Autonomous University of Zacatecas

Academic Unit of Electrical Engineering

Software Engineering Academic Program

Group: 5B - Semester: 2022-5º

Practice Number: 10

Using Conversion Functions and Conditional Expressions

Date: 20/October/2022

Professor:

Aldonso Becerra Sánchez.

Student:

Cristian Omar Alvarado Rodríguez.

Contents

Data selection and projection: single row functions

October 20th, 2022

1 Introduction

All DBMS implement functions to facilitate the creation of complex queries. These functions vary in each Database Management System.

Conversion functions convert a value from one data type to another. Now, ORACLE SERVER has an internal implicit conversion mechanism which works like this:

Implicit Conversions:

If X column, variable, function, etc expects a data type and receives a different one, ORACLE SERVER converts it implicitly whenever possible. I repeat, ORACLE SERVER does these conversions whenever possible, that is, the conversion from VARCHAR2 to NUMBER is possible if the character string represents a valid number. Example: salary = '12000'. Likewise, the conversion from VARCHAR2 to DATE is possible if the character string represents a valid date. Example: hire_date = '12-Jun-06'. The case of conversions from NUMBER or DATE to VARCHAR2 are always possible since any character string can be VARCHAR2.

Explicit Conversions:

As Implicit conversions are not always possible and also when there is a need to convert the data with a specific format, it is required to use the Conversion Functions available in ORACLE SQL.

Function: TO_CHAR:

Syntax: TO_CHAR(number—date,[fmt], [nlsparams])

Description: Converts a numeric or date value to a VARCHAR2 character string with the fmt format model (optional). Date Conversion: The nlsparams parameter (optional) specifies the language in which month and day names and abbreviations will be returned. If this parameter is omitted, this function uses the default date languages for the session. Numeric conversion: The nlsparams parameter specifies the following characters, which are returned by numeric format elements:

- Decimal character.
- Group separator.
- Local currency symbol.
- International Currency Symbol, If nlsparams or any other parameters are omitted, this function uses the default parameter values for the session.

Function: TO_NUMBER.

Syntax: TO_NUMBER(char,[fmt], [nlsparams])

Description: Converts a character string containing digits to a number in the format specified by the fmt optional format model. The nlsparams parameter has the same purpose in this function as it does in the TO_CHAR numeric conversion function.

Function: TO_DATE:

Syntax: TO_DATE(char,[fmt], [nlsparams])

Description: Converts a character string representing a date to a date value according to the specified fmt. If fmt is omitted, the format is DD-MON-YY. The nlsparams parameter has the same purpose in this function as it does in the date conversion TO_CHAR function.

Conditional Expressions:

• Allow the use of **IF-THEN-ELSE** logic within an SQL statement.

• The two methods used to implement conditional processing (**IF-THEN-ELSE logic**) in an SQL statement are the CASE expression and the DECODE function.

Note: The CASE expression complies with ANSI SQL. The DECODE function is specific to Oracle syntax.

2 Practice objective

Use SQL SELECT statements for retrieving data from database by means of different contexts using different Oracle functions and conditional expressions.

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 VARIOUS TYPES OF CONVERSION FUNCTIONS AVAILABLE IN SQL.

1. What type of conversion is performed by the following statement:
SELECT LENGTH(3.14285) FROM DUAL; (Choose the best answer.)

A) Explicit conversion.

B) Implicit conversion.

C) TO_NUMBER function conversion.

D) None of the above.

Explanation: B, The number 3.14285 is given as a parameter of the LENGTH function. There is a data type mismatch, but Oracle implicitly converts the parameter to the string ”3.14285”, which allows the function to work correctly.

2. Choose any incorrect statements regarding conversion functions. (Choose all that apply.)

A) TO_CHAR may convert date items to character items.

B) TO_DATE may convert character items to date items.

C) TO_CHAR may convert numbers to character items.

D) TO_DATE may convert date items to character items.

Explanation: D, Dates are only converted into character strings using TO_CHAR and not the TO_DATE function.

USE THE TO_CHAR,TO_NUMBER, AND TO_DATE CONVERSION FUNCTIONS.

3. What value is returned after executing the following statement?
SELECT TO_NUMBER(1234.49, ’999999.9’) FROM DUAL;
(Choose the best answer.

