Autonomous University of Zacatecas

Academic Unit of Electrical Engineering

Academic Program of Software Engineering

Database Systems Laboratory II
Practice 12 - Displaying Data from Multiple Tables

Professor:
Aldonso Becerra Sánchez

Student:
Daniel Alejandro Morales Castillo

1 Introduction

SQL language allows the realization of projection and selection of data from several tables to satisfy the needs of reports that may be required for a programmer, developer or end user.

In the week of the theory class we saw chapter 6, which tells us about JOINS, it is an interesting topic, the chapter was somewhat extensive but it was possible to understand the concept of the different types of JOINS, in this practice we will use what we learned to solve some tasks.

2 Development

Activity 1

Write the section that describes the Work developed in the following activities. Read all the choices carefully because there might be more than one correct answer. Choose all the correct answers for each question.

Explain the reason for your answer.

WRITE SELECT STATEMENTS TO ACCESS DATA FROM MORE THAN ONE TABLE USING EQUIJOINS AND NONEQUI-JOINS

1. The EMPLOYEES and DEPARTMENTS tables have two identically named columns: DEPARTMENTID and MANAGERID. Which of these statements joins these tables based only on common DEPARTMENTID values? (Choose all that apply.)

A. SELECT * FROM EMPLOYEES NATURAL JOIN DEPARTMENTS;
B. SELECT * FROM EMPLOYEES E NATURAL JOIN DEPARTMENTS D ON
E.DEPARTMENTIDD.DEPARTMENTID;
C. SELECT * FROM EMPLOYEES NATURAL JOIN DEPARTMENTS USING (DEPARTMENTID);
D. None of the above

Answer: D
All the statements above use NATURAL JOIN, the natural join is based in all the equals columns, it is a procedure implicit of the NATURAL JOIN, to retrieve data based only on common DEPARTMENT IDs we must use JOIN USING or JOIN ON, using NATURAL JOIN is an incorrect way.

2. The EMPLOYEES and DEPARTMENTS tables have two identically named columns: DEPARTMENTID and MANAGERID. Which statements join these tables based on both column values? (Choose all that apply.)

A. SELECT * FROM EMPLOYEES NATURAL JOIN DEPARTMENTS;
B. SELECT * FROM EMPLOYEES JOIN DEPARTMENTS USING (DEPARTMENTID,MANAGERID);
C. SELECT * FROM EMPLOYEES E JOIN DEPARTMENTS D ON
E.DEPARTMENTID=D.DEPARTMENTID AND
E.MANAGERID=D.MANAGERID;
D. None of the above

Answer: A, B, C

The A, B and C sentences used both columns to join the tables, the NATURAL JOIN did it implicitly and the other 2 JOINS did it explicitly

IDENTIFY THE AVAILABLE GROUP FUNCTIONS

3. Which join is performed by the following query? SELECT E.JOBID,J.JOBID FROM EMPLOYEES E JOIN JOBS J ON (E.SALARY ¡ J.MAXSALARY); (Choose the best answer.)

A. Equijoin
B. Nonequijoin
C. Cross join
D. Outer join

Answer: B

The answer is B, the JOIN is a nonequijoin because it is using a comparison operator different than equal (=).

4. Which of the following statements are syntactically correct? (Choose all that apply.)

A. SELECT * FROM EMPLOYEES E JOIN DEPARTMENTS D USING (DEPARTMENTID);
B. SELECT * FROM EMPLOYEES JOIN DEPARTMENTS D USING (D.DEPARTMENTID);
C. SELECT D.DEPARTMENTID FROM EMPLOYEES JOIN DEPARTMENTS D USING (DEPARTMENTID);
D. None of the above

Answer: A, C

These two statements are using JOIN USING correctly, the letter B returns an error because you can not use the table alias with the column in the USING because the column lost the belong to the table in the join.

5. Which of the following statements are syntactically correct? (Choose all that apply.)) FROM EMPLOYEES; Assume there are ten employee records and each contains a SALARY value of 100, except for one employee, who has a null value in the SALARY field. (Choose the best answer.

A. SELECT E.EMPLOYEEID, J.JOBID PREVIOUSJOB, E.JOBID CURRENTJOB FROM JOBHISTORY J CROSS JOIN EMPLOYEES E ON (J.STARTDATE=E.HIREDATE);

B. SELECT E.EMPLOYEEID, J.JOBID PREVIOUSJOB, E.JOBID CURRENTJOB FROM JOBHISTORY J JOIN EMPLOYEES E ON (J.STARTDATE=E.HIREDATE);

C. SELECT E.EMPLOYEEID, J.JOBID PREVIOUSJOB, E.JOBID CURRENTJOB FROM JOBHISTORY J OUTER JOIN EMPLOYEES E ON (J.STARTDATE=E.HIREDATE);

D. None of the above

Answer: B

The letter A are incorrect because in a CROSS JOIN you don not have to use ON, the letter C are incorrect because you need to specify what kind of OUTER JOIN you are using, the letter B are correctly wrote.

GROUP DATA USING THE GROUP BY CLAUSE

6. Choose one correct statement regarding the following query: FROM EMPLOYEES E JOIN DEPARTMENTS D ON (D.DEPARTMENTID=E.DEPARTMEN JOIN LOCATIONSL ON (L.LOCATIONID =D.LOCATIONID);

A. Joining three tables is not permitted.

B. A Cartesian product is generated.

C. The JOIN...ON clause may be used for joins between multiple tables.

D. None of the above

Answer: C

The JOIN ON can be used to join multiple tables, two tables is not the limit, you can use JOIN with more tables.

JOIN A TABLE TO ITSELF USING A SELF-JOIN

7. How many rows are returned after executing the following statement? SELECT * FROM REGIONS R1 JOIN REGIONS R2 ON (R1.REGIONID=LENGTH(R2. REGIONNAME)/2); The REGIONS table contains the following row data. (Choose the best answer.)

A. 2
B. 3
C. 4
D. None ofthe above

Answer: 3
rows are returned because there are three names with lengths of 4, 6 and 8, the lengths divided by 2 are 2, 3 and 4, there are 3 records with id 2, 3 and 4, there are 3 matches

VIEW DATA THAT DOES NOT MEET A JOIN CONDITION USING OUTER JOINS

8. Choose one correct statement regarding the following query. SELECT C.COUNTRYID FROM LOCATIONS L RIGHT OUTER JOIN COUNTRIES C ON (L.COUNTRYID=C.COUNTRYID) WHERE L.COUNTRYID is NULL

A. No rows in the LOCATIONS table have the COUNTRYID values returned.
B. No rows in the COUNTRIES table have the COUNTRYID values returned.
C. The rows returned represent the COUNTRYID values for all the rows in the LOCATIONS table.
D. None of the above

Answer: B
This is because we are using RIGHT OUTER JOIN, the values returned represent the rows of the COUNTRIES table that did not match with the rows in the LOCATIONS table.

INCLUDE OR EXCLUDE GROUPED ROWS USING THE HAVING CLAUSE

9. Which of the following statements are syntactically correct? (Choose all that apply.

A. SELECT JH.JOB ID FROM JOB HISTORY JH RIGHT OUTER JOIN JOBS J ON JH.JOB ID=J.JOB ID

B. SELECT JOB ID FROM JOB HISTORY JH RIGHT OUTER JOIN JOBS J ON(JH.JOB ID=J.JOB ID)

C. SELECT JOB HISTORY.JOB ID FROM JOB HISTORY OUTER JOIN JOBS ON JOB HISTORY.JOB ID=JOBS.JOB ID

D. None of the above

Answer: A

The sentence A are correctly wrote, the sentence B need to specify the table of the JOB ID column in the beginning of the SELECT, the sentence C need to specify what kind of OUTER JOIN we are using.

