Practice 12: Displaying Data from Multiple Tables

Jesús Manuel Juárez Pasillas

November 04, 2021

1 Introduction

The unions between tables allow us to consult data from different tables which share one or more values for one or more columns, which helps us to make more complete queries where the information obtained not only comes from a table but is also a complement of a set of tables which contain values of importance for what is being queried.

2 Development

2.1 Activity 1:

Read all the choices carefully because there might be more than one correct answer. Choose all the correct answers for each question.

1. The EMPLOYEES and DEPARTMENTS tables have two identically named columns:
   DEPARTMENT_ID and MANAGER_ID. Which of these statements joins these tables based only on common DEPARTMENT_ID values?

   A. SELECT * FROM EMPLOYEES NATURAL JOIN DEPARTMENTS;

   B. SELECT * FROM EMPLOYEES E NATURAL JOIN DEPARTMENTS D ON E.DEPARTMENT_ID=D.DEPARTMENT_ID;

   C. SELECT * FROM EMPLOYEES NATURAL JOIN DEPARTMENTS USING (DEPARTMENT_ID);

   D. **None of the above**

   To join the two tables with the departments_id field, the natural join cannot be used because it joins the tables by all the columns that match it, for this it is necessary to use a join with its using clause and the column that will be used to join the boards.

2. The EMPLOYEES and DEPARTMENTS tables have two identically named columns:
   DEPARTMENT_ID and MANAGER_ID. Which statements join these tables based on both column values?

   A. **SELECT * FROM EMPLOYEES NATURAL JOIN DEPARTMENTS;**

   B. SELECT * FROM EMPLOYEES JOIN DEPARTMENTS USING (DEPARTMENT_ID,MANAGER_ID);

   C. **SELECT * FROM EMPLOYEES E JOIN DEPARTMENTS D ON E.DEPARTMENT_ID=D.DEPARTMENT_ID AND E.MANAGER_ID=D.MANAGER_ID;**

   D. None of the above

   To join the tables with the two said columns, it is easier to use a natural join since it is only these two columns that coincide, although a join on can also be used where it is evaluated that the columns of each table coincide with the columns of the other table.

3. Which join is performed by the following query?
   SELECT E.JOB_ID,J.JOB_ID FROM EMPLOYEES E
   JOIN JOBS J ON (E.SALARY ¡ J.MAX_SALARY);

   A. Equijoin

   B. **Nonequijoin**

   C. Cross join

   D. Outer join

   As the union of the two tables is not given by equality in the fields, but it only proposes a condition with which the tables are going to be joined.

4. Which of the following statements are syntactically correct?

   A. **SELECT * FROM EMPLOYEES E JOIN DEPARTMENTS D USING (DEPARTMENT_ID);**

   B. SELECT * FROM EMPLOYEES JOIN DEPARTMENTS D USING (D.DEPARTMENT_ID);

   C. SELECT D.DEPARTMENT_ID FROM EMPLOYEES JOIN DEPARTMENTS D USING (DEPARTMENT_ID);

   D. None of the above

   Section A makes correct use of the syntax of a join using, the others make some error in their syntax.

5. Which of the following statements are syntactically correct?

   A. SELECT E.EMPLOYEE_ID, J.JOB_ID PREVIOUS_JOB, E.JOB_ID CURRENT_JOB FROM JOB_HISTORY J CROSS JOIN EMPLOYEES E ON (J.START_DATE=E.HIRE_DATE);

   B. SELECT E.EMPLOYEE_ID, J.JOB_ID PREVIOUS_JOB, E.JOB_ID CURRENT_JOB FROM JOB_HISTORY J JOIN EMPLOYEES E ON (J.START_DATE=E.HIRE_DATE);

   C. SELECT E.EMPLOYEE_ID, J.JOB_ID PREVIOUS_JOB, E.JOB_ID CURRENT_JOB FROM JOB_HISTORY J OUTER JOIN EMPLOYEES E ON (J.START_DATE=E.HIRE_DATE);

   D. **None of the above**

   No statement makes correct use of the joins it uses. Only section B does not return an error when executing the sentence, however, it makes use of the on where it compares dates, which makes it compare data from different employees and not from the same employee.

6. Choose one correct statement regarding the following query:
   SELECT * FROM EMPLOYEES E JOIN DEPARTMENTS D ON (D.DEPARTMENT_ID=E.DEPARTMENT_ID) JOIN LOCATIONS L ON (L.LOCATION_ID =D.LOCATION_ID);

   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

   The join on can be used to join the tables that you want, in which in the on clause equality conditions between columns are met to avoid a Cartesian product.

7. How many rows are returned after executing the following statement?
   SELECT * FROM REGIONS R1 JOIN REGIONS R2 ON (R1.REGION_ID = LENGTH(R2.REGION_NAME)/2);

   A. 2

   B. **3**

   C. 4

   D. None of the above

   It returns 3 values, since when doing the operation of the number of characters of the name of the region divided by 2, a number is produced which is equal to the id of a region.

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

    A. No rows in the LOCATIONS table have the COUNTRY_ID values returned.

    B. No rows in the COUNTRIES table have the COUNTRY_ID values returned.

    C. The rows returned represent the COUNTRY_ID values for all the rows in the LOCATIONS table.

    D. **None of the above**

    The rows that the statement returns are the rows from the COUNTRIES table with a country_id value that is not found in the LOCATIONS table.

