



Autonomous University of Zacatecas

ACADEMIC UNIT OF ELECTRICAL ENGINEERING

Software Engineering Academic Program

Group: 5B - Semester: 2022-5^o

Practice Number: 14

Using Set Operators to Solve Problems

DATE: 15/NOVEMBER/2022

Professor:

Aldonso Becerra Sánchez.

Student:

Cristian Omar Alvarado Rodríguez.

Contents

1	Introduction	3
2	Practice objective	5
3	Developing	5
4	Pre-assessment	20
5	Conclusion	20

Using Set Operators to Solve Problems

November 15th, 2022

1 Introduction

The SET Operators.

SET operators combine the results of queries from two or more components into one result.

Queries that contain SET operators are called compound queries.

All SET operators have the same precedence. If an SQL statement contains multiple SET operators, the Oracle Server will evaluate them from left (top) to right (bottom) if there are no parentheses that explicitly specify another order. You must use parentheses to explicitly specify the order of evaluation in queries that use the INTERSECT operator with other SET operators.

The UNION Operator:

The UNION operator returns all the rows selected by any two queries. Use the UNION operator to return all rows from multiple tables and remove duplicate rows.

Instructions

- The number of columns and the data types of the columns being selected must be identical in all SELECT statements used in the query. The column names do not have to be identical.
- UNION operates on all the columns that are being selected.

- NULL values are not ignored during the duplicate check.
- The IN operator has a higher priority than the UNION operator.
- By default, the output is sorted in ascending order of the first column of the SELECT clause.

The UNION operator removes duplicate records.

The UNION ALL Operator:

Use the UNION ALL operator to return all rows from multiple queries.

Instructions.

- Unlike UNION, duplicate rows are not removed and the output is not sorted by default.
- The DISTINCT keyword cannot be used.

Note: With the exception of the above, the instructions for UNION and UNION ALL are the same.

The INTERSECT Operator:

Use the INTERSECT operator to return all rows common to multiple queries.

Instructions

- The number of columns and the data types of the columns being selected by the SELECT statements in the queries must be identical in all SELECT statements used in the query. The column names do not have to be identical.
- The inversion of the order of the tables that are crossed does not alter the result.
- INTERSECT does not ignore NULL values.

The MINUS Operator:

Use the MINUS operator to return the rows returned by the first query that are not present in the second (the first SELECT statement minus (MINUS) the second SELECT statement).

Instructions.

- The number of columns and the data types of the columns being selected by the SELECT statements in the queries must be identical in all SELECT statements used in the query. The column names do

not have to be identical.

- For the MINUS operator to work, all columns in the WHERE clause must be in the SELECT clause.

2 Practice objective

Use SQL SELECT statements for retrieving data from several sources using set operators

3 Developing

Activity 1: 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.

DESCRIBE THE SET OPERATORS.

1. Which of these set operators will not sort the rows? (Choose the best answer.)

- A) INTERSECT.
- B) MINUS.
- C) UNION.
- D) UNION ALL.

Explanation: D. The UNION ALL operator does not perform automatic ordering.

2. Which of these operators will remove duplicate rows from the final result? (Choose all that apply.)

- A) INTERSECT.
- B) MINUS.

C) UNION.

D) UNION ALL.

Explanation: A, B, C. INTERSECT, MINUS and UNION all remove duplicate rows. **USE A SET OPERATOR TO COMBINE MULTIPLE QUERIES INTO A SINGLE QUERY.**

3. If a compound query contains both a MINUS and an INTERSECT operator, which will be applied first? (Choose the best answer.)

A) The INTERSECT, because INTERSECT has higher precedence than MINUS.

B) The MINUS, because MINUS has a higher precedence than INTERSECT.

C) The precedence is determined by the order in which they are specified.

D) It is not possible for a compound query to include both MINUS and INTERSECT.

Explanation: C. All set operators have equal precedence, so the precedence is determined by the sequence in which they occur

4. There are four rows in the REGIONS table. Consider the following statements and choose how many rows will be returned for each: 0, 4, 8, or 16.

A) select * from regions union select * from regions;

B) select * from regions union all select * from regions;

C) select * from regions minus select * from regions;

D) select * from regions intersect select * from regions;

Explanation: A = 4, B = 8, C = 0 and D = 4

5. Consider this compound query:

`select empno, hired from emp union all select emp_id, hired, fired from ex_emp;`

The columns `EMP.EMPNO` and `EX_EMP.EMP_ID` are integer;

the column `EMP.HIRED` is timestamp; the columns `EX_EMP.HIRED` and `EX_EMP.FIRED` are date. Why will the statement fail? (Choose the best answer.)