A) 1234.49

B) 001234.5

C) 1234.5

D) None of the above.

Explanation: D, An error is returned because the statement attempts to convert a number using an incompatible format mask. If the expression is TO_NUMBER (1234.49, ’999999.99’), the number 1234.49 will be returned.

4. What value is returned after executing the following statement?

SELECT TO_CHAR(1234.49, '999999.9') FROM DUAL;

(Choose the best answer.)

A) 1234.49

B) 001234.5

C) 1234.5

D) None of the above.

Explanation: C, In order for the number 1234.49 to match the character format mask to a decimal place, the number is first rounded to 1234.5 before TO_CHAR converts it to the string "1234.5".

5. If SYSDATE returns 12-JUL-2009, what is returned by the following statement?

SELECT TO_CHAR(SYSDATE, 'fmMONTH, YEAR') FROM DUAL;

(Choose the best answer.)

A) JUL, 2009.

B) JULY, TWO THOUSAND NINE.

C) JUL-09.

D) None of the above.

Explanation: B, The MONTH and YEAR components of the format mask separated by a comma and a space indicate that TO_CHAR must extract the spelled out month and year values in uppercase separated by a comma and a space. The fm modifier removes extra blanks from the spelled out components.

6. If SYSDATE returns 12-JUL-2009, what is returned by the following statement?

SELECT TO_CHAR(SYSDATE, 'fmDDth MONTH') FROM DUAL;

(Choose the best answer.)

A) 12TH JULY.

B) 12th July.

C) TWELFTH JULY.

D) None of the above.

Explanation: A, The DD returns the day of the month in uppercase, since it is a number, it does not matter, unless the 'th' mask is applied, in which case that component is specified in uppercase. MONTH returns the month spelled out in uppercase.

APPLY CONDITIONAL EXPRESSIONS IN A SELECT STATEMENT.

7. If SYSDATE returns 12-JUL-2009, what is returned by the following statement? SELECT TO_CHAR(TO_DATE(TO_CHAR(SYSDATE,'DD'),'DD'),'YEAR' FROM DUAL; (Choose the best answer.)

A) 2009

B) TWO THOUSAND NINE.

C) 12-JUL-2009.

D) None of the above.

Explanation: B, The innermost nested function is TO_CHAR(SYSDATE,'DD'), which extracts the day component of SYSDATE and returns the character 12. The next function executed is TO_DATE('12','DD') where the character 12 is cast as the day component. When such an incomplete date is provided, Oracle substitutes values from the SYSDATE function; since SYSDATE is 12-JUL-2009, this is the date used. The outermost function executed in TO_CHAR('12-JUL-2009','YEAR') returns the year spelled out as TWO THOUSAND NINE.

8. What value is returned after executing the following statement?
SELECT NVL2(NULLIF('CODA','SID'),'SPANIEL','TERRIER') FROM DUAL;
(Choose the best answer.)

A) SPANIEL.

B) TERRIER.

C) NULL.

D) None of the above.

Explanation: A, The NULLIF function compares its two parameters and, since they are different, the first parameter is returned. The NVL2('CODA', 'SPANIEL','TERRIER') function call returns SPANIEL since its first parameter is not null.

9. What value is returned after executing the following statement?
SELECT NVL(SUBSTR('AM I NULL',10),'YES I AM') FROM DUAL;
(Choose the best answer.)

A) NO.

B) NULL.

C) YES I AM.

D) None of the above.

Explanation: C, The character literal 'AM I NULL' is nine characters long. Therefore, trying to obtain a substring beginning at the tenth character returns a null. The outer function then becomes NVL(NULL,'YES I AM'), resulting in the string ‘YES I AM’ being returned.

10. If SYSDATE returns 12-JUL-2009, what is returned by the following statement?)
SELECT DECODE(TO_CHAR(SYSDATE,'MM'),'02','TAX DUE','PARTY') FROM DUAL;
(Choose the best answer.)

A) TAX DUE.

B) PARTY.

C) 02.

D) None of the above.

Explanation: B, The innermost function TO_CHAR(SYSDATE, 'MM') results in the character string '07' being returned. The outer function is DECODE('07','02','TAX DUE','PARTY'). Since '07' is not equal to '02', the else component 'PARTY' is returned.