GENERATE A CARTESIAN PRODUCT OF TWO OR MORE TABLES

10. If the REGIONS table, which contains 4 rows, is cross joined to the COUNTRIES table, which contains 25 rows, how many rows appear in the final results set? (Choose the best answer.)

A. 100 rows

B. 4 rows

C. 25 rows

D. None of the above

Answer: A

The result are 100 rows because 4 rows multiplied by 25 rows equals 100 rows.

Activity 2:

Propose an answer to the following issues:

• You are required to retrieve information from multiple tables, group the results, and apply an aggregate function to them. Can a group function be used against data from multiple table sources?

I think yes, because the group functions are used with multiple rows to produce multiple results, the JOINS retrieve multiple rows that can be used in a group functions.

• When joining two tables, there is a risk that between them they contain common column names. Does Oracle know which tables to fetch data from if such columns are present in the SELECT list?

No, an error is returned because the column is ambiguous, you need to specify the table that belongs the column.

• The NATURAL JOIN clause is used to join rows from two tables based on columns with common names sharing identical values. Is it possible to join two tables based on some of the shared columns and not all of them?

NATURAL JOIN do an implicit work to join the tables based on the equal column names, if you want to specify the column the best option is use another JOIN, maybe a JOIN USING or JOIN ON

• The data in two tables you wish to join is related but does not share any identically named columns. Is it possible to join tables using columns that do not share the same name?

Yes, you can use a JOIN ON and specify the columns to compare, not necessarily with the same name, for example you can use EMPLOYEE ID and MANAGER ID, the unique restriction is the data type

• You wish to divide staff into four groups named after the four regions in the REGIONS table. Is it possible to obtain a list of EMPLOYEEID, LASTNAME, and REGIONNAME values for each employee by joining the EMPLOYEEID and REGIONID columns in a round-robin manner?

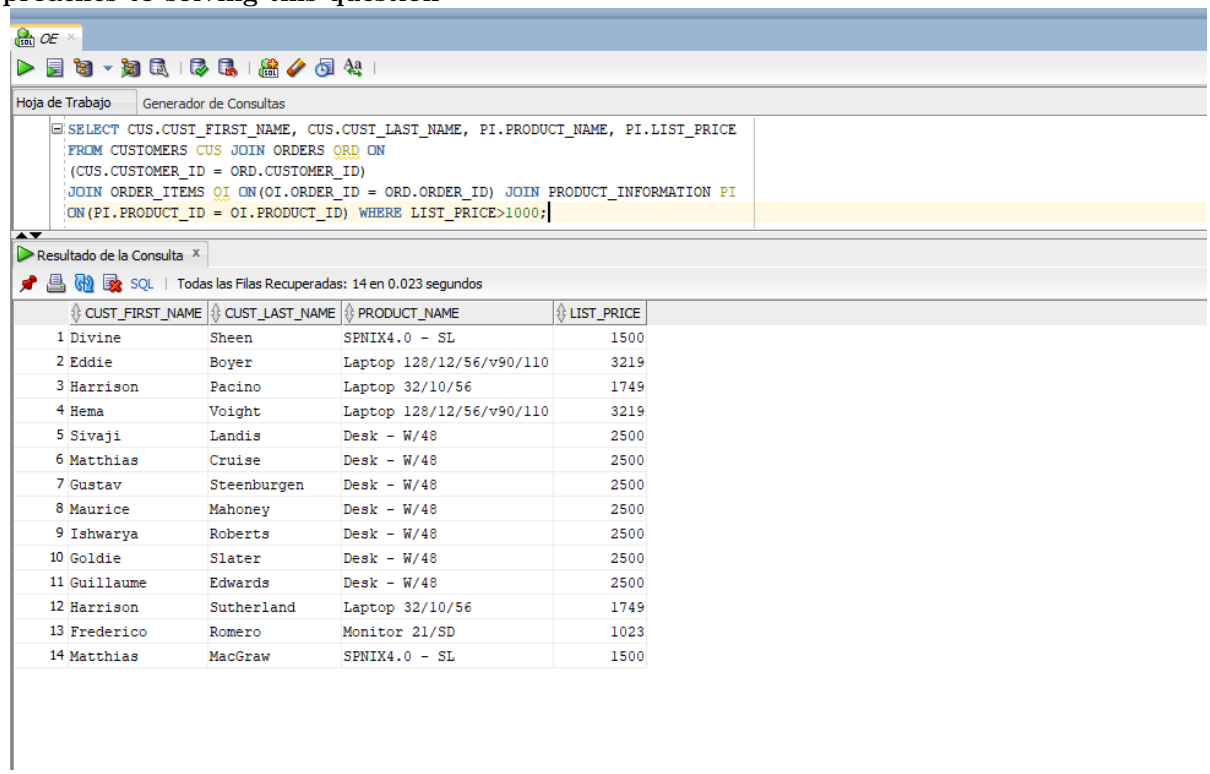
The two tables do not match, there are no columns to join, maybe you can use another table like departments and join the three tables to retrieve correct data with a correct match

• You are required to retrieve a list of DEPARTMENTNAME and LASTNAME values for all departments, including those that currently have no employees assigned to them. In such cases the string 'No Employees' should be displayed as the LASTNAME column value. Can this be done using joins?

It can be done, but not using only joins, maybe you can use functions like NVL to remplace the NULL values and customize the output

Activity 3:

Connect to the OE schema and complete the following tasks. You are required to produce a report of customers who purchased products with list prices of more than 1000. The report must contain customer first and last names and the product names and their list prices. Customer information is stored in the CUSTOMERS table, which has the CUSTOMERID column as its primary key. The product name and list price details are stored in the PRODUCTINFORMATION table with the PRODUCTID column as its primary key. Two other related tables may assist in generating the required report: the ORDERS table, which stores the CUSTOMERID and ORDERID information, and the ORDERITEMS table, which stores the PRODUCTID values associated with each ORDERID. There are several approaches to solving this question