9. Which of the following statements are syntactically correct?

    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

    Even though it only shows one column, this query makes good use of rigth outer join as it uses the job$_idvaluesofeachtabletojointhem.$

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 fina results set?

    A. **100 rows**

    B. 4 rows

    C. 25 rows

    D. None of the above

    It will return 100 rows, since for each row of the REGIONS table it is joined with each row of the COUNTRIES table.

2.2 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?
  - **Yes, table joins return rows that the group functions use and thereby return a value.**
- 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, it is necessary to specify from which table and which column you want to show, since there may be columns with the same names and that would give an error.**
- 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?
  - **When using a natural join it is not possible to join two tables only with a column with equal values if they have more columns with equal values, for this the join using is used.**
- 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?
  - **If it is possible to do this with the join on where in the on clause the condition that they have to meet to join the rows of the tables is made, a condition is made where the value of one column of the first table is equal to the value of another column of the second table.**
- 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 EMPLOYEE_ID, LAST_NAME, and REGION_NAME values for each employee by joining the EMPLOYEE_ID and REGION_ID columns in a round-robin manner?
  - **Yes and no, if the table can be joined, however, we would not obtain any response value since the values of employee_id and region_id never coincide so it would not do what you want and it is better to use a group function.**
- You are required to retrieve a list of DEPARTMENT_NAME and LAST_NAME values for all departments, including those that currently

have no employees assigned to them. In such cases the string ‘No Employees’ should be displayed a the LAST_NAME column value. Can this be done using joins?

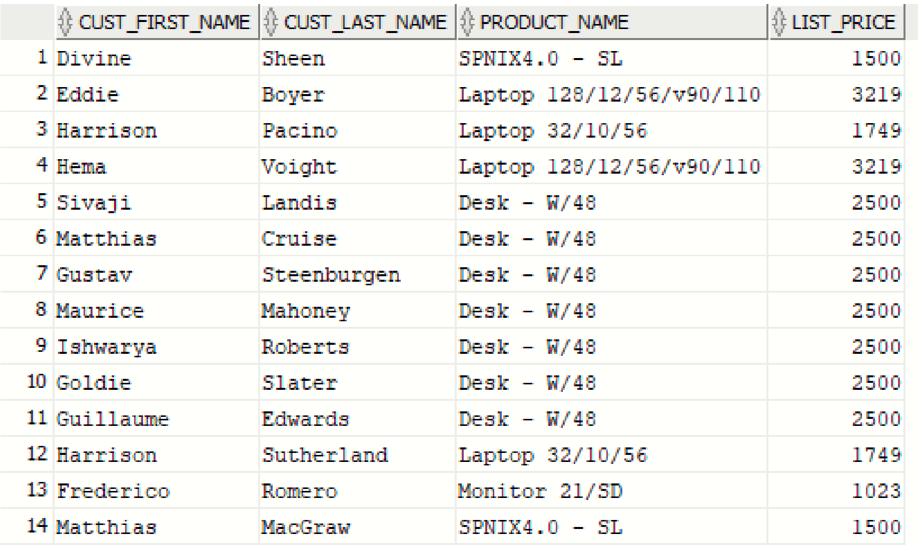
- **Yes, making use of the right outer join... on and with the nvl function in which the last name of the employee is contained as arguments, if it is null then it returns the specified string.**

2.3 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.

- **Select c.cust_first_name, c.cust_last_name, pf.product_name, pf.list_price from Customers c join Orders o on (c.customer_id=o.customer_id) join Order_Items oi on(o.order_id=oi.order_id) join Product_Information pf on (oi.product_id=pf.product_id and pf.list_price > 1000);**



	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

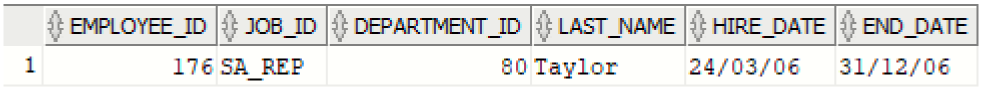
Figure 1: Consult the oe scheme.

2.4 Activity 4:

This exercise must be performed in the HR schema.

a) You are required to describe the tables and fetch the EMPLOYEE_ID, JOB_ID, DEPARTMENT_ID, LAST_NAME, HIRE_DATE, and END_DATE values for all rows retrieved using a pure natural join.

- **Select employee_id, job_id, department_id, emp.last_name, emp.hire_date, jh.end_date from Employees emp natural join Job_History jh;**

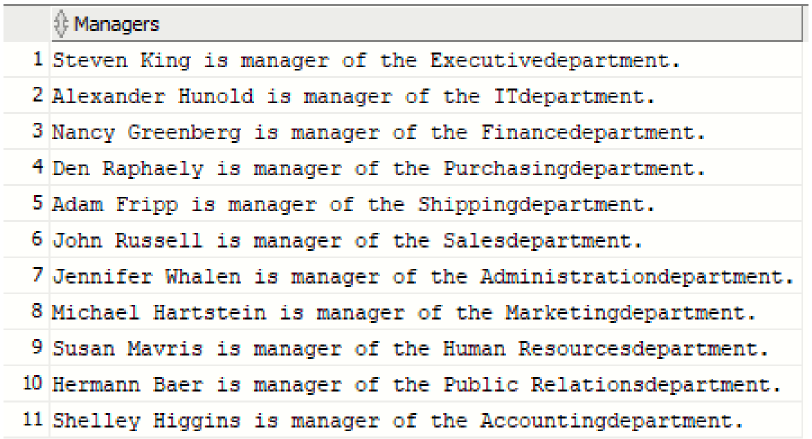


	EMPLOYEE_ID	JOB_ID	DEPARTMENT_ID	LAST_NAME	HIRE_DATE	END_DATE
1	176	SA_REP	80	Taylor	24/03/06	31/12/06

Figure 2: Employees natural join Job History.

b) You are required to produce a report with one column aliased as Managers. Each row must contain a sentence of the format FIRST_NAME LAST_NAME is manager of the DEPARTMENT_NAME department.