- A) Because the columns `EMPNO` and `EMP_ID` have different names,
- B) Because the columns `EMP.HIRED` and `EX_EMP.HIRED` are different data types.
- C) Because there are two columns in the first query and three columns in the second query.
- D) For all the reasons above.
- E) The query will succeed.

Explanation: C. Every query in a compound query must return the same number of columns.

CONTROL THE ORDER OF ROWS RETURNED.

6. Which line of this statement will cause it to fail? (Choose the best answer.)

- A) `select ename, hired from current_staff.`
- B) `order by ename.`
- C) `minus.`
- D) `select ename, hired from current staff.`
- E) `where deptno = 10`
- F) `order by ename;`

Explanation: B. You can't use `ORDER BY` for a query in a compound query you can only put a single `ORDER BY` clause at the end.

7. Study this statement:

select ename from emp union all select ename from ex_emp;

In what order will the rows be returned? (Choose the best answer.

- A) The rows from each table will be grouped and within each group will be sorted on ENAME.
- B) The rows from each table will be grouped but not sorted.
- C) The rows will not be grouped but will all be sorted on ENAME.
- D) The rows will be neither grouped nor sorted.

Explanation: B. The rows from each query will be together but there will be no sorting operators.

Activity 2: Propose an answer to the following issues:

- How can you present several tables with similar data as one table?

R = Performing a set operation (Using the INTERSECT, MINUS, UNION, or UNION ALL operators to join queries).

- Are there performance issues with compound queries?.

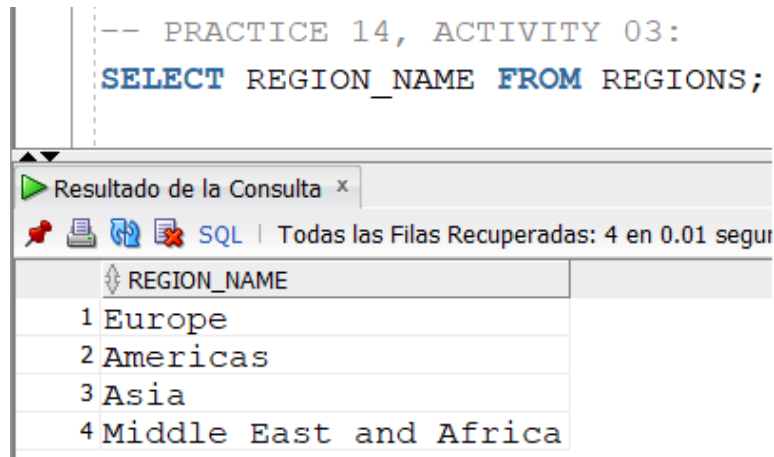
R = Oracle does have problems with the performance of compound queries since sometimes many joins of multiple tables are performed.

Activity 3: This exercise must be performed using HR schema.

a) In this exercise, you will see the effect of the set operators.

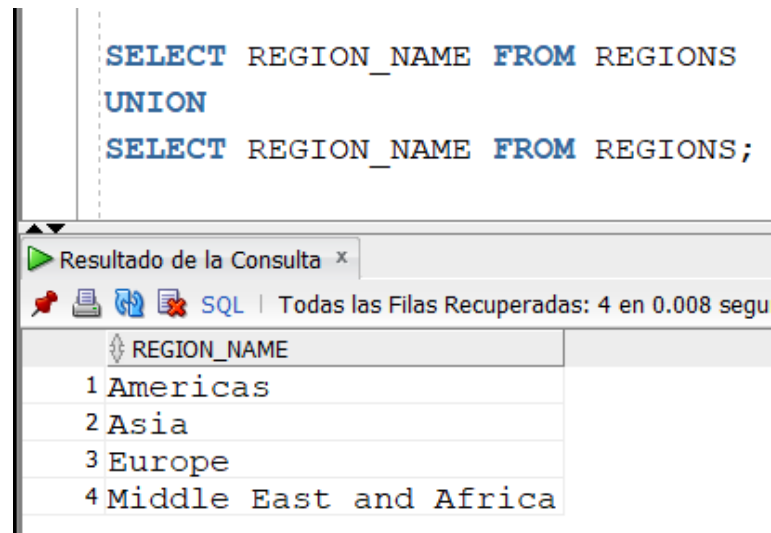
1. Connect to your database as user HR.
2. Run a query that consult the regions table (region_name): **see figure 1.**

Note the result, in particular the order of the rows. If the table is as originally created, there will be four rows returned. The order will be Europe, America, Asia, Middle East.