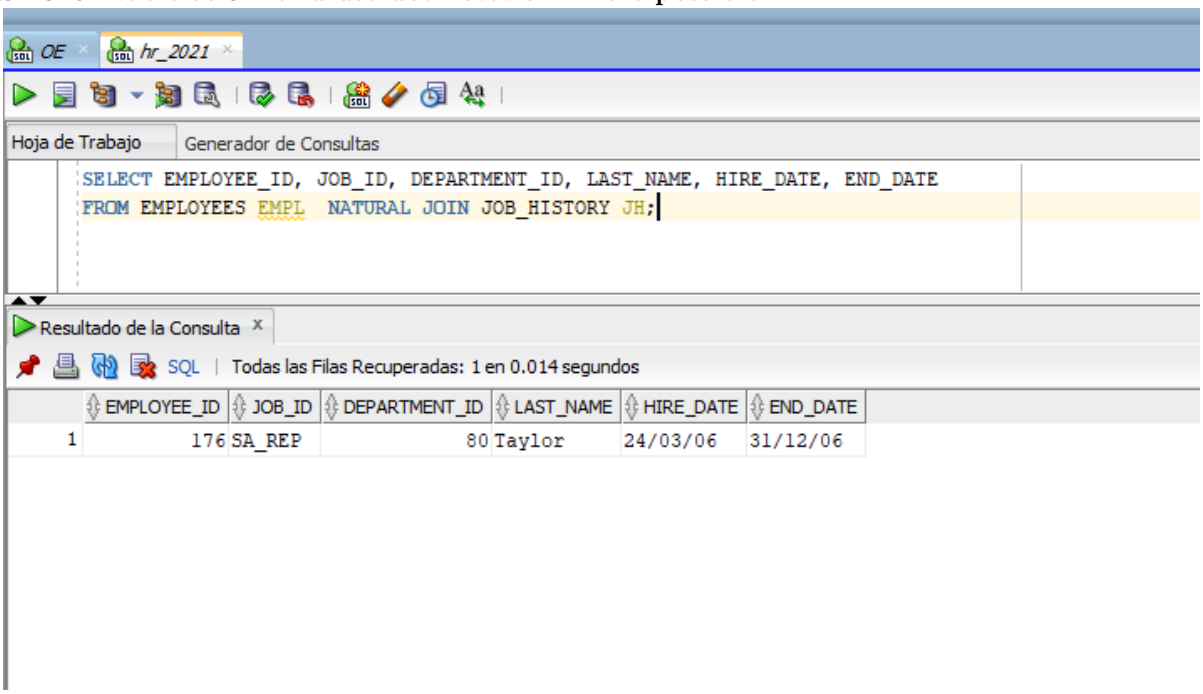


	CUST_FIRST_NAME	CUST_LAST_NAME	PRODUCT_NAME	LIST_PRICE
1	Divine	Sheen	SPNIX4.0 - SL	1500
2	Eddie	Boyer	Laptop 128/12/56/v90/110	3219
3	Harrison	Pacino	Laptop 32/10/56	1749
4	Hema	Voight	Laptop 128/12/56/v90/110	3219
5	Sivaji	Landis	Desk - W/48	2500
6	Matthias	Cruise	Desk - W/48	2500
7	Gustav	Steenburgen	Desk - W/48	2500
8	Maurice	Mahoney	Desk - W/48	2500
9	Ishwarya	Roberts	Desk - W/48	2500
10	Goldie	Slater	Desk - W/48	2500
11	Guillaume	Edwards	Desk - W/48	2500
12	Harrison	Sutherland	Laptop 32/10/56	1749
13	Frederico	Romero	Monitor 21/SD	1023
14	Matthias	MacGraw	SPNIX4.0 - SL	1500

Activity 4:

This exercise must be performed using HR schema.

a) The JOBHISTORY table shares three identically named columns with the EMPLOYEES table: EMPLOYEEID, JOBID, and DEPARTMENTID. You are required to describe the tables and fetch the EMPLOYEEID, JOBID, DEPARTMENTID, LASTNAME, HIREDATE, and ENDDATE values for all rows retrieved using a pure natural join. Alias the EMPLOYEES table as EMP and the JOBHISTORY table as JH and use dot notation where possible.



b) Each record in the DEPARTMENTS table has a MANAGERID column matching an EMPLOYEEID value in the EMPLOYEES table. You are required to produce a report with one column aliased as Managers. Each row must contain a sentence of the format FIRSTNAME LASTNAME is manager of the DEPARTMENTNAME department. Alias the EMPLOYEES table as E and the DEPARTMENTS table as D and use dot notation where possible.

```
--b
SELECT E.FIRST_NAME || '' || E.LAST_NAME || 'is manager of the' || D.DEPARTMENT_NAME || '.' AS "MANAGERS"
FROM EMPLOYEES E JOIN DEPARTMENTS D ON(E.EMPLOYEE_ID = D.MANAGER_ID) ;
```

Resultado de la Consulta

SQL | Todas las Filas Recuperadas: 11 en 0.012 segundos

	MANAGERS
1	StevenKingis manager of theExecutive.
2	AlexanderHunoldis manager of theIT.
3	NancyGreenbergis manager of theFinance.
4	DenRaphaelyis manager of thePurchasing.
5	AdamFrippis manager of theShipping.
6	JohnRussellis manager of theSales.
7	JenniferWhalenis manager of theAdministration.
8	MichaelHartsteinis manager of theMarketing.
9	SusanMavrisis manager of theHuman Resources.
10	HermannBaeris manager of thePublic Relations.
11	ShelleyHigginsis manager of theAccounting.

c) There is a hierarchical relationship between employees and their managers. For each row in the EMPLOYEES table the MANAGERID column stores the EMPLOYEEID of every employee's manager. Using a self-join on the EMPLOYEES table, you are required to retrieve the employee's LASTNAME, EMPLOYEEID, manager's LASTNAME, and employee's DEPARTMENTID for the rows with DEPARMENTID values of 10, 20, or 30. Alias the EMPLOYEES table as E and the second instance of the EMPLOYEES table as M. Sort the results based on the DEPARTMENTID column.

```
--c
SELECT E.LAST_NAME "EMPLOYEE", E.EMPLOYEE_ID, E.MANAGER_ID "MANAGER", M.LAST_NAME, E.DEPARTMENT_ID
FROM EMPLOYEES E JOIN EMPLOYEES M
ON(M.EMPLOYEE_ID = E.MANAGER_ID) WHERE E.DEPARTMENT_ID IN(10,20,30)
ORDER BY DEPARTMENT_ID;
```

Resultado de la Consulta

SQL | Todas las Filas Recuperadas: 9 en 0.01 segundos

	EMPLOYEE	EMPLOYEE_ID	MANAGER	LAST_NAME	DEPARTMENT_ID
1	Whalen	200	101	Kochhar	10
2	Fay	202	201	Hartstein	20
3	Hartstein	201	100	King	20
4	Tobias	117	114	Raphaely	30
5	Baida	116	114	Raphaely	30
6	Himuro	118	114	Raphaely	30
7	Khoo	115	114	Raphaely	30
8	Raphaely	114	100	King	30
9	Colmenares	119	114	Raphaely	30

d) Write a query for the HR department to produce the addresses of all the departments. Use the LOCATIONS and COUNTRIES tables. Show the location ID, street address, city, state or province, and country in the output. Use a NATURAL JOIN to produce the results.

```
SELECT * FROM LOCATIONS;
--d
SELECT L.LOCATION_ID, L.STREET_ADDRESS,L.CITY, L.STATE_PROVINCE, C.COUNTRY_NAME
FROM LOCATIONS L NATURAL JOIN COUNTRIES C;
```

Resultado de la Consulta

SQL | Todas las Filas Recuperadas: 23 en 0.011 segundos

	LOCATION_ID	STREET_ADDRESS	CITY	STATE_PROVINCE	COUNTRY_NAME
1	1000	1297 Via Cola di Rie	Roma	(null)	Italy
2	1100	93091 Calle della Testa	Venice	(null)	Italy
3	1200	2017 Shinjuku-ku	Tokyo	Tokyo Prefecture	Japan
4	1300	9450 Kamiya-cho	Hiroshima	(null)	Japan
5	1400	2014 Jabberwocky Rd	Southlake	Texas	United States of America
6	1500	2011 Interiors Blvd	South San Francisco	California	United States of America
7	1600	2007 Zagora St	South Brunswick	New Jersey	United States of America
8	1700	2004 Charade Rd	Seattle	Washington	United States of America
9	1800	147 Spadina Ave	Toronto	Ontario	Canada
10	1900	6092 Boxwood St	Whitehorse	Yukon	Canada
11	2000	40-5-12 Laogianggen	Beijing	(null)	China
12	2100	1298 Vileparle (E)	Bombay	Maharashtra	India
13	2200	12-98 Victoria Street	Sydney	New South Wales	Australia
14	2300	198 Clementi North	Singapore	(null)	Singapore
15	2400	8204 Arthur St	London	(null)	United Kingdom
16	2500	Magdalen Centre, The Oxford Science Park	Oxford	Oxford	United Kingdom
17	2600	9702 Chester Road	Stretford	Manchester	United Kingdom
18	2700	Schwanthalerstr. 7031	Munich	Bavaria	Germany
19	2800	Rua Frei Caneca 1360	Sao Paulo	Sao Paulo	Brazil
20	2900	20 Rue des Corps-Saints	Geneva	Geneve	Switzerland
21	3000	Murtenstrasse 921	Bern	BE	Switzerland
22	3100	Pieter Breughelstraat 837	Utrecht	Utrecht	Netherlands
23	3200	Mariano Escobedo 9991	Mexico City	Distrito Federal,	Mexico

e) The HR department needs a report of all employees. Write a query to display the last name, department number, and department name for all the employees.