Activity 2: Propose an answer to the following issues:

a) Your task is to extract the day and month portion of a date column and compare it with the corresponding components of the current system date. Can such a comparison be performed?

R = Yes this is possible as long as both dates have the same format; this would be easier by making use of the EXTRACT function.

b) A report of profit and loss is required with the results displayed as follows: if the amount is negative, it must be enclosed in angle brackets. The amount must be displayed with a leading dollar sign. Can results be retrieved in the specified format?

R = If it is possible since with the conditional expressions CASE and DECODE they allow us to do this in a very simple way. The numeric amount must be converted into a character string using the TO_CHAR function with a format mask that encloses it in angle brackets if it is negative and precedes it with a dollar sign. The following function call retrieves the results in the required format: TO_CHAR(AMOUNT, '$999999PR')

c) You are asked to input past employee data into the JOB_HISTORY table from a paper-based source, but the start date information is only available as the year the employee started. Can this value be converted into the first of January of the year?

R = Yes it is possible for example consider the conversion function called TO_DATE('2002', 'YYYY') for an employee who started working in the year 2002. If this date is extracted as follows, the string of characters 01/01/2002 TO_CHAR(TO_DATE('2002','YYYY'), 'MM/DD/YYYY').

d) Are nested functions evaluated from the outermost level to the innermost level?

R = No, nested functions are resolved from the innermost nested level moving outward.

e) Must all functions in a nested expression return the same data type?

R = No, since the data types of the parameters of the nested functions can be different from each other. It is important to ensure that the correct data types are always supplied to functions to avoid errors.

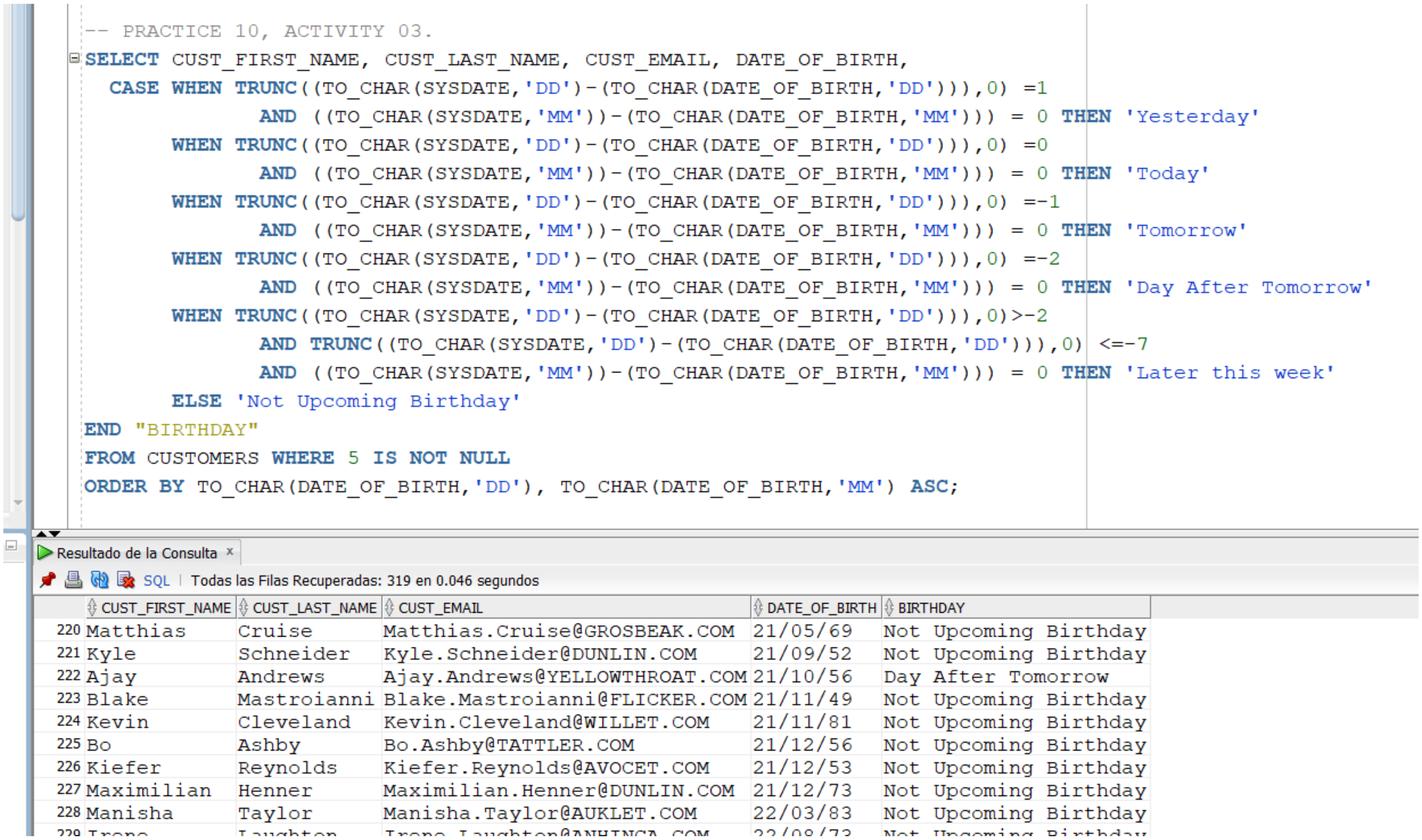
f) Is there a simpler way to display the SALARY information from the EMPLOYEES table in the form $19,000 without using the following statement?
SELECT '$' ll SUBSTR(SALARY,1, MOD(LENGTH(SALARY),3)) ll ',' ll SUBSTR(SALARY, MOD (LENGTH(SALARY),3)+1);