Figure 1: *Using set operations.*

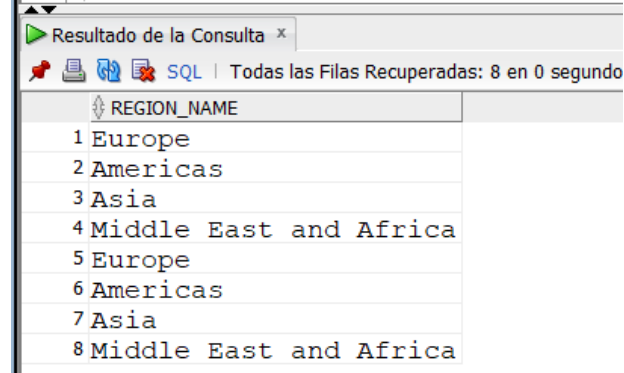
3. Query the Regions table twice, using UNION, see **figure 2**.

The rows returned will be as for step 1 but sorted alphabetically

Figure 2: *Using set operations.*

4. This time, use UNION ALL: There will be double the number of rows, and they will not be sorted, see figure 3.

```
SELECT REGION_NAME FROM REGIONS
UNION ALL
SELECT REGION_NAME FROM REGIONS;
```

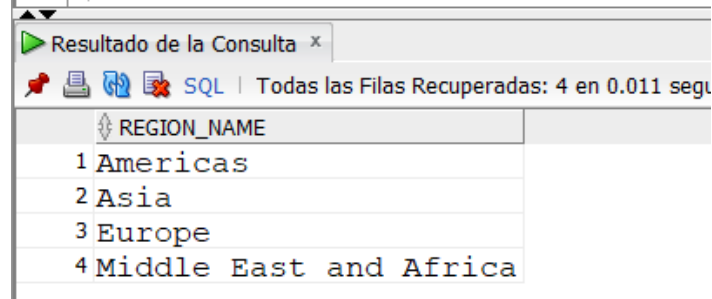


REGION_NAME
1 Europe
2 Americas
3 Asia
4 Middle East and Africa
5 Europe
6 Americas
7 Asia
8 Middle East and Africa

Figure 3: *Using set operations.*

5. An intersection will retrieve rows common to two queries. All four rows are common, and the result is sorted, see figure 4.

```
SELECT REGION_NAME FROM REGIONS
INTERSECT
SELECT REGION_NAME FROM REGIONS;
```



REGION_NAME
1 Americas
2 Asia
3 Europe
4 Middle East and Africa

Figure 4: *Using set operations.*

6. A MINUS will remove common rows. The second query will remove all the rows in the first query.
Result: no rows left, see figure 5.

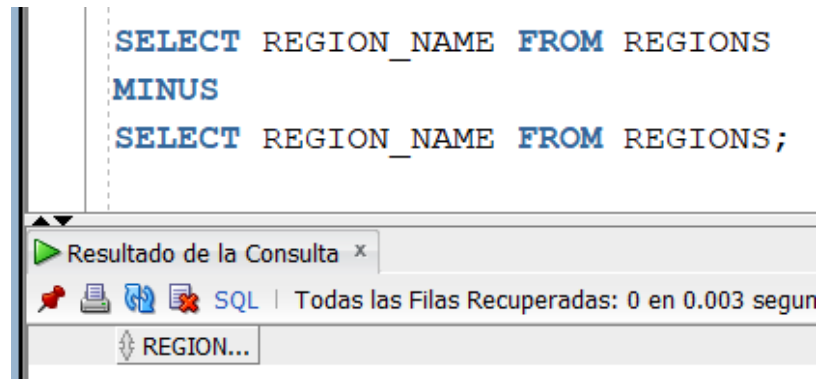


Figure 5: *Using set operations.*

7. Execute these statements and show results.

b) In this exercise, you will run more complex compound queries.