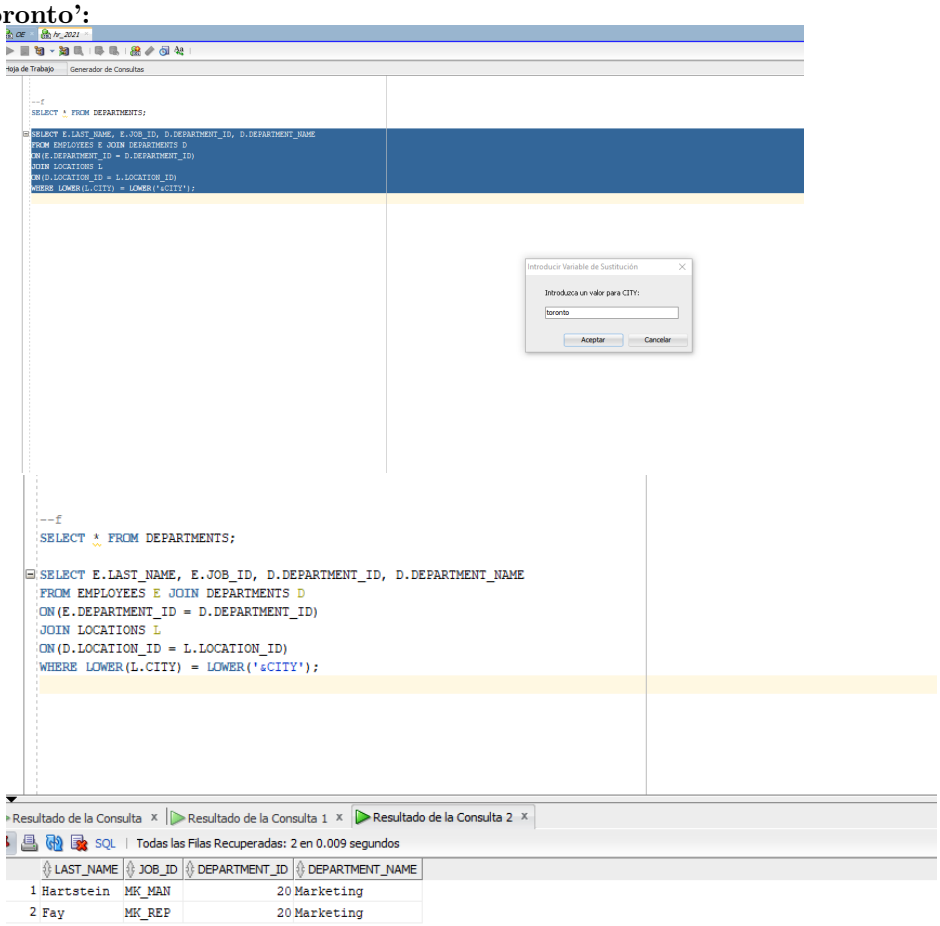
```
--e
SELECT E.LAST_NAME, E.DEPARTMENT_ID, D.DEPARTMENT_NAME FROM EMPLOYEES
E LEFT OUTER JOIN DEPARTMENTS D ON(E.DEPARTMENT_ID = D.DEPARTMENT_ID);
```

Resultado de la Consulta

SQL | Se han recuperado 50 filas en 0.005 segundos

	LAST_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
1	Whalen	10	Administration
2	Fay	20	Marketing
3	Hartstein	20	Marketing
4	Colmenares	30	Purchasing
5	Himuro	30	Purchasing
6	Tobias	30	Purchasing
7	Baida	30	Purchasing
8	Khoo	30	Purchasing
9	Raphaely	30	Purchasing
10	Mavris	40	Human Resources
11	Grant	50	Shipping
12	OConnell	50	Shipping
13	Feeney	50	Shipping
14	Walsh	50	Shipping
15	Jones	50	Shipping
16	McCain	50	Shipping
17	Everett	50	Shipping
18	Bell	50	Shipping
19	Perkins	50	Shipping
20	Gates	50	Shipping
21	Dilly	50	Shipping
22	Chung	50	Shipping
23	Cabrio	50	Shipping
24	Dellinger	50	Shipping
25	Bull	50	Shipping
26	Sarchand	50	Shipping
27	Geoni	50	Shipping
28	Sullivan	50	Shipping

f) The HR department needs a report to display the last name, job, department number, and department name for all employees who work in a city entered by a user parameter. For instance, city equals 'toronto':



g) Crete a report to display employees' last names and employee number along with their managers' last names and manager number. Label the columns Employee, Emp , Manager, and Mgr, respectively

```
--g
SELECT EMP#.LAST_NAME "EMPLOYEE" , EMP#.EMPLOYEE_ID, MAN#.LAST_NAME "MANAGER", MAN#.EMPLOYEE_ID
FROM EMPLOYEES EMP# JOIN EMPLOYEES MAN#
ON(EMP#.MANAGER_ID = MAN#.EMPLOYEE_ID);
```

Resultado de la Consulta | Resultado de la Consulta 1 | Resultado de la Consulta 2 | Resultado de la Consulta 3

SQL | Se han recuperado 50 filas en 0.006 segundos

	EMPLOYEE	EMPLOYEE_ID	MAN...	EMPLOYEE_ID_1
1	Kumar	173	Cambrault	148
2	Bates	172	Cambrault	148
3	Smith	171	Cambrault	148
4	Fox	170	Cambrault	148
5	Bloom	169	Cambrault	148
6	Ozer	168	Cambrault	148
7	Hunold	103	De Haan	102
8	Banda	167	Errazuriz	147
9	Ande	166	Errazuriz	147
10	Lee	165	Errazuriz	147
11	Marvins	164	Errazuriz	147
12	Greene	163	Errazuriz	147
13	Vishney	162	Errazuriz	147
14	Cabrio	187	Fripp	121
15	Dellinger	186	Fripp	121
16	Bull	185	Fripp	121
17	Sarchand	184	Fripp	121
18	Olson	132	Fripp	121
19	Marlow	131	Fripp	121
20	Atkinson	130	Fripp	121
21	Bissot	129	Fripp	121
22	Popp	113	Greenberg	108
23	Urman	112	Greenberg	108
24	Sciarra	111	Greenberg	108

h) Modify previous sentence to display all employees including those who have no manager. Order the results by the employee number.

```
--h
SELECT EMP#.LAST_NAME "EMPLOYEE" , EMP#.EMPLOYEE_ID, MAN#.LAST_NAME "MANAGER", MAN#.EMPLOYEE_ID
FROM EMPLOYEES EMP# JOIN EMPLOYEES MAN#
ON(EMP#.MANAGER_ID = MAN#.EMPLOYEE_ID) ORDER BY EMP#.EMPLOYEE_ID;

--i

--j
```

Resultado de la Consulta | Resultado de la Consulta 1 | Resultado de la Consulta 2 | Resultado de la Consulta 3

SQL | Se han recuperado 50 filas en 0.005 segundos

	EMPLOYEE	EMPLOYEE_ID	MANAGER	EMPLOYEE_ID_1
1	Kochhar	101	King	100
2	De Haan	102	King	100
3	Hunold	103	De Haan	102
4	Ernst	104	Hunold	103
5	Austin	105	Hunold	103
6	Pataballa	106	Hunold	103
7	Lorentz	107	Hunold	103
8	Greenberg	108	Kochhar	101
9	Faviet	109	Greenberg	108
10	Chen	110	Greenberg	108
11	Sciarra	111	Greenberg	108
12	Urman	112	Greenberg	108
13	Popp	113	Greenberg	108
14	Raphaely	114	King	100
15	Khoo	115	Raphaely	114
16	Baida	116	Raphaely	114
17	Tobias	117	Raphaely	114
18	Himuro	118	Raphaely	114
19	Colmenares	119	Raphaely	114
20	Weiss	120	King	100
21	Fripp	121	King	100
22	Kaufling	122	King	100
23	Vollman	123	King	100
24	Mourgos	124	King	100

i) Create a report for the HR department that displays employee last names, department name for all the employees along with their colleagues. Give each column an appropriate label and order the results as you consider appropriate to the study case.

