAME, S.NAME, C.CITY
FROM CUSTOMER AS C LEFT JOIN SALESMAN AS S
ON C.SALESMAN_ID = S.SALESMAN_ID
```

LEFT JOIN
LIST OF ALL CUSTOMERS
FOR CUSTOMERS THAT DON’T HAVE A SID WILL HAVE NULL FOR S.NAME

RIGHT JOIN
LIST ALL THE SALESMAN NAMES BUT HAVE NULL CUSTOMER_NAME IF THE SALESMAN HAS NO CUSTOMERS

C_ID	S_ID
1	1
2	1
3	2
4	3

S_ID
1
2
3
4
5

S_NAME	C-NAME
1	1
1	2
2	3
3	4
4	NULL
5	NULL