- Select e.first_name ||’ ’ || e.last_name || ’ is manager of the ’ || d.department_name || ’department.’ ”Managers” from Employees e join Departments d on (e.employee_id = d.manager_id);

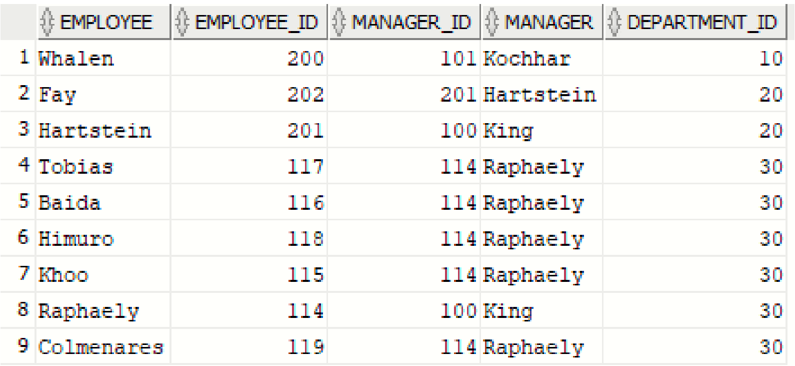


	Managers
1	Steven King is manager of the Executivedepartment.
2	Alexander Hunold is manager of the ITdepartment.
3	Nancy Greenberg is manager of the Financedepartment.
4	Den Raphaely is manager of the Purchasingdepartment.
5	Adam Fripp is manager of the Shippingdepartment.
6	John Russell is manager of the Salesdepartment.
7	Jennifer Whalen is manager of the Administrationdepartment.
8	Michael Hartstein is manager of the Marketingdepartment.
9	Susan Mavris is manager of the Human Resourcesdepartment.
10	Hermann Baer is manager of the Public Relationsdepartment.
11	Shelley Higgins is manager of the Accountingdepartment.

Figure 3: Managers.

c) For each row in the EMPLOYEES table the MANAGER_ID column stores the EMPLOYEE_ID of every employee’s manager. Using a self-join on the EMPLOYEES table, you are required to retrieve the employee’s LAST_NAME, EMPLOYEE_ID, manager's LAST_NAME, and employee's DEPARTMENT_ID for the rows with DEPARTMENT_ID values of 10, 20, or 30.

- **Select e.last_name employee, e.employee_id,e.manager_id, m.last_name as "MANAGER", e.department_id from Employees e join Employees m on (e.manager_id=m.employee_id) where e.department_id in (10,20,30) order by department_id;**

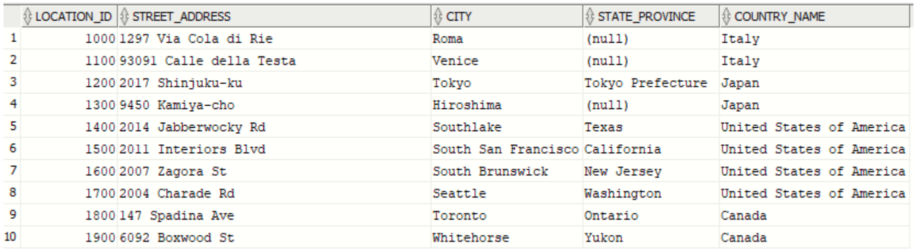


	EMPLOYEE	EMPLOYEE_ID	MANAGER_ID	MANAGER	DEPARTMENT_ID
1	Whalen	200	101	Kochhar	10
2	Fay	202	201	Hartstein	20
3	Hartstein	201	100	King	20
4	Tobias	117	114	Raphaely	30
5	Baida	116	114	Raphaely	30
6	Himuro	118	114	Raphaely	30
7	Khoo	115	114	Raphaely	30
8	Raphaely	114	100	King	30
9	Colmenares	119	114	Raphaely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

Figure 4: Employee and Manager.

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

- **Select l.location_id, l.street_address, l.city, l.state_province, c.country_name from Locations l natural join Countries c;**

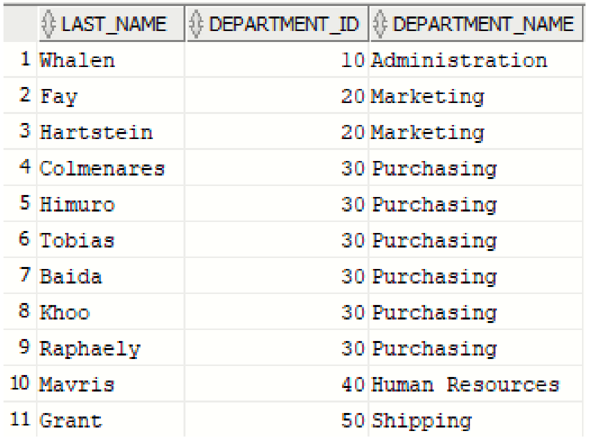


	LOCATION_ID	STREET_ADDRESS	CITY	STATE_PROVINCE	COUNTRY_NAME
1	1000	1297 Via Cola di Rie	Roma	(null)	Italy
2	1100	93091 Calle della Testa	Venice	(null)	Italy
3	1200	2017 Shinjuku-ku	Tokyo	Tokyo Prefecture	Japan
4	1300	9450 Kamiya-cho	Hiroshima	(null)	Japan
5	1400	2014 Jabberwocky Rd	Southlake	Texas	United 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

Figure 5: Address.