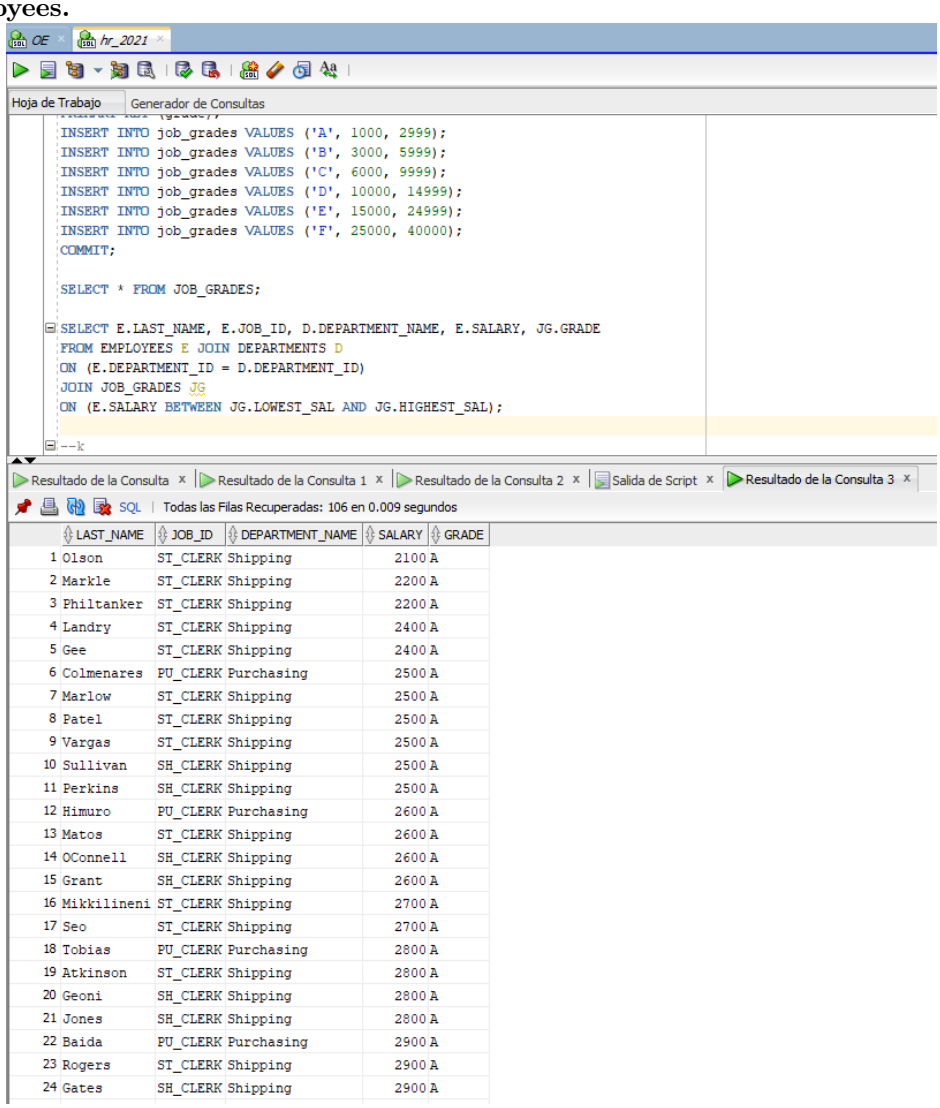
```
--i
SELECT E1.DEPARTMENT_ID "DEPARTMENT", E1.LAST_NAME "EMPLOYEE", E2.LAST_NAME "COLLEAGE"
FROM EMPLOYEES E1 JOIN EMPLOYEES E2 ON (E1.DEPARTMENT_ID = E2.DEPARTMENT_ID)
AND E1.EMPLOYEE_ID <> E2.EMPLOYEE_ID
ORDER BY E1.DEPARTMENT_ID, E1.LAST_NAME;


--j

--k

--l

--m
```

Resultado de la Consulta | Resultado de la Consulta 1 | Resultado de la Consulta 2 | Resultado de la Consulta 3

SQL | Se han recuperado 50 filas en 0.006 segundos

	DEPARTMENT	EMPLOYEE	COLLEAGE
1	20	Fay	Hartstein
2	20	Hartstein	Fay
3	30	Baida	Khoo
4	30	Baida	Tobias
5	30	Baida	Raphaely
6	30	Baida	Colmenares
7	30	Baida	Himuro
8	30	Colmenares	Tobias
9	30	Colmenares	Khoo
10	30	Colmenares	Baida
11	30	Colmenares	Himuro
12	30	Colmenares	Raphaely
13	30	Himuro	Raphaely
14	30	Himuro	Baida
15	30	Himuro	Khoo
16	30	Himuro	Tobias
17	30	Himuro	Colmenares
18	30	Khoo	Baida
19	30	Khoo	Raphaely
20	30	Khoo	Colmenares
21	30	Khoo	Himuro
22	30	Khoo	Tobias
23	30	Raphaely	Himuro
24	30	Raphaely	Tobias

j) The HR department needs a report on job grades and salaries. To familiarize yourself with the JOBGRADES table, first show the structure of the JOBGRADES table. Then create a query that displays the name, job, department name, salary, and grade for all employees.



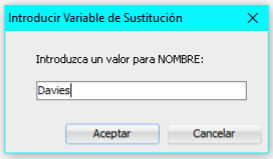
```
INSERT INTO job_grades VALUES ('A', 1000, 2999);
INSERT INTO job_grades VALUES ('B', 3000, 5999);
INSERT INTO job_grades VALUES ('C', 6000, 9999);
INSERT INTO job_grades VALUES ('D', 10000, 14999);
INSERT INTO job_grades VALUES ('E', 15000, 24999);
INSERT INTO job_grades VALUES ('F', 25000, 40000);
COMMIT;

SELECT * FROM JOB_GRADES;

SELECT E.LAST_NAME, E.JOB_ID, D.DEPARTMENT_NAME, E.SALARY, JG.GRADE
FROM EMPLOYEES E JOIN DEPARTMENTS D
ON (E.DEPARTMENT_ID = D.DEPARTMENT_ID)
JOIN JOB_GRADES JG
ON (E.SALARY BETWEEN JG.LOWEST_SAL AND JG.HIGHEST_SAL);
```