1. Connect to your database as user HR.

2. Run a simple query to count the employees in three departments (20,30,40), grouped by them, see figure 6.

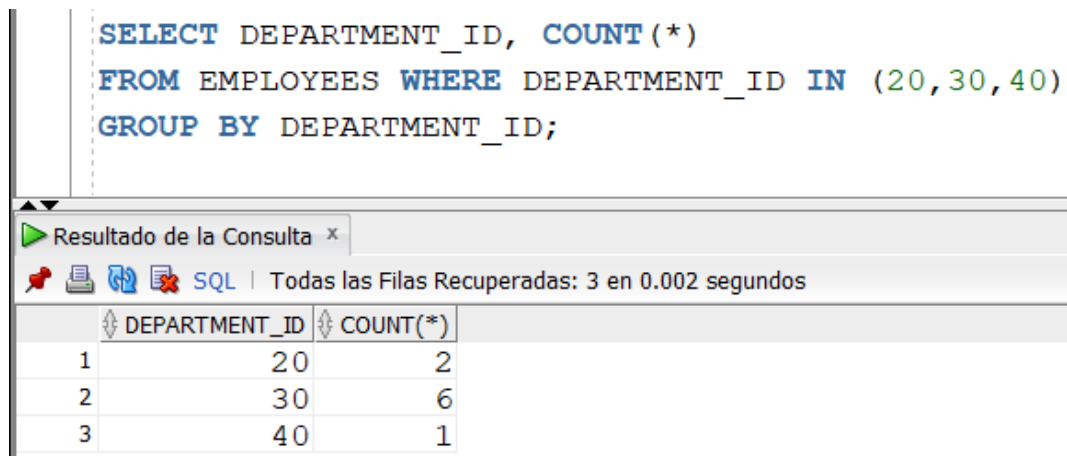
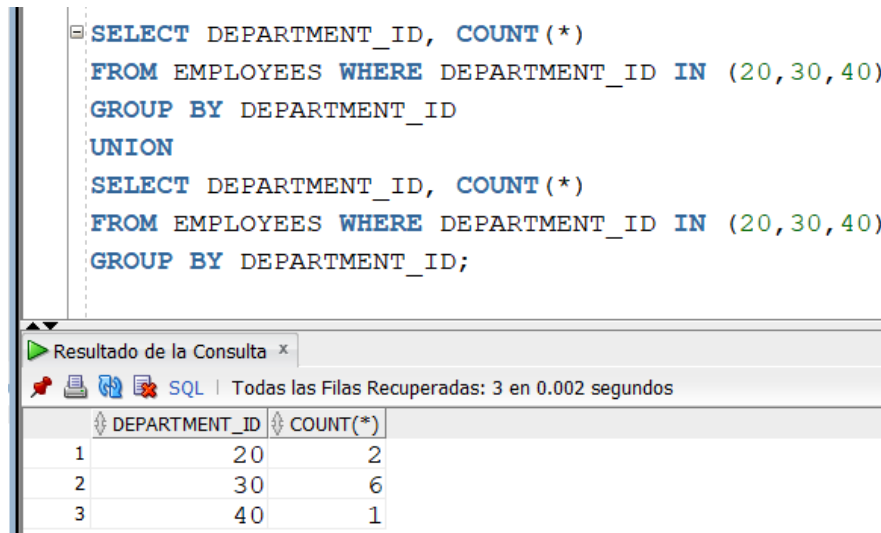


Figure 6: *Using set operations.*

3. Obtain the same result with a compound query, see figure 7.



```
SELECT DEPARTMENT_ID, COUNT(*)
FROM EMPLOYEES WHERE DEPARTMENT_ID IN (20,30,40)
GROUP BY DEPARTMENT_ID
UNION
SELECT DEPARTMENT_ID, COUNT(*)
FROM EMPLOYEES WHERE DEPARTMENT_ID IN (20,30,40)
GROUP BY DEPARTMENT_ID;
```

Resultado de la Consulta x

SQL | Todas las Filas Recuperadas: 3 en 0.002 segundos

	DEPARTMENT_ID	COUNT(*)
1	20	2
2	30	6
3	40	1

Figure 7: *Using set operations.*

4. Find out (using compound queries) if any managers manage staff in both departments 20 and 30, and exclude any managers with staff in department 40, see figure 8.



```
SELECT MANAGER_ID FROM EMPLOYEES WHERE DEPARTMENT_ID = 20
INTERSECT
SELECT MANAGER_ID FROM EMPLOYEES WHERE DEPARTMENT_ID = 30
MINUS
SELECT MANAGER_ID FROM EMPLOYEES WHERE DEPARTMENT_ID = 40;
```