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.

- **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);**

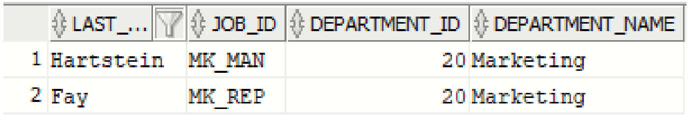


	LAST_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
1	Whalen	10	Administration
2	Fay	20	Marketing
3	Hartstein	20	Marketing
4	Colmenares	30	Purchasing
5	Himuro	30	Purchasing
6	Tobias	30	Purchasing
7	Baida	30	Purchasing
8	Khoo	30	Purchasing
9	Raphaely	30	Purchasing
10	Mavris	40	Human Resources
11	Grant	50	Shipping

Figure 6: Department in which each employee works.

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’:

- **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 and l.city=’&CITY’);**

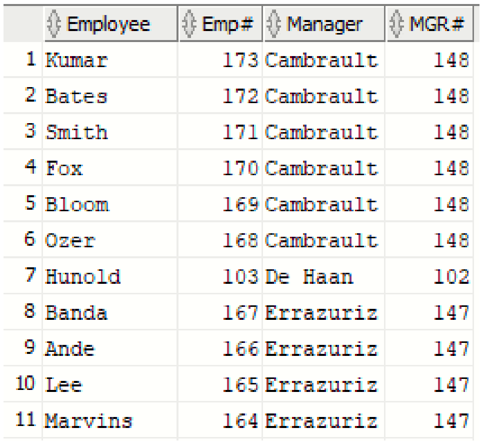


	LAST_...	JOB_ID	DEPARTMENT_ID	DEPARTMENT_NAME
1	Hartstein	MK_MAN	20	Marketing
2	Fay	MK_REP	20	Marketing

Figure 7: Employees who work in a city.

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.

- **Select e.last_name ”Employee”, e.employee_id ”Emp#”, m.last_name ”Manager”, m.employee_id ”MGR#” from Employees e join Employees m on (e.manager_id=m.employee_id);**

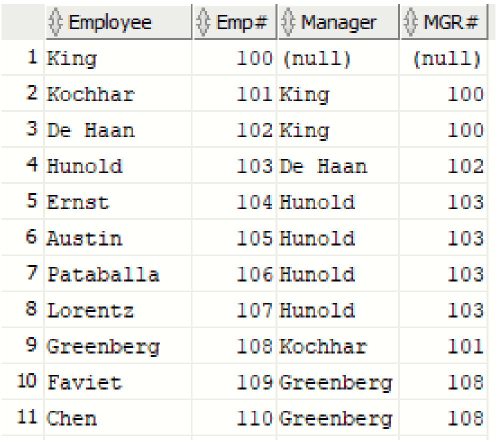


	Employee	Emp#	Manager	MGR#
1	Kumar	173	Cambrault	148
2	Bates	172	Cambrault	148
3	Smith	171	Cambrault	148
4	Fox	170	Cambrault	148
5	Bloom	169	Cambrault	148
6	Ozer	168	Cambrault	148
7	Hunold	103	De Haan	102
8	Banda	167	Errazuriz	147
9	Ande	166	Errazuriz	147
10	Lee	165	Errazuriz	147
11	Marvins	164	Errazuriz	147

Figure 8: Employee-Manager.

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

- **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 (e.manager_id=m.employee_id) order by e.employee_id;**

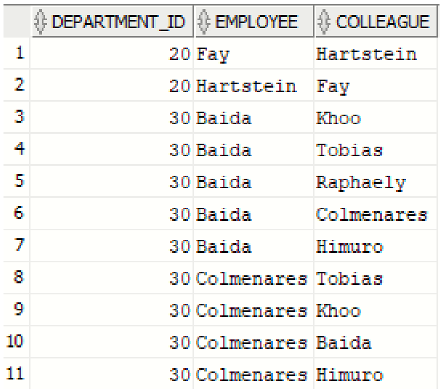


	Employee	Emp#	Manager	MGR#
1	King	100	(null)	(null)
2	Kochhar	101	King	100
3	De Haan	102	King	100
4	Hunold	103	De Haan	102
5	Ernst	104	Hunold	103
6	Austin	105	Hunold	103
7	Pataballa	106	Hunold	103
8	Lorentz	107	Hunold	103
9	Greenberg	108	Kochhar	101
10	Faviet	109	Greenberg	108
11	Chen	110	Greenberg	108

Figure 9: Employee-Manager including those who have no manager.

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.