Todas las Filas Recuperadas: 106 en 0.009 segundos

	LAST_NAME	JOB_ID	DEPARTMENT_NAME	SALARY	GRADE
1	Olson	ST_CLERK	Shipping	2100	A
2	Markle	ST_CLERK	Shipping	2200	A
3	Philtanker	ST_CLERK	Shipping	2200	A
4	Landry	ST_CLERK	Shipping	2400	A
5	Gee	ST_CLERK	Shipping	2400	A
6	Colmenares	PU_CLERK	Purchasing	2500	A
7	Marlow	ST_CLERK	Shipping	2500	A
8	Patel	ST_CLERK	Shipping	2500	A
9	Vargas	ST_CLERK	Shipping	2500	A
10	Sullivan	SH_CLERK	Shipping	2500	A
11	Perkins	SH_CLERK	Shipping	2500	A
12	Himuro	PU_CLERK	Purchasing	2600	A
13	Matos	ST_CLERK	Shipping	2600	A
14	OConnell	SH_CLERK	Shipping	2600	A
15	Grant	SH_CLERK	Shipping	2600	A
16	Mikkilineni	ST_CLERK	Shipping	2700	A
17	Seo	ST_CLERK	Shipping	2700	A
18	Tobias	PU_CLERK	Purchasing	2800	A
19	Atkinson	ST_CLERK	Shipping	2800	A
20	Geoni	SH_CLERK	Shipping	2800	A
21	Jones	SH_CLERK	Shipping	2800	A
22	Baida	PU_CLERK	Purchasing	2900	A
23	Rogers	ST_CLERK	Shipping	2900	A
24	Gates	SH_CLERK	Shipping	2900	A

k) HR department wants to create a query to display the name and hire date of any employee hired after an employee entered by a user parameter. For Instance, after 'Davies':

```
--k
SELECT E.LAST_NAME, E.HIRE_DATE FROM EMPLOYEES E JOIN EMPLOYEES E2
ON(E.EMPLOYEE_ID = E2.EMPLOYEE_ID)
AND E.HIRE_DATE > (SELECT HIRE_DATE FROM EMPLOYEES WHERE LOWER(LAST_NAME)=LOWER('&NOMBRE'))
ORDER BY E.LAST_NAME;
--l

--m

--n
```



```
--k
SELECT E.LAST_NAME, E.HIRE_DATE FROM EMPLOYEES E JOIN EMPLOYEES E2
ON(E.EMPLOYEE_ID = E2.EMPLOYEE_ID)
AND E.HIRE_DATE > (SELECT HIRE_DATE FROM EMPLOYEES WHERE LOWER(LAST_NAME)=LOWER('&NOMBRE'))
ORDER BY E.LAST_NAME;
--l

--m

--n
```

Resultado de la Consulta

SQL | Se han recuperado 50 filas en 0.004 segundos

	LAST_NAME	HIRE_DATE
1	Ande	24/03/08
2	Atkinson	30/10/05
3	Austin	25/06/05
4	Baida	24/12/05
5	Banda	21/04/08
6	Bates	24/03/07
7	Bernstein	24/03/05
8	Bissot	20/08/05
9	Bloom	23/03/06
10	Bull	20/02/05
11	Cabrio	07/02/07
12	Cambrault	15/10/07
13	Cambrault	09/12/06
14	Chen	28/09/05
15	Chung	14/06/05
16	Colmenares	10/08/07
17	Dellinger	24/06/06
18	Dilly	13/08/05
19	Doran	15/12/05
20	Ernst	21/05/07
21	Errazuriz	10/03/05
22	Everett	03/03/05
23	Fay	17/08/05
24	Feeney	23/05/06

l) The HR department needs to find the names and hire dates for all employees who were hired before their managers, along with their managers' names and hire dates.

```
--l
SELECT E.LAST_NAME, E.HIRE_DATE, M.LAST_NAME, M.HIRE_DATE
FROM EMPLOYEES E JOIN EMPLOYEES M
ON(E.MANAGER_ID = M.EMPLOYEE_ID)
AND M.HIRE_DATE>E.HIRE_DATE;
--m

--n
```

Resultado de la Consulta

SQL | Todas las Filas Recuperadas: 37 en 0.004 segundos

	LAST_NAME	HIRE_DATE	LAST_NAME_1	HIRE_DATE_1
1	Kaufling	01/05/03	King	17/06/03
2	Raphaely	07/12/02	King	17/06/03
3	De Haan	13/01/01	King	17/06/03
4	Higgins	07/06/02	Kochhar	21/09/05
5	Baer	07/06/02	Kochhar	21/09/05
6	Mavris	07/06/02	Kochhar	21/09/05
7	Whalen	17/09/03	Kochhar	21/09/05
8	Greenberg	17/08/02	Kochhar	21/09/05
9	Austin	25/06/05	Hunold	03/01/06
10	Faviet	16/08/02	Greenberg	17/08/02
11	Bull	20/02/05	Fripp	10/04/05
12	Sarchand	27/01/04	Fripp	10/04/05
13	Marlow	16/02/05	Fripp	10/04/05
14	Everett	03/03/05	Vollman	10/10/05
15	Bell	04/02/04	Vollman	10/10/05
16	Ladwig	14/07/03	Vollman	10/10/05
17	OConnell	21/06/07	Mourgos	16/11/07
18	Feeney	23/05/06	Mourgos	16/11/07
19	Walsh	24/04/06	Mourgos	16/11/07
20	Vargas	09/07/06	Mourgos	16/11/07
21	Matos	15/03/06	Mourgos	16/11/07
22	Davies	29/01/05	Mourgos	16/11/07
23	Rajs	17/10/03	Mourgos	16/11/07
24	McEwen	01/08/04	Partners	05/01/05

m) The DEPARTMENTS table contains details of all departments in the organization. You are required to retrieve the DEPARTMENTNAME and DEPARTMENTID values for those departments to which no employees are currently assigned.

```
--m
SELECT D.DEPARTMENT_NAME, D.DEPARTMENT_ID FROM DEPARTMENTS
D LEFT OUTER JOIN EMPLOYEES E ON(D.DEPARTMENT_ID = E.DEPARTMENT_ID)
WHERE E.EMPLOYEE_ID IS NULL;
--n
```

Resultado de la Consulta

SQL | Todas las Filas Recuperadas: 16 en 0.008 segundos

	DEPARTMENT_NAME	DEPARTMENT_ID
1	Treasury	120
2	Corporate Tax	130
3	Control And Credit	140
4	Shareholder Services	150
5	Benefits	160
6	Manufacturing	170
7	Construction	180
8	Contracting	190
9	Operations	200
10	IT Support	210
11	NOC	220
12	IT Helpdesk	230
13	Government Sales	240
14	Retail Sales	250
15	Recruiting	260
16	Payroll	270

n) You are required to obtain the number of rows in the EMPLOYEES and DEPARTMENTS table as well as the number of records that would be created by a Cartesian product of these two tables. Confirm your results by explicitly counting and multiplying the number of rows present in each of these tables.

```
--n
SELECT COUNT(*), COUNT(UNIQUE E.EMPLOYEE_ID) "COUNT(*)", COUNT(UNIQUE D.DEPARTMENT_ID) "COUNT (*)", 107*27
FROM EMPLOYEES E CROSS JOIN DEPARTMENTS D;
```

Resultado de la Consulta

SQL | Todas las Filas Recuperadas: 1 en 0.003 segundos

	COUNT(*)	COUNT(*)_1	COUNT (*)	107*27
1	2889	107	27	2889

Activity 5:

This practice is intended to give you experience in extracting data from more than one table using the SQL:1999–compliant joins.