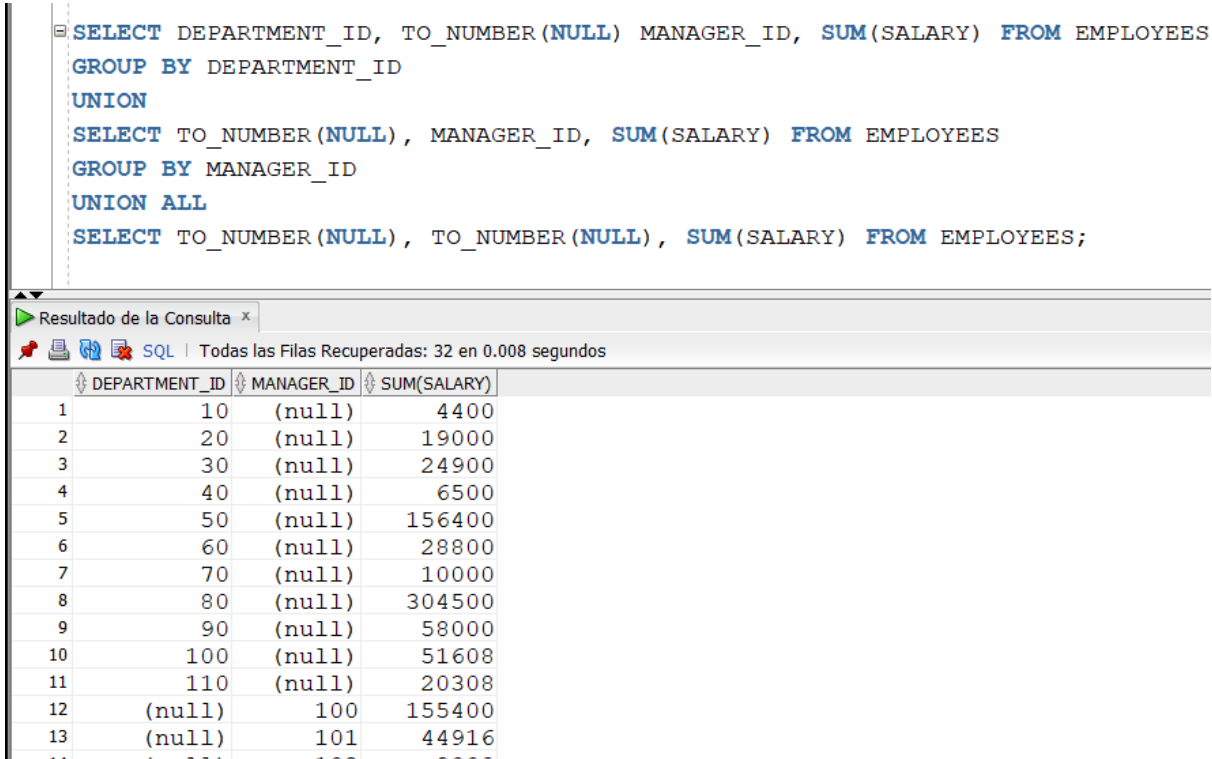
Resultado de la Consulta x

SQL | Todas las Filas Recuperadas: 1 en 0.009 segundos

	MANAGER_ID
1	100

Figure 8: *Using set operations.*

5. Use a compound query (3 sentences using two set operator) to report salaries (from employees) subtotaled by department (grouped by department_id), by manager (grouped by manager_id), and the overall total. Order the query, see figure 9.



```

SELECT DEPARTMENT_ID, TO_NUMBER(NULL) MANAGER_ID, SUM(SALARY) FROM EMPLOYEES
GROUP BY DEPARTMENT_ID
UNION
SELECT TO_NUMBER(NULL), MANAGER_ID, SUM(SALARY) FROM EMPLOYEES
GROUP BY MANAGER_ID
UNION ALL
SELECT TO_NUMBER(NULL), TO_NUMBER(NULL), SUM(SALARY) FROM EMPLOYEES;

```

DEPARTMENT_ID	MANAGER_ID	SUM(SALARY)
1	10	(null)
2	20	(null)
3	30	(null)
4	40	(null)
5	50	(null)
6	60	(null)
7	70	(null)
8	80	(null)
9	90	(null)
10	100	(null)
11	110	(null)
12	(null)	100
13	(null)	101

Figure 9: *Using set operations.*

6. Execute these statements and show results.

c) Working in the HR schema, design some queries that will generate reports using the set operators. The reports required are as follows:

1. Employees have their current job (identified by JOB_ID) recorded in their EMPLOYEES row. Jobs they have held previously (but not their current job) are recorded in JOB_HISTORY. Which employees have never changed jobs? The listing should include the employees' EMPLOYEE_ID and LAST_NAME, see figure 10.

```
SELECT EMPLOYEE_ID, LAST_NAME FROM EMPLOYEES
MINUS
SELECT EMPLOYEE_ID, TO_CHAR(NULL) FROM JOB_HISTORY;
```

Resultado de la Consulta x

SQL | Todas las Filas Recuperadas: 107 en 0.057 segundos

EMPLOYEE_ID	LAST_NAME
1	100 King
2	101 Kochhar
3	102 De Haan
4	103 Hunold
5	104 Ernst
6	105 Austin
7	106 Pataballa
8	107 Lorentz
9	108 Greenberg
10	109 Faviet

