\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
**DBS201 Assignment 1 - (40 Marks)**  
   
Due Friday November 04 in class. Late penalty is 20% per day.   
  
Your group must consist of 3 or 4 people. Submissions done by a single person will receive a 20% penalty.   
Print out submission is required, must include clear screenshots mainly for sql statements.   
  
You must hand in the Student Assignment Submission Form (below) with your assignment (one per group).  
Student Assignment Submission Form  
  
I/we declare that the attached assignment is my/our own work in accordance with the Seneca Academic Policy.   
No part of this assignment has been copied manually or electronically from any other source (including web sites) or distributed to other students  
.  
 Name(s) Student ID(s) Signature  
  
1 Adriel Arce De La Cruz 106336167  
  
2 Saif Husain Khan 125444158  
  
3 Alejandro Mesa Suarez 038515151  
  
ZEUS ACCOUNT and Password: DA201A35 AB112358

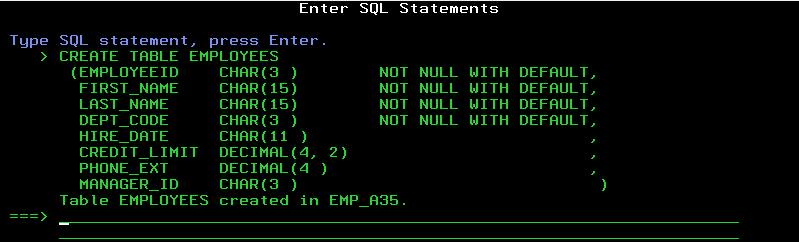
**\*\*\* SCHEMA NAME IS EMP\_A35 \*\*\***

**\*\*\* TABLE NAMES ARE EMPLOYEES AND LOCATIONS \*\*\***

**\*\*\* HAD TO CREATE A DIFFERENT TABLE BECAUSE DEPARTMENTS WAS ALREADY CREATED BY THE ADMIN \*\*\***

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
 ::ASSIGNMENT #1:: PART A  
  
Total Marks : 20  
  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
Create a collection with name as EMP\_XXX. XXX are the last three characters from your userid.   
Inside the collection create the following Employee table. Insert the data inside it.  
>> Note: Employees is the work tables for these queries

Table: Employees (EmployeeID, First\_Name, Last\_Name, Dept\_Code, Hire\_Date, Credit\_Limit, Phone\_Ext, Manager\_id)

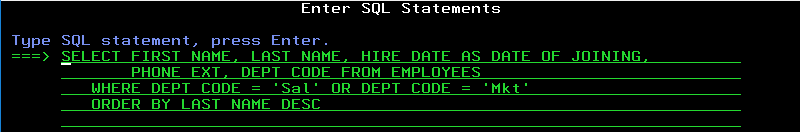


\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
Employee First\_Name Last\_Name Dept Hire\_Date Credit Phone Manager\_id  
Id Code Limit Ext   
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
  
201 Susan Brown Exe 01-Jun-1998 $30.00 3484 (null)  
202 Jim Kern Sal 16-Aug-1999 $25.00 8722 201  
203 Martha Woods Shp 02-Feb-2004 $25.00 7591 201  
204 Ellen Owens Sal 01-Jul-2003 $15.00 6830 202  
205 Henry Perkins Sal 01-Mar-2000 $25.00 5286 202  
206 Carol Rose Act null null null (null)  
207 Dan Smith Shp 01-Dec-2004 $25.00 2259 203  
208 Fred Campbell Shp 01-Apr-2003 $30.00 1752 203  
209 Paula Jacobs Mkt 17-Mar-1999 $15.00 3357 201  
210 Nancy Hoffman Sal 16-Feb-2004 $25.00 2974 203

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
  
  
  
>> Write the query for the following tasks [2 marks each]:  
  
  
  
#1. List the following columns of the Employees table in this order:  
 first name, last name, hire\_date, phone\_ext and department code  
  
Change the name of the hire\_date column to Date\_of\_Joining within the result table. List the employees from the Sales and Marketing department only.  
  