1. Write a query for the HR department to produce the addresses of all the departments. Use the LOCATIONS and COUNTRIES tables. Show the location ID, street address, city, state or province, and country in the output. Use a NATURAL JOIN to produce the results.

```
--1
SELECT L.LOCATION_ID, L.STREET_ADDRESS,L.CITY, L.STATE_PROVINCE, C.COUNTRY_NAME
FROM LOCATIONS L NATURAL JOIN COUNTRIES C WHERE STATE_PROVINCE IS NOT NULL;
```

Resultado de la Consulta

SQL | Todas las Filas Recuperadas: 17 en 0.002 segundos

	LOCATION_ID	STREET_ADDRESS	CITY	STATE_PROVINCE	COUNTRY_NAME
1	3000	Murtenstrasse 921	Bern	BE	Switzerland
2	2700	Schwanthalerstr. 7031	Munich	Bavaria	Germany
3	1500	2011 Interiors Blvd	South San Francisco	California	United States of America
4	3200	Mariano Escobedo 9991	Mexico City	Distrito Federal,	Mexico
5	2900	20 Rue des Corps-Saints	Geneva	Geneve	Switzerland
6	2100	1298 Vileparle (E)	Bombay	Maharashtra	India
7	2600	9702 Chester Road	Stretford	Manchester	United Kingdom
8	1600	2007 Zagora St	South Brunswick	New Jersey	United States of America
9	2200	12-98 Victoria Street	Sydney	New South Wales	Australia
10	1800	147 Spadina Ave	Toronto	Ontario	Canada
11	2500	Magdalen Centre, The Oxford Science Park	Oxford	Oxford	United Kingdom
12	2800	Rua Frei Caneca 1360	Sao Paulo	Sao Paulo	Brazil
13	1400	2014 Jabberwocky Rd	Southlake	Texas	United States of America
14	1200	2017 Shinjuku-ku	Tokyo	Tokyo Prefecture	Japan
15	3100	Pieter Breughelstraat 837	Utrecht	Utrecht	Netherlands
16	1700	2004 Charade Rd	Seattle	Washington	United States of America
17	1900	6092 Boxwood St	Whitehorse	Yukon	Canada

2. The HR department needs a report of all employees. Write a query to display the last name, department number, and department name for all the employees.

```
--2

SELECT * FROM EMPLOYEES;
SELECT * FROM DEPARTMENTS;

SELECT LAST_NAME,DEPARTMENT_ID,DEPARTMENT_NAME
FROM EMPLOYEES NATURAL JOIN DEPARTMENTS ;
```

Resultado de la Consulta

SQL | Todas las Filas Recuperadas: 32 en 0.004 segundos

	LAST_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
1	Kochhar	90	Executive
2	De Haan	90	Executive
3	Ernst	60	IT
4	Austin	60	IT
5	Pataballa	60	IT
6	Lorentz	60	IT
7	Faviet	100	Finance
8	Chen	100	Finance
9	Sciarra	100	Finance
10	Urman	100	Finance
11	Popp	100	Finance
12	Khoo	30	Purchasing
13	Baida	30	Purchasing
14	Tobias	30	Purchasing
15	Himuro	30	Purchasing
16	Colmenares	30	Purchasing
17	Bissot	50	Shipping
18	Atkinson	50	Shipping
19	Marlow	50	Shipping
20	Olson	50	Shipping
21	Tucker	80	Sales
22	Bernstein	80	Sales
23	Hall	80	Sales
24	Olsen	80	Sales
25	Cambrault	80	Sales

3. The HR department needs a report of employees in Toronto. Display the last name, job, department number, and the department name for all employees who work in Toronto.

```
--3
SELECT E.LAST_NAME, E.JOB_ID, E.DEPARTMENT_ID, D.DEPARTMENT_NAME
FROM EMPLOYEES E JOIN DEPARTMENTS D
ON(E.DEPARTMENT_ID = D.DEPARTMENT_ID)
JOIN LOCATIONS L
ON (D.LOCATION_ID = L.LOCATION_ID)
WHERE LOWER(L.CITY) = 'toronto';
```

Resultado de la Consulta | Resultado de la Consulta 1

SQL | Todas las Filas Recuperadas: 2 en 0.002 segundos

	LAST_NAME	JOB_ID	DEPARTMENT_ID	DEPARTMENT_NAME
1	Hartstein	MK_MAN	20	Marketing
2	Fay	MK_REP	20	Marketing

4. Create a report to display employees' last name and employee number along with their manager's last name and manager number. Label the columns Employee, Emp, Manager, and Mgr, respectively. Save your SQL statement as lab1204.sql. Run the query.

```
--4
SELECT E.LAST_NAME "EMPLOYEE", E.EMPLOYEE_ID "EMP#",E.LAST_NAME "MANAGER", M.EMPLOYEE_ID "Mgr#"
FROM EMPLOYEES E JOIN EMPLOYEES M ON(E.MANAGER_ID = M.EMPLOYEE_ID);
```

Resultado de la Consulta | Resultado de la Consulta 1

SQL | Se han recuperado 50 filas en 0.003 segundos

	EMPLOYEE	EMP#	MANAGER	Mgr#
1	Kochhar	101	Kochhar	100
2	De Haan	102	De Haan	100
3	Hunold	103	Hunold	102
4	Ernst	104	Ernst	103
5	Austin	105	Austin	103
6	Pataballa	106	Pataballa	103
7	Lorentz	107	Lorentz	103
8	Greenberg	108	Greenberg	101
9	Faviet	109	Faviet	108
10	Chen	110	Chen	108
11	Sciarra	111	Sciarra	108
12	Urman	112	Urman	108
13	Popp	113	Popp	108
14	Raphaely	114	Raphaely	100
15	Khoo	115	Khoo	114
16	Baida	116	Baida	114
17	Tobias	117	Tobias	114
18	Himuro	118	Himuro	114
19	Colmenares	119	Colmenares	114

5. Modify lab1204.sql to display all employees including King, who has no manager. Order the results by the employee number. Save your SQL statement as lab1205.sql. Run the query in lab$_{1}2_{0}5.sql$

```
--5
SELECT E.LAST_NAME "EMPLOYEE", E.EMPLOYEE_ID "EMP#",M.LAST_NAME "Manager", M.EMPLOYEE_ID "Mgr#"
FROM EMPLOYEES E LEFT OUTER JOIN EMPLOYEES M ON (M.MANAGER_ID = M.EMPLOYEE_ID);
```

Resultado de la Consulta | Resultado de la Consulta 1 | Resultado de la Consulta 2

SQL | Todas las Filas Recuperadas: 107 en 0.008 segundos

	EMPLOYEE	EMP#	Manager	Mgr#
1	Abel	174	(null)	(null)
2	Ande	166	(null)	(null)
3	Atkinson	130	(null)	(null)
4	Austin	105	(null)	(null)
5	Baer	204	(null)	(null)
6	Baida	116	(null)	(null)
7	Banda	167	(null)	(null)
8	Bates	172	(null)	(null)
9	Bell	192	(null)	(null)
10	Bernstein	151	(null)	(null)
11	Bissot	129	(null)	(null)
12	Bloom	169	(null)	(null)
13	Bull	185	(null)	(null)
14	Cabrio	187	(null)	(null)
15	Cambrault	148	(null)	(null)
16	Cambrault	154	(null)	(null)
17	Chen	110	(null)	(null)
18	Chung	188	(null)	(null)
19	Colmenares	119	(null)	(null)
20	Davies	142	(null)	(null)
21	De Haan	102	(null)	(null)
22	Dellinger	186	(null)	(null)
23	Dilly	189	(null)	(null)
24	Doran	160	(null)	(null)
25	Ernst	104	(null)	(null)

6. Create a report for the HR department that displays employee last names, department numbers, and all the employees who work in the same department as a given employee. Give each column an appropriate label. Save the script to a file named lab1206.sql.

```
--6
SELECT E.DEPARTMENT_ID DEPARTMENT, E.LAST_NAME EMPLOYEE,C.LAST_NAME COLLEAGUE
FROM EMPLOYEES E
JOIN EMPLOYEES C
ON (E.DEPARTMENT_ID = C.DEPARTMENT_ID)
WHERE E.EMPLOYEE_ID <> C.EMPLOYEE_ID ORDER BY E.DEPARTMENT_ID, E.LAST_NAME, C.LAST_NAME;
```

Resultado de la Consulta | Resultado de la Consulta 1 | Resultado de la Consulta 2 | Resultado de la Consulta 3