R = Yes a simple and elegant solution is to use the TO_CHAR function with the '$99G999' format mask SELECT TO_CHAR(SALARY, '$99G999') FROM EMPLOYEES; The letter G helps us with the separation of the thousands depending on the region where ORACLE DATABASE is installed since it is dynamic.

Activity 3: Connect to the OE schema and complete the following tasks.

- As part of a new marketing initiative, you are asked to prepare a list of customer birthdays that occur between two days ago and seven days from now. The list should retrieve rows from the CUSTOMERS table which include the CUST_FIRST_NAME, CUST_LAST_NAME, CUST_EMAIL, and DATE_OF_BIRTH columns in ascending order based on the day and month components of the DATE_OF_BIRTH value. An additional expression aliased as BIRTHDAY is required to return a descriptive message based on the following table, **see figure 1**. The current date in SYSDATE is equal to 10/19/2022.

Note: Capture an image for each statement output.



```sql
-- PRACTICE 10, ACTIVITY 03.
SELECT CUST_FIRST_NAME, CUST_LAST_NAME, CUST_EMAIL, DATE_OF_BIRTH,
  CASE WHEN TRUNC((TO_CHAR(SYSDATE,'DD')-(TO_CHAR(DATE_OF_BIRTH,'DD'))),0) =1
              AND ((TO_CHAR(SYSDATE,'MM'))-(TO_CHAR(DATE_OF_BIRTH,'MM'))) = 0 THEN 'Yesterday'
       WHEN TRUNC((TO_CHAR(SYSDATE,'DD')-(TO_CHAR(DATE_OF_BIRTH,'DD'))),0) =0
              AND ((TO_CHAR(SYSDATE,'MM'))-(TO_CHAR(DATE_OF_BIRTH,'MM'))) = 0 THEN 'Today'
       WHEN TRUNC((TO_CHAR(SYSDATE,'DD')-(TO_CHAR(DATE_OF_BIRTH,'DD'))),0) =-1
              AND ((TO_CHAR(SYSDATE,'MM'))-(TO_CHAR(DATE_OF_BIRTH,'MM'))) = 0 THEN 'Tomorrow'
       WHEN TRUNC((TO_CHAR(SYSDATE,'DD')-(TO_CHAR(DATE_OF_BIRTH,'DD'))),0) =-2
              AND ((TO_CHAR(SYSDATE,'MM'))-(TO_CHAR(DATE_OF_BIRTH,'MM'))) = 0 THEN 'Day After Tomorrow'
       WHEN TRUNC((TO_CHAR(SYSDATE,'DD')-(TO_CHAR(DATE_OF_BIRTH,'DD'))),0)>-2
              AND TRUNC((TO_CHAR(SYSDATE,'DD')-(TO_CHAR(DATE_OF_BIRTH,'DD'))),0) <=-7
              AND ((TO_CHAR(SYSDATE,'MM'))-(TO_CHAR(DATE_OF_BIRTH,'MM'))) = 0 THEN 'Later this week'
       ELSE 'Not Upcoming Birthday'
END "BIRTHDAY"
FROM CUSTOMERS WHERE 5 IS NOT NULL
ORDER BY TO_CHAR(DATE_OF_BIRTH,'DD'), TO_CHAR(DATE_OF_BIRTH,'MM') ASC;
```