Sort the rows of the result table by the last\_name column in descending order.  
  
SQL Query:  
  
SELECT FIRST\_NAME, LAST\_NAME, HIRE\_DATE AS DATE\_OF\_JOINING,

PHONE\_EXT, DEPT\_CODE FROM EMPLOYEES

WHERE DEPT\_CODE = 'Sal' OR DEPT\_CODE = 'Mkt'

ORDER BY LAST\_NAME DESC   
  


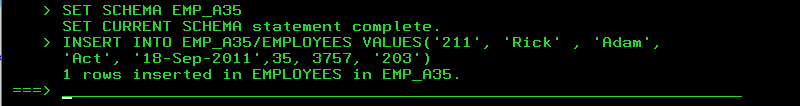
Query Result:



------------------------------------------------------------------------------------------------------------------  
------------------------------------------------------------------------------------------------------------------  
  
#2. Write a SQL statement to add a new employee Rick Adam to the Employees table. Rick joined the Accounting department.  
His date of joining is 18th Sept, 2011. He works for manager with id: 203. He is allotted any phone extension - 3757.  
His credit limit is set to $35.00  
-----  
SQL QUery:

INSERT INTO EMP\_A35/EMPLOYEES VALUES('211', 'Rick' , 'Adam',

'Act', '18-Sep-2011',35, 3757, '203')

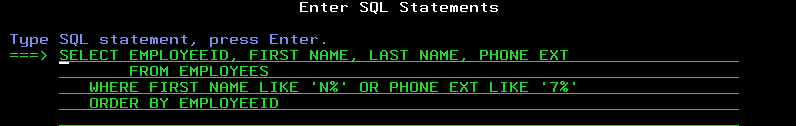


---------------------------------------------------------------------------------------------------------------------  
---------------------------------------------------------------------------------------------------------------------  
  
#3. List the employees that have a phone number starting with 7 or whose names are starting with letter - 'N'.   
Show the employee\_id, first\_name, last\_name and phone\_ext. Sort the rows by employee\_id.  
  
-----  
SQL Query:  
  
 SELECT EMPLOYEEID, FIRST\_NAME, LAST\_NAME, PHONE\_EXT

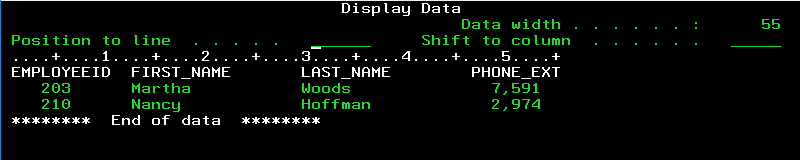
FROM EMPLOYEES

WHERE FIRST\_NAME LIKE 'N%' OR PHONE\_EXT LIKE '7%'

ORDER BY EMPLOYEEID



Query Result:

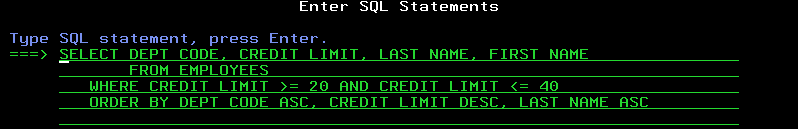


---------------------------------------------------------------------------------------------------------------------  
---------------------------------------------------------------------------------------------------------------------  
  
#4. Using the Employees table, list the following columns:  
 dept\_code, credit\_limit, last\_name, first\_name  
 Place the columns in that order. Sort the rows by :  
 dept\_code in ascending order  
 credit\_limit in descending order  
 last\_name in ascending order.  
And get only those employees who are having credit limits between 20 and 40.  
------  
Sql Query:  
   