Figure 10: *Using set operations.*

2. Which employees were recruited into one job, then changed to a different job, but are now back in a job they held before? Again, you will need to construct a query that compares EMPLOYEES with JOB_HISTORY. The report should show the employees' names and the job titles. Job titles are stored in the table JOBS, see figure 11.

```
SELECT EMPLOYEE_ID,
       (SELECT J.JOB_TITLE FROM JOBS J WHERE
        E.JOB_ID = J.JOB_ID) "JOB TITLE" FROM EMPLOYEES E
INTERSECT
SELECT EMPLOYEE_ID,
       (SELECT J2.JOB_TITLE FROM JOBS J2 WHERE
        J2.JOB_ID = JH.JOB_ID) "JOB TITLE" FROM JOB_HISTORY JH;
```

Resultado de la Consulta x

SQL | Todas las Filas Recuperadas: 2 en 0 segundos

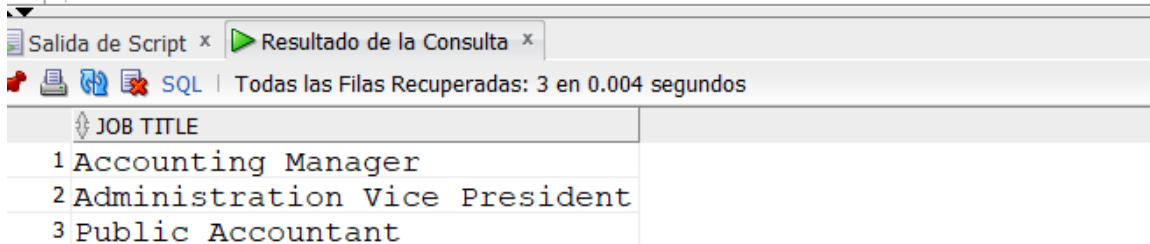
EMPLOYEE_ID	JOB TITLE
1	176 Sales Representative
2	200 Administration Assistant

Figure 11: *Using set operations.*

3. What jobs has any one employee held? This will be the JOB_ID for the employee's current job (in EMPLOYEES) and all previous jobs (in JOB_HISTORY). If the employee has held a job more than once, there is no need to list it more than once. Use a replacement variable to prompt for the EMPLOYEE_ID and display the job title(s). Employees 101 and 200 will be suitable employees for testing.

Employee 101: see figure 12.

```
SELECT (SELECT J.JOB_TITLE FROM JOBS J WHERE
        E.JOB_ID = J.JOB_ID) "JOB TITLE"
FROM EMPLOYEES E WHERE E.EMPLOYEE_ID = &EMPLOYEE_ID
UNION
SELECT (SELECT J2.JOB_TITLE FROM JOBS J2 WHERE
        J2.JOB_ID = JH.JOB_ID) "JOB TITLE"
FROM JOB_HISTORY JH WHERE JH.EMPLOYEE_ID = &EMPLOYEE_ID;
```

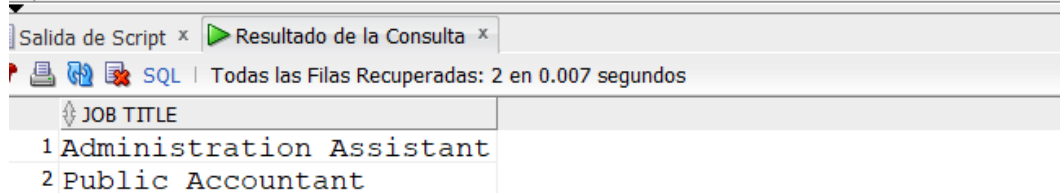


JOB TITLE
1 Accounting Manager
2 Administration Vice President
3 Public Accountant

Figure 12: *Using set operations.*

Employee 200: see figure 13.

```
SELECT (SELECT J.JOB_TITLE FROM JOBS J WHERE
        E.JOB_ID = J.JOB_ID) "JOB TITLE"
FROM EMPLOYEES E WHERE E.EMPLOYEE_ID = &EMPLOYEE_ID
UNION
SELECT (SELECT J2.JOB_TITLE FROM JOBS J2 WHERE
        J2.JOB_ID = JH.JOB_ID) "JOB TITLE"
FROM JOB_HISTORY JH WHERE JH.EMPLOYEE_ID = &EMPLOYEE_ID;
```



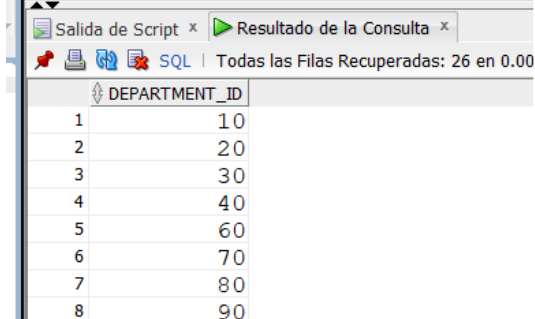
JOB TITLE
1 Administration Assistant
2 Public Accountant

Figure 13: *Using set operations.*

Activity 4: In this activity you will write several queries using the set operators.