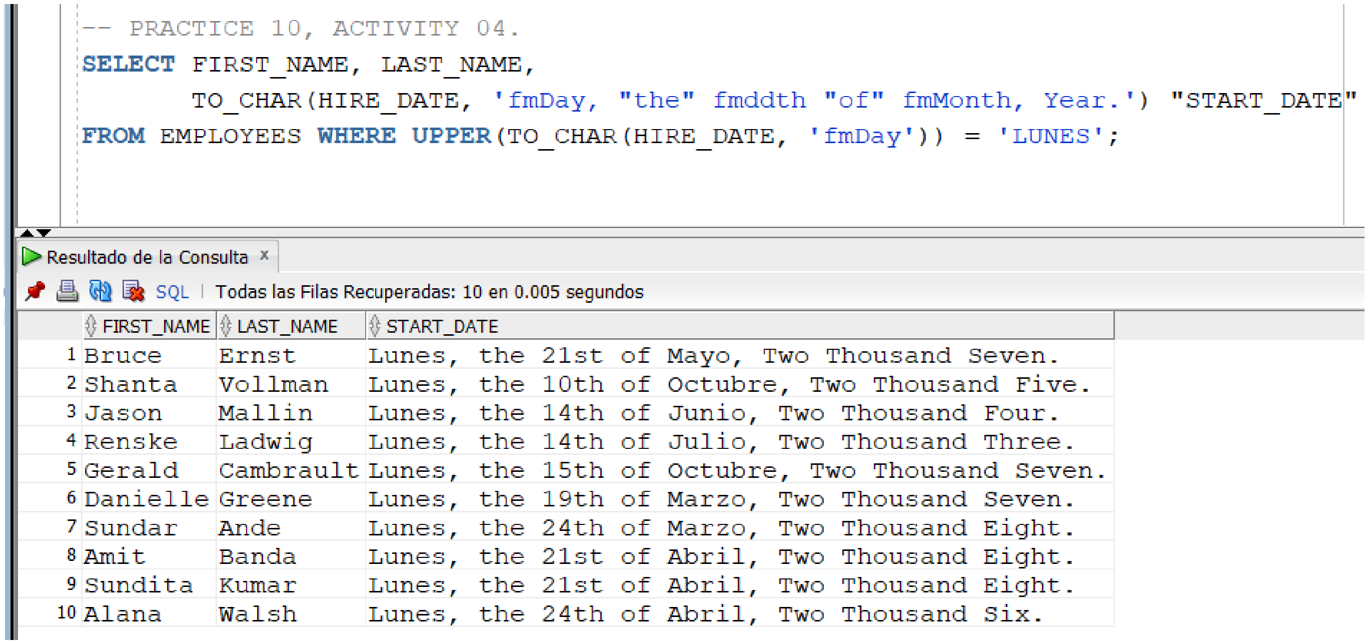
Todas las Filas Recuperadas: 319 en 0.046 segundos

```
    CUST_FIRST_NAME CUST_LAST_NAME CUST_EMAIL                     DATE_OF_BIRTH BIRTHDAY
220 Matthias        Cruise         Matthias.Cruise@GROSBEAK.COM   21/05/69      Not Upcoming Birthday
221 Kyle            Schneider      Kyle.Schneider@DUNLIN.COM      21/09/52      Not Upcoming Birthday
222 Ajay            Andrews        Ajay.Andrews@YELLOWTHROAT.COM  21/10/56      Day After Tomorrow
223 Blake           Mastroianni    Blake.Mastroianni@FLICKER.COM  21/11/49      Not Upcoming Birthday
224 Kevin           Cleveland      Kevin.Cleveland@WILLET.COM     21/11/81      Not Upcoming Birthday
225 Bo              Ashby          Bo.Ashby@TATTLER.COM           21/12/56      Not Upcoming Birthday
226 Kiefer          Reynolds       Kiefer.Reynolds@AVOCET.COM     21/12/53      Not Upcoming Birthday
227 Maximilian      Henner         Maximilian.Henner@DUNLIN.COM   21/12/73      Not Upcoming Birthday
228 Manisha         Taylor         Manisha.Taylor@AUKLET.COM      22/03/83      Not Upcoming Birthday
229 Irene           Laughton       Irene.Laughton@ANHINGA.COM     22/08/73      Not Upcoming Birthday
```