 SELECT DEPT\_CODE, CREDIT\_LIMIT, LAST\_NAME, FIRST\_NAME

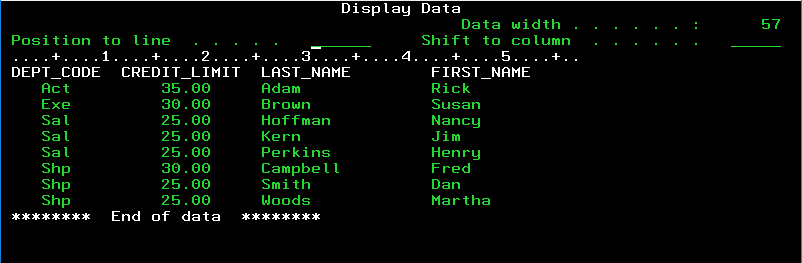
FROM EMPLOYEES

WHERE CREDIT\_LIMIT >= 20 AND CREDIT\_LIMIT <= 40

ORDER BY DEPT\_CODE ASC, CREDIT\_LIMIT DESC, LAST\_NAME ASC



Query Result:



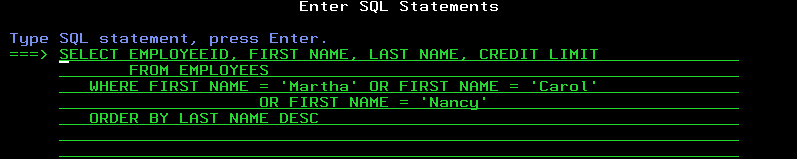
----------------------------------------------------------------------------------------------------------------------  
----------------------------------------------------------------------------------------------------------------------  
  
#5. From the Employees table, list the employee\_id, first\_name, last\_name and credit\_limit columns for employees   
 with the first names: Martha, Carol, Nancy  
 Sort the rows by the last name.  
------  
SQl Query:  
  
 SELECT EMPLOYEEID, FIRST\_NAME, LAST\_NAME, CREDIT\_LIMIT

FROM EMPLOYEES

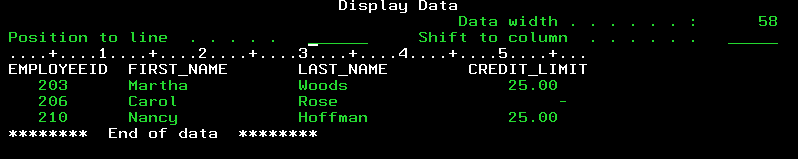
WHERE FIRST\_NAME = 'Martha' OR FIRST\_NAME = 'Carol'

OR FIRST\_NAME = 'Nancy'

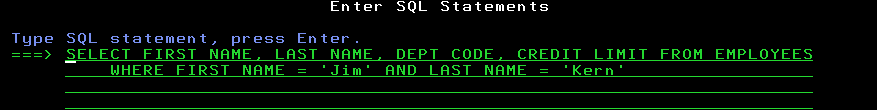
ORDER BY LAST\_NAME DESC



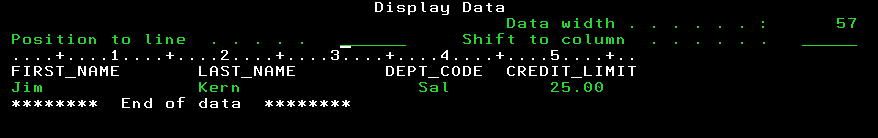
Query Result:

  
  
----------------------------------------------------------------------------------------------------------------------  
----------------------------------------------------------------------------------------------------------------------  
  
#6. From the Employees table, get the department and credit limits for the employees 'Jim Kern'.  
  