- **Select e.department_id, e.last_name employee, m.last_name colleague from Employees e join Employees m on (e.department_id=m.department_id and e.employee_id != m.employee_id) order by e.department_id,e.last_name;**

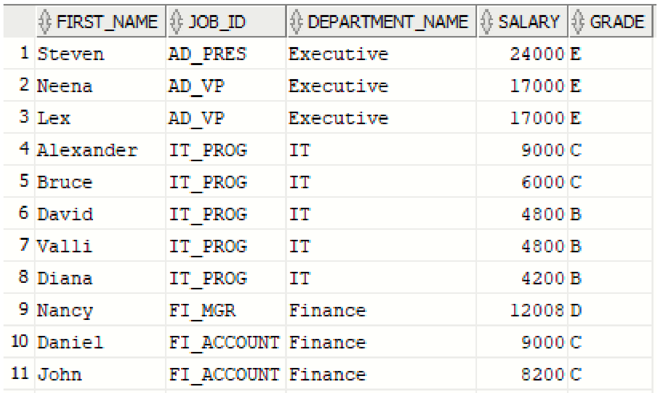


	DEPARTMENT_ID	EMPLOYEE	COLLEAGUE
1	20	Fay	Hartstein
2	20	Hartstein	Fay
3	30	Baida	Khoo
4	30	Baida	Tobias
5	30	Baida	Raphaely
6	30	Baida	Colmenares
7	30	Baida	Himuro
8	30	Colmenares	Tobias
9	30	Colmenares	Khoo
10	30	Colmenares	Baida
11	30	Colmenares	Himuro

Figure 10: Employees along with their colleagues.

j) To familiarize yourself with the JOB_GRADES table, first show the structure of the JOB_GRADES table. Then create a query that displays the name, job, department name, salary, and grade for all employees.

- **Select e.first_name, e.job_id, d.department_name, e.salary, g.grade from Employees e join Departments d on (e.department_id=d.department_id) join Job_Grades g on (e.salary >= g.lowest_sal and e.salary ¡= g.highest_sal) order by e.employee_id;**

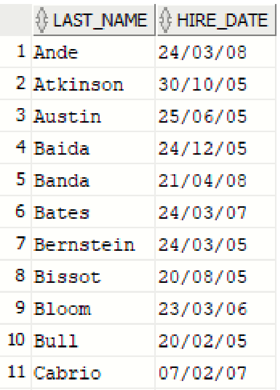


	FIRST_NAME	JOB_ID	DEPARTMENT_NAME	SALARY	GRADE
1	Steven	AD_PRES	Executive	24000	E
2	Neena	AD_VP	Executive	17000	E
3	Lex	AD_VP	Executive	17000	E
4	Alexander	IT_PROG	IT	9000	C
5	Bruce	IT_PROG	IT	6000	C
6	David	IT_PROG	IT	4800	B
7	Valli	IT_PROG	IT	4800	B
8	Diana	IT_PROG	IT	4200	B
9	Nancy	FI_MGR	Finance	12008	D
10	Daniel	FI_ACCOUNT	Finance	9000	C
11	John	FI_ACCOUNT	Finance	8200	C

Figure 11: Job Grades and salaries.

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’:

- **Select last_name, hire_date from Employees where hire_date > (Select hire_date from Employees where last_name = ’&NAME’) order by last_name;**

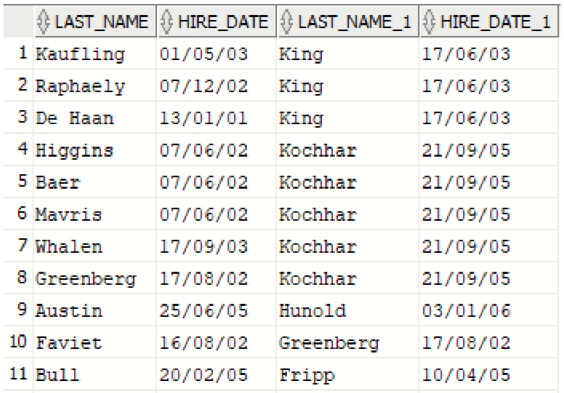


	LAST_NAME	HIRE_DATE
1	Ande	24/03/08
2	Atkinson	30/10/05
3	Austin	25/06/05
4	Baida	24/12/05
5	Banda	21/04/08
6	Bates	24/03/07
7	Bernstein	24/03/05
8	Bissot	20/08/05
9	Bloom	23/03/06
10	Bull	20/02/05
11	Cabrio	07/02/07

Figure 12: Employee hired after an employee entered.

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.

- **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 e.hire_date < m.hire_date);**

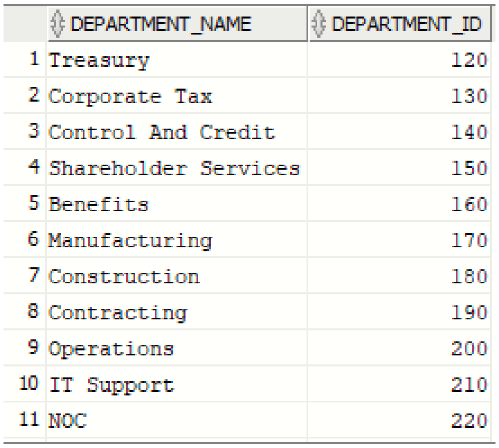


	LAST_NAME	HIRE_DATE	LAST_NAME_1	HIRE_DATE_1
1	Kaufling	01/05/03	King	17/06/03
2	Raphaely	07/12/02	King	17/06/03
3	De Haan	13/01/01	King	17/06/03
4	Higgins	07/06/02	Kochhar	21/09/05
5	Baer	07/06/02	Kochhar	21/09/05
6	Mavris	07/06/02	Kochhar	21/09/05
7	Whalen	17/09/03	Kochhar	21/09/05
8	Greenberg	17/08/02	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

Figure 13: Employees who were hired before their managers.

m) You are required to retrieve the DEPARTMENT_NAME and DEPARTMENT_ID values for those departments to which no employees are currently assigned.