1. The HR department needs a list of department IDs for departments that do not contain the job ID ST_CLERK. Use the set operators to create this report, see figure 14.

```
-- PRACTICE 14, ACTIVITY 04:
SELECT DEPARTMENT_ID
FROM DEPARTMENTS
MINUS
SELECT DEPARTMENT_ID
FROM EMPLOYEES
WHERE JOB_ID = 'ST_CLERK';
```



DEPARTMENT_ID
1 10
2 20
3 30
4 40
5 60
6 70
7 80
8 90
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26

Figure 14: *Using set operations.*

2. The HR department needs a list of countries that have no departments located in them. Display the country ID and the name of the countries. Use the set operators to create this report, **see figure 15**.

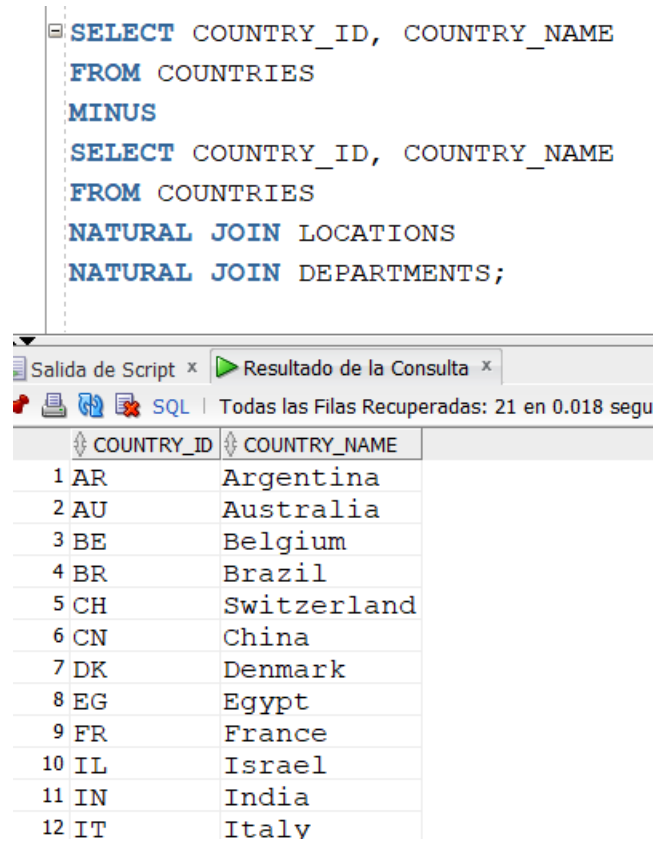


Figure 15: *Using set operations.*

3. Produce a list of jobs for departments 10, 50, and 20, in that order. Display the job ID and department ID by using the set operators, **see figure 16**.

```

SELECT JOB_ID, DEPARTMENT_ID FROM EMPLOYEES
WHERE DEPARTMENT_ID = 10
UNION
SELECT JOB_ID, DEPARTMENT_ID FROM EMPLOYEES
WHERE DEPARTMENT_ID = 50
UNION
SELECT JOB_ID, DEPARTMENT_ID FROM EMPLOYEES
WHERE DEPARTMENT_ID = 20;

```

JOB_ID	DEPARTMENT_ID
1 AD ASST	10
2 MK MAN	20
3 MK REP	20
4 SH CLERK	50
5 ST CLERK	50
6 ST MAN	50

Figure 16: *Using set operations.*

4. Create a report that lists the employee IDs and job IDs of those employees who currently have a job title that is the same as their job title when they were initially hired by the company (that is, they changed jobs but have now gone back to doing their original job), see figure 17.

```

SELECT EMPLOYEE_ID, JOB_ID
FROM EMPLOYEES
INTERSECT
SELECT EMPLOYEE_ID, JOB_ID
FROM JOB_HISTORY;