------  
SQl Query:  
 SELECT FIRST\_NAME, LAST\_NAME ,DEPT\_CODE, CREDIT\_LIMIT FROM EMPLOYEES

WHERE FIRST\_NAME = 'Jim' AND LAST\_NAME = 'Kern'



Query Result:

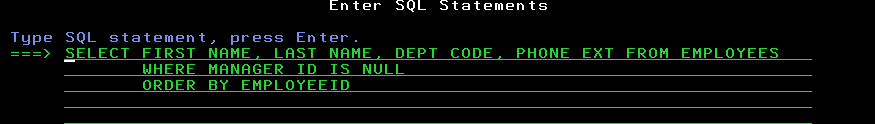


**\*\* I’m adding first name and last name for a better output , on the exercise is asking only Department and Credit Limit thought.\*\***

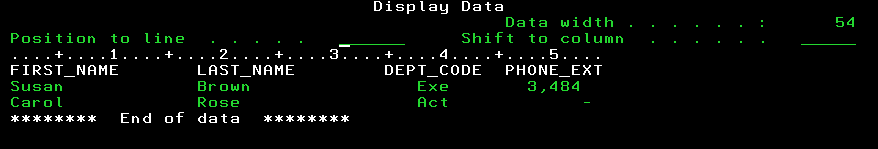
----------------------------------------------------------------------------------------------------------------------  
----------------------------------------------------------------------------------------------------------------------  
  
#7. List the department and phone extension for all the employees who don't have a manager.  
 Sort the rows by the employee\_id.  
----  
Sql Query:  
  
 SELECT DEPT\_CODE, CREDIT\_LIMIT FROM EMPLOYEES

WHERE MANAGER\_ID IS NULL

ORDER BY EMPLOYEEID DESC

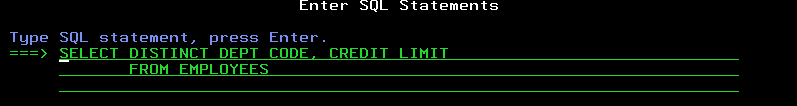


Query Result:

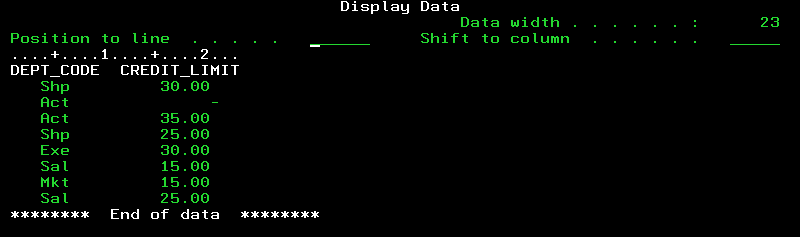


**\*\* I’m adding first name and last name for a better output , on the exercise is asking only Department and Phone Ext thought.\*\***  
  
----------------------------------------------------------------------------------------------------------------------  
----------------------------------------------------------------------------------------------------------------------  
  
#8. Write a query to display distinct department codes credit\_limits.  
  
----  
SQl Query:  
  
 SELECT DISTINCT DEPT\_CODE, CREDIT\_LIMIT

FROM EMPLOYEES

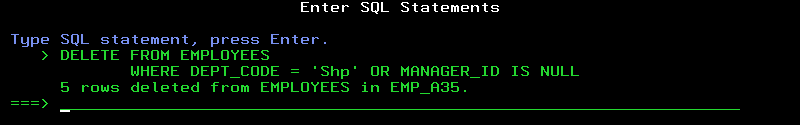


Query Result:



----------------------------------------------------------------------------------------------------------------------  
----------------------------------------------------------------------------------------------------------------------  
  
#9. Delete all the employees who work for Shipping Department or who have manager\_id null.  
  
----  
SQl Query:  
  
 DELETE FROM EMPLOYEES

WHERE DEPT\_CODE = 'Shp' OR MANAGER\_ID IS NULL



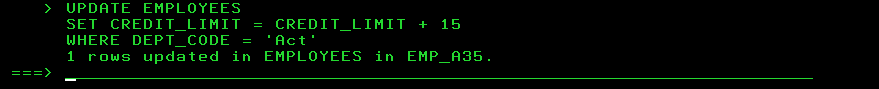
Query Result (Result of the table after deleting rows):



---------------------------------------------------------------------------------------------------------------------  
----------------------------------------------------------------------------------------------------------------------  
  
  
#10. Increase the credit limits of all the employees by $15.00 who work for Accounting department.   
  