SQL | Se han recuperado 50 filas en 0.007 segundos

	DEPARTMENT	EMPLOYEE	COLLEAGUE
1	20	Fay	Hartstein
2	20	Hartstein	Fay
3	30	Baida	Colmenares
4	30	Baida	Himuro
5	30	Baida	Khoo
6	30	Baida	Raphaely
7	30	Baida	Tobias
8	30	Colmenares	Baida
9	30	Colmenares	Himuro
10	30	Colmenares	Khoo
11	30	Colmenares	Raphaely
12	30	Colmenares	Tobias
13	30	Himuro	Baida
14	30	Himuro	Colmenares
15	30	Himuro	Khoo
16	30	Himuro	Raphaely
17	30	Himuro	Tobias
18	30	Khoo	Baida
19	30	Khoo	Colmenares
20	30	Khoo	Himuro
21	30	Khoo	Raphaely
22	30	Khoo	Tobias
23	30	Raphaely	Baida
24	30	Raphaely	Colmenares
25	30	Raphaely	Himuro

7. The HR department needs a report on job grades and salaries. To familiarize yourself with the JOBGRADES table, first show the structure of the JOBGRADES table. Then create a query that displays the name, job, department name, salary, and grade for all employees.

```
--7
SELECT E.LAST_NAME,E.JOB_ID, D.DEPARTMENT_NAME,E.SALARY, J.GRADE "GRADE LEVEL"
FROM EMPLOYEES E
JOIN DEPARTMENTS D
ON (E.DEPARTMENT_ID = D.DEPARTMENT_ID)
JOIN JOB_GRADES J ON (E.SALARY BETWEEN J.LOWEST_SAL AND J.HIGHEST_SAL);

--8
```

Resultado de la Consulta | Resultado de la Consulta 1

SQL | Se han recuperado 50 filas en 0.005 segundos

	LAST_NAME	JOB_ID	DEPARTMENT_NAME	SALARY	GRADE LEVEL
1	Olson	ST_CLERK	Shipping	2100	A
2	Markle	ST_CLERK	Shipping	2200	A
3	Philtanker	ST_CLERK	Shipping	2200	A
4	Landry	ST_CLERK	Shipping	2400	A
5	Gee	ST_CLERK	Shipping	2400	A
6	Colmenares	PU_CLERK	Purchasing	2500	A
7	Marlow	ST_CLERK	Shipping	2500	A
8	Patel	ST_CLERK	Shipping	2500	A
9	Vargas	ST_CLERK	Shipping	2500	A
10	Sullivan	SH_CLERK	Shipping	2500	A
11	Perkins	SH_CLERK	Shipping	2500	A
12	Himuro	PU_CLERK	Purchasing	2600	A
13	Matos	ST_CLERK	Shipping	2600	A
14	OConnell	SH_CLERK	Shipping	2600	A
15	Grant	SH_CLERK	Shipping	2600	A
16	Mikkilineni	ST_CLERK	Shipping	2700	A
17	Seo	ST_CLERK	Shipping	2700	A
18	Tobias	PU_CLERK	Purchasing	2800	A
19	Atkinson	ST_CLERK	Shipping	2800	A
20	Geoni	SH_CLERK	Shipping	2800	A
21	Jones	SH_CLERK	Shipping	2800	A
22	Baida	PU_CLERK	Purchasing	2900	A
23	Rogers	ST_CLERK	Shipping	2900	A
24	Gates	SH_CLERK	Shipping	2900	A
25	Cabrio	SH_CLERK	Shipping	3000	B

8. The HR department wants to determine the names of all the employees who were hired after Davies. Create a query to display the name and hire date of any employee hired after employee Davies

```
--8
SELECT E.LAST_NAME, E.HIRE_DATE FROM  EMPLOYEES E
JOIN EMPLOYEES DAVIES ON  (DAVIES.LAST_NAME = 'Davies')WHERE DAVIES.HIRE_DATE < E.HIRE_DATE;
```

Resultado de la Consulta | Resultado de la Consulta 1

SQL | Se han recuperado 50 filas en 0.002 segundos

	LAST_NAME	HIRE_DATE
1	Kochhar	21/09/05
2	Hunold	03/01/06
3	Ernst	21/05/07
4	Austin	25/06/05
5	Pataballa	05/02/06
6	Lorentz	07/02/07
7	Chen	28/09/05
8	Sciarra	30/09/05
9	Urman	07/03/06
10	Popp	07/12/07
11	Baida	24/12/05
12	Tobias	24/07/05
13	Himuro	15/11/06
14	Colmenares	10/08/07
15	Fripp	10/04/05
16	Vollman	10/10/05
17	Mourgos	16/11/07
18	Nayer	16/07/05
19	Mikkilineni	28/09/06
20	Landry	14/01/07
21	Markle	08/03/08
22	Bissot	20/08/05
23	Atkinson	30/10/05
24	Marlow	16/02/05
25	Olson	10/04/07

9. The HR department needs to find the names and hire dates of all the employees who were hired before their managers, along with their managers' names and hire dates. Save the script to a file named lab109.sql.

```
--9
--parte de script

SELECT E.LAST_NAME, E.HIRE_DATE, M.LAST_NAME, M.HIRE_DATE FROM EMPLOYEES E
JOIN EMPLOYEES M ON (E.MANAGER_ID = M.EMPLOYEE_ID)
WHERE E.HIRE_DATE < M.HIRE_DATE;
```

Resultado de la Consulta | Resultado de la Consulta 1

SQL | Todas las Filas Recuperadas: 37 en 0.003 segundos

	LAST_NAME	HIRE_DATE	LAST_NAME_1	HIRE_DATE_1
1	Kaufling	01/05/03	King	17/06/03
2	Raphaely	07/12/02	King	17/06/03
3	De Haan	13/01/01	King	17/06/03
4	Higgins	07/06/02	Kochhar	21/09/05
5	Baer	07/06/02	Kochhar	21/09/05
6	Mavris	07/06/02	Kochhar	21/09/05
7	Whalen	17/09/03	Kochhar	21/09/05
8	Greenberg	17/08/02	Kochhar	21/09/05
9	Austin	25/06/05	Hunold	03/01/06
10	Faviet	16/08/02	Greenberg	17/08/02
11	Bull	20/02/05	Fripp	10/04/05
12	Sarchand	27/01/04	Fripp	10/04/05
13	Marlow	16/02/05	Fripp	10/04/05
14	Everett	03/03/05	Vollman	10/10/05
15	Bell	04/02/04	Vollman	10/10/05
16	Ladwig	14/07/03	Vollman	10/10/05
17	OConnell	21/06/07	Mourgos	16/11/07
18	Feeney	23/05/06	Mourgos	16/11/07
19	Walsh	24/04/06	Mourgos	16/11/07
20	Vargas	09/07/06	Mourgos	16/11/07
21	Matos	15/03/06	Mourgos	16/11/07
22	Davies	29/01/05	Mourgos	16/11/07
23	Rajs	17/10/03	Mourgos	16/11/07
24	McEwen	01/08/04	Partners	05/01/05
25	Sully	04/03/04	Partners	05/01/05

3 PRE-EVALUATION

Practices pre-Assessment for Database Systems Laboratory II Pre-Assessment PRACTICE 12 carried out by student

1 COMPLIES WITH THE REQUESTED FUNCTIONALITY
YES

4 HAS THE CORRECT INDENTATION
YES

6 HAS AN EASY WAY TO ACCESS THE PROVIDED FILES
YES

7 HAS A REPORT WITH IDC FORMAT
YES

8 REPORT INFORMATION IS FREE OF SPELLING ERRORS
YES

9 DELIVERED IN TIME AND FORM
YES

10 IS FULLY COMPLETED (SPECIFY THE PERCENTAGE COMPLETED)
YES,100 percent

4 Conclusion

This was a very complete practice, always it is important to review and practice

the content of the current chapter of the theory class to improve and learn more. About this chapter 6 there were many things that complicated me when I worked on it in this practice, you must always be attentive to the sentences that you are writing so as not to make mistakes, the JOINS are important to be able to make more advanced queries and join several tables.