- **Select d.department_name, d.department_id from Departments d left outer join Employees e on (d.department_id=e.department_id) where e.department_id is null;**

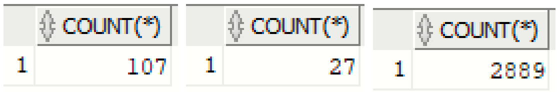


	DEPARTMENT_NAME	DEPARTMENT_ID
1	Treasury	120
2	Corporate Tax	130
3	Control And Credit	140
4	Shareholder Services	150
5	Benefits	160
6	Manufacturing	170
7	Construction	180
8	Contracting	190
9	Operations	200
10	IT Support	210
11	NOC	220

Figure 14: Departments to which no employees are currently assigned.

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.

- **Select count(*) from Employees;**
- **Select count(*) from Departments;**
- **Select count(*) from Employees cross join Departments;**



	COUNT(*)
1	107

	COUNT(*)
1	27

	COUNT(*)
1	2889

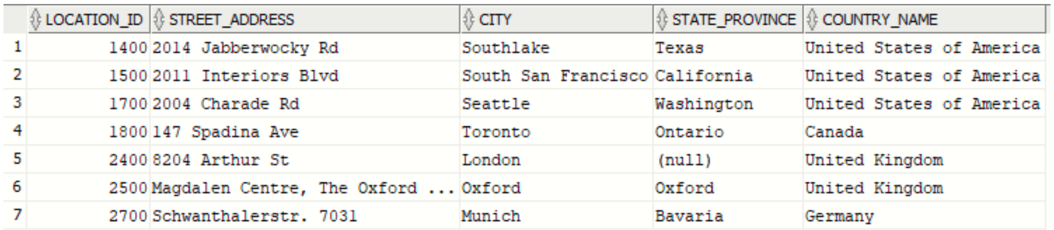
Figure 15: Employees and Departments.

2.5 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. Show the location ID, street address, city, state or province, and country in the output.

   - **Select distinct location_id,street_address,city,state_province, country_name from Departments natural join Locations natural join Countries order by location_id;**

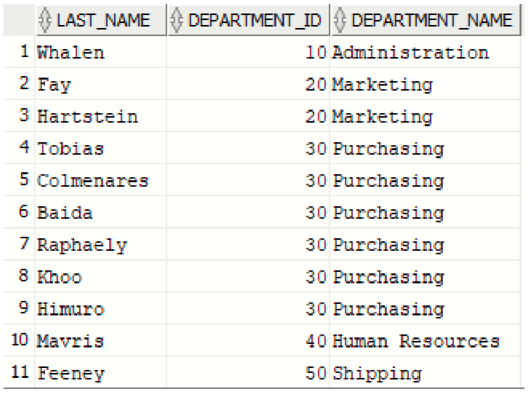


	LOCATION_ID	STREET_ADDRESS	CITY	STATE_PROVINCE	COUNTRY_NAME
1	1400	2014 Jabberwocky Rd	Southlake	Texas	United States of America
2	1500	2011 Interiors Blvd	South San Francisco	California	United States of America
3	1700	2004 Charade Rd	Seattle	Washington	United States of America
4	1800	147 Spadina Ave	Toronto	Ontario	Canada
5	2400	8204 Arthur St	London	(null)	United Kingdom
6	2500	Magdalen Centre, The Oxford ...	Oxford	Oxford	United Kingdom
7	2700	Schwanthalerstr. 7031	Munich	Bavaria	Germany

Figure 16: Departments addresses.

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.

   - **Select e.last_name, d.department_id, d.department_name from Employees e join Departments d on (e.department_id = d.department_id);**

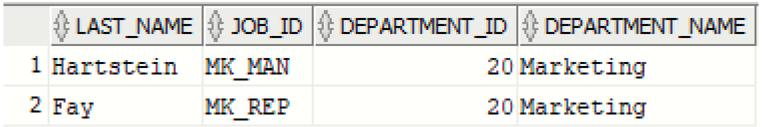


	LAST_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
1	Whalen	10	Administration
2	Fay	20	Marketing
3	Hartstein	20	Marketing
4	Tobias	30	Purchasing
5	Colmenares	30	Purchasing
6	Baida	30	Purchasing
7	Raphaely	30	Purchasing
8	Khoo	30	Purchasing
9	Himuro	30	Purchasing
10	Mavris	40	Human Resources
11	Feeney	50	Shipping

Figure 17: Employee and Department name.

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.

   - **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 and l.city='Toronto');**

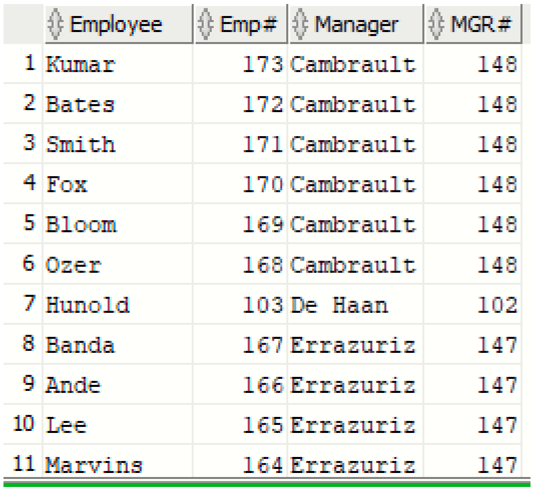


	LAST_NAME	JOB_ID	DEPARTMENT_ID	DEPARTMENT_NAME
1	Hartstein	MK_MAN	20	Marketing
2	Fay	MK_REP	20	Marketing

Figure 18: Employees who work in Toronto.