```

EMPLOYEE_ID	JOB_ID
1	176 SA REP
2	200 AD ASST

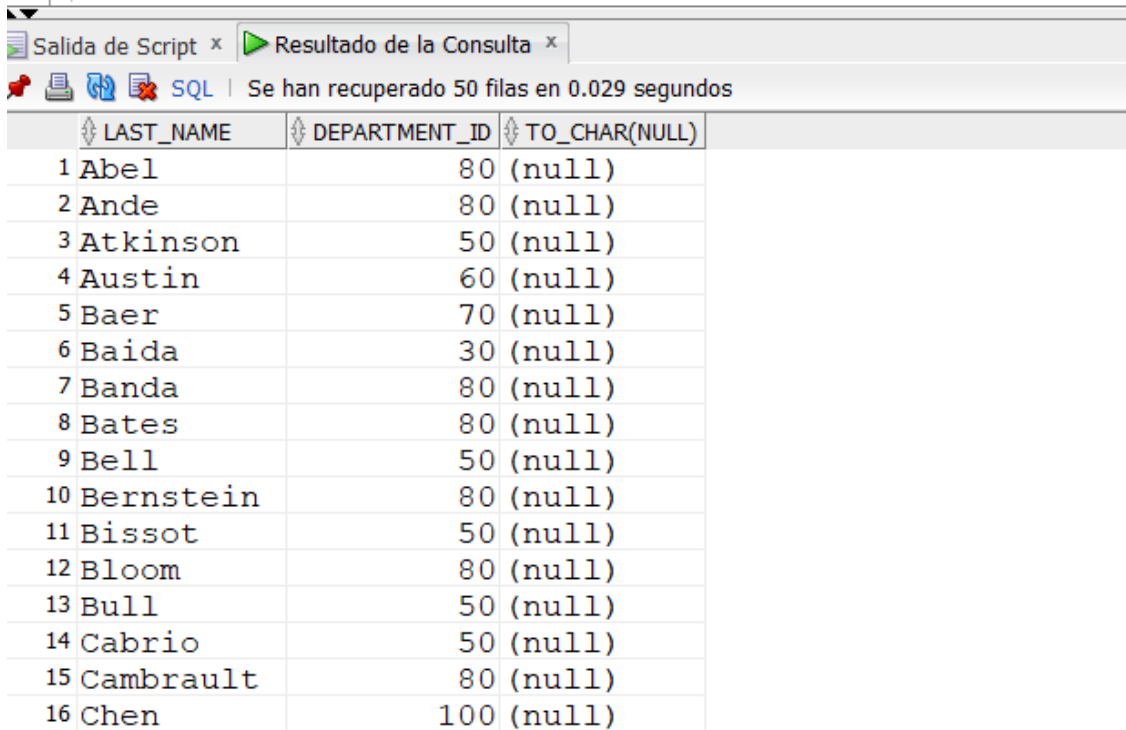
Figure 17: *Using set operations.*

5. The HR department needs a report with the following specifications:

- Last name and department ID of all employees from the EMPLOYEES table, regardless of whether or not they belong to a department.
- Department ID and department name of all departments from the DEPARTMENTS table, regardless of whether or not they have employees working in them.

Write a compound query to accomplish this.

```
SELECT LAST_NAME, DEPARTMENT_ID, TO_CHAR(NULL)
FROM EMPLOYEES
UNION
SELECT TO_CHAR(NULL), DEPARTMENT_ID, DEPARTMENT_NAME
FROM DEPARTMENTS;
```



	LAST_NAME	DEPARTMENT_ID	TO_CHAR(NULL)
1	Abel	80	(null)
2	Ande	80	(null)
3	Atkinson	50	(null)
4	Austin	60	(null)
5	Baer	70	(null)
6	Baida	30	(null)
7	Banda	80	(null)
8	Bates	80	(null)
9	Bell	50	(null)
10	Bernstein	80	(null)
11	Bissot	50	(null)
12	Bloom	80	(null)
13	Bull	50	(null)
14	Cabrio	50	(null)
15	Cambrault	80	(null)
16	Chen	100	(null)

Figure 18: *Using set operations.*

4 Pre-assessment

In this section you will find the Pre-assessment.

Criteria to be evaluate	Does it comply?	(%)
COMPLIES WITH THE REQUESTED FUNCTIONALITY	YES	
HAS THE CORRECT INDENTATION	YES	
HAS AN EASY WAY TO ACCESS THE PROVIDED FILES	YES	
HAS A REPORT WITH IDC FORMAT	YES	
REPORT INFORMATION IS FREE OF SPELLING ERRORS	YES	
DELIVERED IN TIME AND FORM	YES	
IS FULLY COMPLETED (SPECIFY THE PERCENTAGE COMPLETED)	YES	100%

5 Conclusion

The projection of the data within a relational database is stored in the table in the form of rows and columns. Projections are the first items identified during query execution. They are the selected columns within a table for which a query has been designed. Projections are mentioned in the first part of the SQL query, that is, the SELECT statement. After identifying the projections within the query frame, the next step would be to identify the rows that are relevant to the query. Filters are mentioned within the WHERE clause of the query and will identify the rows to be included in the results, the latter is called a selection.

This practice number 14 helped me practice the uses of the SELECT statement for data retrieval and projection. Finally, something important to mention is that the SQL language allows the projection and selection of data to meet the reporting needs that a programmer, developer or end user may need.