----  
SQl Query:  
  
 UPDATE EMPLOYEES

SET CREDIT\_LIMIT = CREDIT\_LIMIT + 15

WHERE DEPT\_CODE = 'Act'

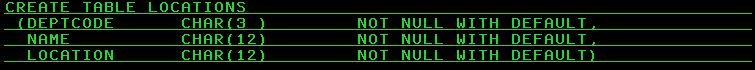


Query Result:



=============================================================================================  
  
 :: PART B::   
  
  
Total Marks : 20  
  
>> Note: Employees is work tables for these queries  
  
  
Table: Employees  
  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
Employee First\_Name Last\_Name Dept Hire\_Date Credit Phone Manager\_id  
Id Code Limit Ext   
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
  
201 Susan Brown Exe 01-Jun-1998 $30.00 3484 (null)  
202 Jim Kern Sal 16-Aug-1999 $25.00 8722 201  
203 Martha Woods Shp 02-Feb-2004 $25.00 7591 201  
204 Ellen Owens Sal 01-Jul-2003 $15.00 6830 202  
205 Henry Perkins Sal 01-Mar-2000 $25.00 5286 202  
206 Carol Rose Act null null null (null)  
207 Dan Smith Shp 01-Dec-2004 $25.00 2259 203  
208 Fred Campbell Shp 01-Apr-2003 $30.00 1752 203  
209 Paula Jacobs Mkt 17-Mar-1999 $15.00 3357 201  
210 Nancy Hoffman Sal 16-Feb-2004 $25.00 2974 203  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
  
Create the following table in the collection you created in Part A, and insert data in it.

**\*\*Table is called LOCATIONS because the table DEPARTMENTS is locked to the ADMIN as stated above\*\***

  
Table: Departments  
  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
Dept Name Location  
Code   
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
Act Accounting Toronto  
Exe Executive Montreal  
Mkt Marketing Vancouver  
Per Personnel Ottawa  
Sal Sales New York  
Shp Shipping Chicago  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

>> Write the query for the following tasks:  
  
---------------------------------------------------------------------------------------------------  
Task#1:  
  
Consider the above two tables. write a query to set up the Referential Integrity   
between these two tables. Also explain what is pre-requisite for setting up the RI between these two  
tables.[5 marks]  
  
>> Ans:

The pre-requisite for setting up the RI is to first declare a PRIMARY KEY on the LOCATIONS table which

Is the “1” table and then create a FOREIGN KEY pointing to that table on the “many” table which

Is EMPLOYEES

ALTER TABLE LOCATIONS

ADD PRIMARY KEY (DEPTCODE)

--------------------------------------------------

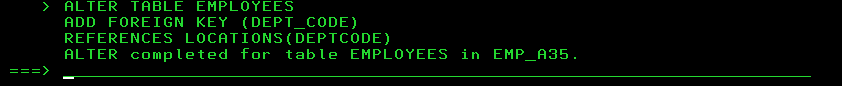
ALTER TABLE EMPLOYEES

ADD FOREIGN KEY (DEPT\_CODE)

REFERENCES LOCATIONS (DEPTCODE)



---------------------------------------------------

  
----------------------------------------------------------------------------------------------------  
----------------------------------------------------------------------------------------------------  
  
Task#2:  
  
For each employee show the employee\_id, first\_name, last\_name, dept\_code and the department name and location.  
Sort the rows by the employee\_id. Show only the employee with ids between 203 and 207.[3 marks]  
  
>> Sql Query:

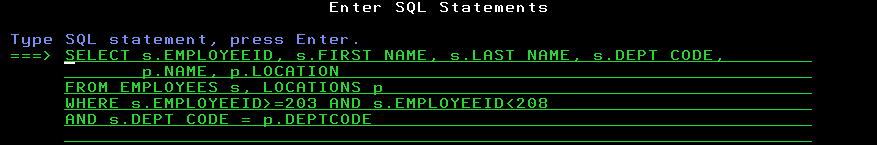
SELECT s.EMPLOYEEID, s.FIRST\_NAME, s.LAST\_NAME, s.DEPT\_CODE,

p.NAME, p.LOCATION

FROM EMPLOYEES s, LOCATIONS p

WHERE s.EMPLOYEEID>=203 AND s.EMPLOYEEID<208

AND s.DEPT\_CODE = p.DEPTCODE

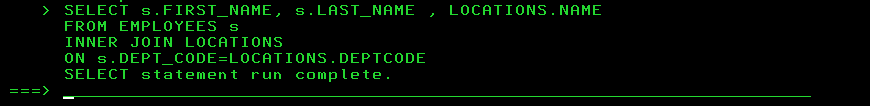


  
  
---------------------------------------------------------------------------------------------------  
---------------------------------------------------------------------------------------------------  
  
Task#3: Consider the above tables for performing the following: [3 marks each]  
  
(A) : Inner Join on dept\_code. Display employee name from the employees table and department name   
 from Department table.   
  
>> Sql Query:  
SELECT s.FIRST\_NAME, s.LAST\_NAME, LOCATIONS.NAME

FROM EMPLOYEES s

INNER JOIN LOCATIONS

ON s.DEPT\_CODE=LOCATIONS.DEPTCODE

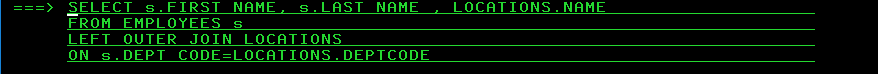


  
  
---------------  
  
  
  
(B) : Left Outer Join on dept\_code. Display employee name from the employees table and department name   
 from Department table.   
  
>> Sql Query:

SELECT s.FIRST\_NAME, s.LAST\_NAME, LOCATIONS.NAME

FROM EMPLOYEES s

LEFT OUTER JOIN LOCATIONS

ON s.DEPT\_CODE=LOCATIONS.DEPTCODE  


  
   
-------------  
  
(C) : Right Outer Join on dept\_code. Display employee name from the employees table and department name   
 from Department table.

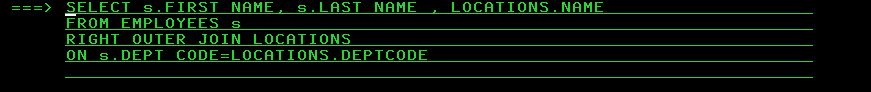
SELECT s.FIRST\_NAME, s.LAST\_NAME, LOCATIONS.NAME

FROM EMPLOYEES s

RIGHT OUTER JOIN LOCATIONS

ON s.DEPT\_CODE=LOCATIONS.DEPTCODE

>> Sql Query:



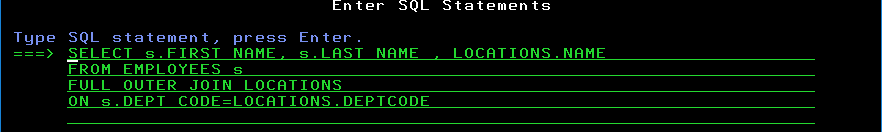
  
  
   
  
-------------  
  
(D) : Full Outer Join on dept\_code. Display employee name from the employees table and department name   
 from Department table.   
  
>> Sql Query:

SELECT s.FIRST\_NAME, s.LAST\_NAME, LOCATIONS.NAME

FROM EMPLOYEES s

FULL OUTER JOIN LOCATIONS

ON s.DEPT\_CODE=LOCATIONS.DEPTCODE



  
   
  
=============================================================================================  
=================================== END - Assignment # 1===================================  
=============================================================================================