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 lab_12_04.sql. Run the query.

   - **The file is located inside the file called "Plugins.zip".**

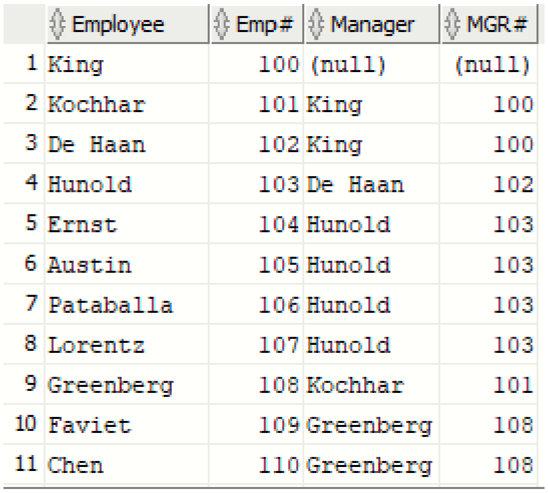


	Employee	Emp#	Manager	MGR#
1	Kumar	173	Cambrault	148
2	Bates	172	Cambrault	148
3	Smith	171	Cambrault	148
4	Fox	170	Cambrault	148
5	Bloom	169	Cambrault	148
6	Ozer	168	Cambrault	148
7	Hunold	103	De Haan	102
8	Banda	167	Errazuriz	147
9	Ande	166	Errazuriz	147
10	Lee	165	Errazuriz	147
11	Marvins	164	Errazuriz	147

Figure 19: Employee, Emp#, Manager, and Mgr#.

5. Modify lab_12_04.sql to display all employees including King, who has no manager. Order the results by the employee number. Save your SQL statement as lab_12_05.sql. Run the query in lab_12_05.sql.

   - **The file is located inside the file called "Plugins.zip".**

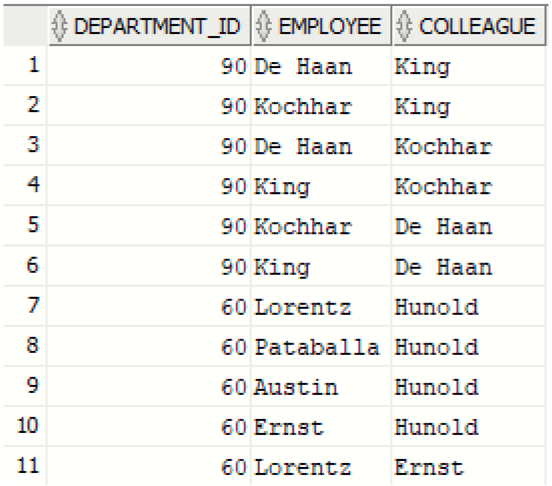


	Employee	Emp#	Manager	MGR#
1	King	100	(null)	(null)
2	Kochhar	101	King	100
3	De Haan	102	King	100
4	Hunold	103	De Haan	102
5	Ernst	104	Hunold	103
6	Austin	105	Hunold	103
7	Pataballa	106	Hunold	103
8	Lorentz	107	Hunold	103
9	Greenberg	108	Kochhar	101
10	Faviet	109	Greenberg	108
11	Chen	110	Greenberg	108

Figure 20: Previous query modified..

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 lab_12_06.sql.

   - **The file is located inside the file called "Plugins.zip".**



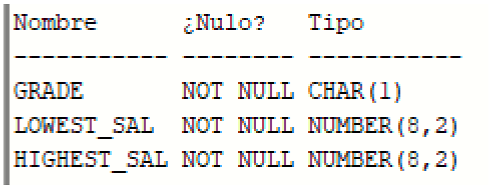
	DEPARTMENT_ID	EMPLOYEE	COLLEAGUE
1	90	De Haan	King
2	90	Kochhar	King
3	90	De Haan	Kochhar
4	90	King	Kochhar
5	90	Kochhar	De Haan
6	90	King	De Haan
7	60	Lorentz	Hunold
8	60	Pataballa	Hunold
9	60	Austin	Hunold
10	60	Ernst	Hunold
11	60	Lorentz	Ernst

Figure 21: Employees who work in the same department.

7. The HR department needs a report on job grades and salaries. To familiarize yourself with the JOB_GRADES table, first show the structure of

the JOB_GRADES table. Then create a query that displays the name, job, department name, salary, and grade for all employees.

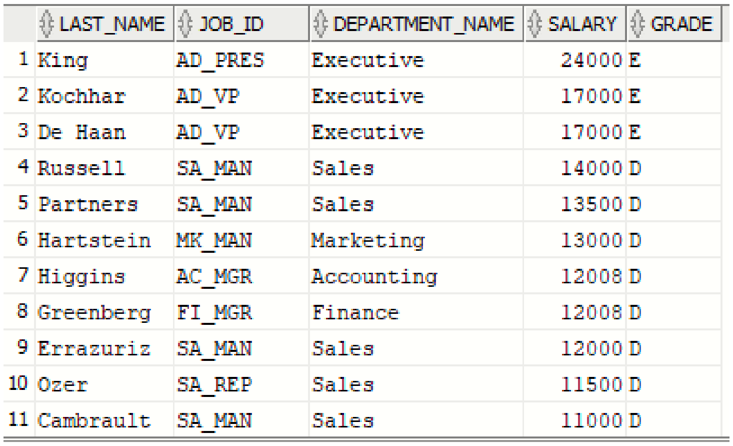
- **describe Job_Grades;**



```
Nombre      ¿Nulo?   Tipo
----------- -------- -----------
GRADE       NOT NULL CHAR(1)
LOWEST_SAL  NOT NULL NUMBER(8,2)
HIGHEST_SAL NOT NULL NUMBER(8,2)
```

Figure 22: Describe Job_Grades.

- **Select e.last_name, e.job_id, d.department_name, e.salary, g.grade from Employees e join Departments d on (e.department_id=d.department_id) join Job_Grades g on (e.salary >= g.lowest_sal and e.salary <= g.highest_sal);**

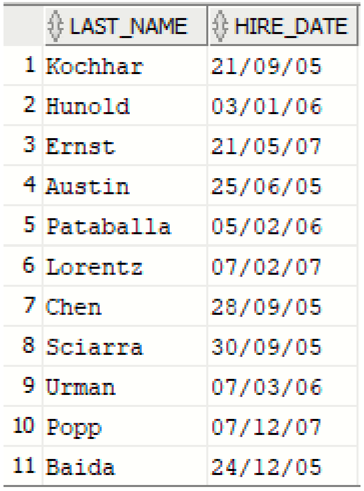


	LAST_NAME	JOB_ID	DEPARTMENT_NAME	SALARY	GRADE
1	King	AD_PRES	Executive	24000	E
2	Kochhar	AD_VP	Executive	17000	E
3	De Haan	AD_VP	Executive	17000	E
4	Russell	SA_MAN	Sales	14000	D
5	Partners	SA_MAN	Sales	13500	D
6	Hartstein	MK_MAN	Marketing	13000	D
7	Higgins	AC_MGR	Accounting	12008	D
8	Greenberg	FI_MGR	Finance	12008	D
9	Errazuriz	SA_MAN	Sales	12000	D
10	Ozer	SA_REP	Sales	11500	D
11	Cambrault	SA_MAN	Sales	11000	D

Figure 23: Grade for all employees.

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.

    - **Select last_name, hire_date from Employees where hire_date > (Select hire_date from Employees where last_name = 'Davies');**

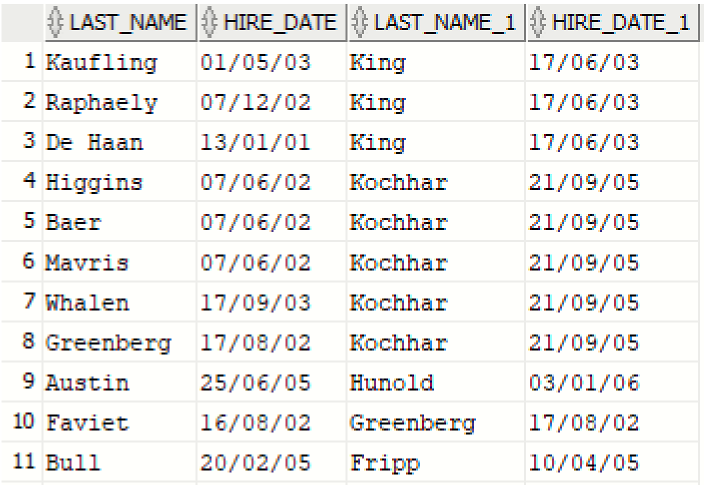


	LAST_NAME	HIRE_DATE
1	Kochhar	21/09/05
2	Hunold	03/01/06
3	Ernst	21/05/07
4	Austin	25/06/05
5	Pataballa	05/02/06
6	Lorentz	07/02/07
7	Chen	28/09/05
8	Sciarra	30/09/05
9	Urman	07/03/06
10	Popp	07/12/07
11	Baida	24/12/05

Figure 24: Employees who were hired after Davies.

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 lab_12_09.sql.

   - **The file is located inside the file called "Plugins.zip".**



	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

Figure 25: Employees who were hired before their managers.

3 Pre-Assessment:

- Practices pre-Assessment for Database Systems Laboratory II

Practice	Pre-Assessment
COMPLIES WITH THE REQUESTED FUNCTIONALITY	X
HAS THE CORRECT INDENTATION	X
HAS AN EASY WAY TO ACCESS THE PROVIDED FILES	X
HAS A REPORT WITH IDC FORMAT	X
REPORT INFORMATION IS FREE OF SPELLING ERRORS	X
DELIVERED IN TIME AND FORM	X
IS FULLY COMPLETED (SPECIFY THE PERCENTAGE COMPLETED)	100%

Table 1: Pre-Assessment.

4 Conclusion:

The unions between the tables are very important, as well as each of the ways in which we can join them since with each form we can obtain a different result due to the way in which these tables are being joined. Without these unions it would be impossible to carry out queries where data has to be extracted from more than one table in which certain fields are obtained from each table. With these unions and their different ways of using them, they allow us to carry out a wide variety of queries where we want to extract data of interest from certain tables of interest.