Figure 1: *SELECT statement, data projection.*

Activity 4: This exercise must be performed using HR schema.

- You are required to retrieve a list of FIRST_NAME and LAST_NAME values and an expression based on the HIRE_DATE column for employees hired on a Saturday. The expression must be aliased as START_DATE and a HIRE_DATE value of 17-FEB-1996 must return the following string: Saturday, the 17th of February, One Thousand Nine Hundred Ninety-Six. Due to the Oracle Database configuration, I decided to change Saturday to Monday since the Oracle server did not recognize the accent on the word Saturday, **see figure 2**.



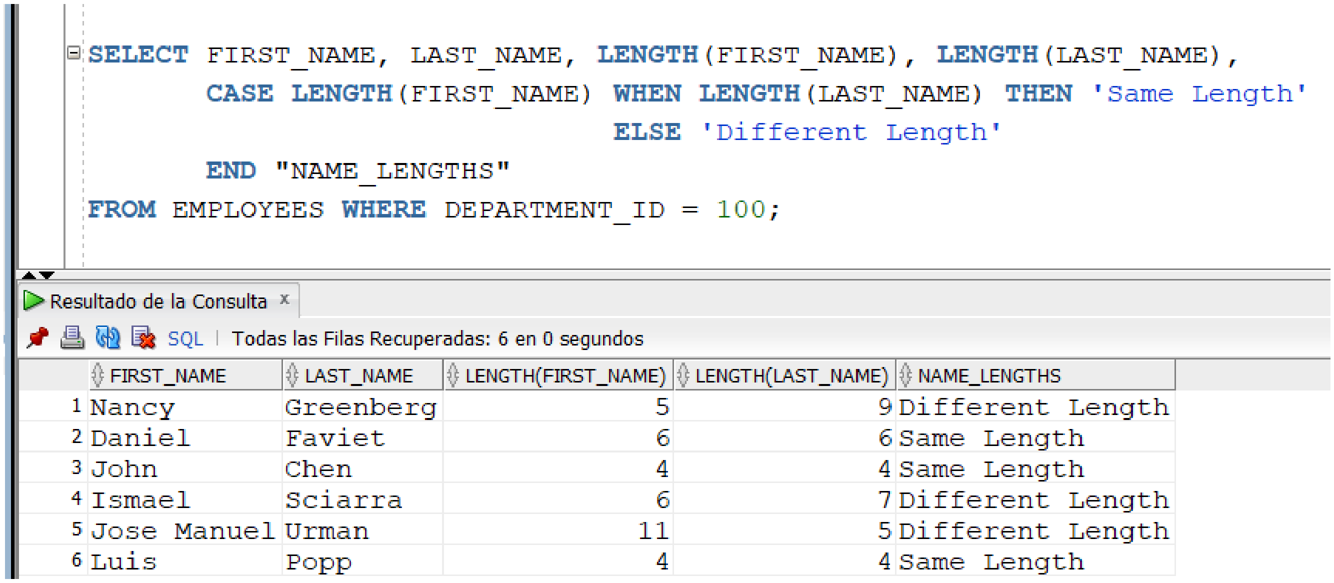
```
-- PRACTICE 10, ACTIVITY 04.
SELECT FIRST_NAME, LAST_NAME,
       TO_CHAR(HIRE_DATE, 'fmDay, "the" fmddth "of" fmMonth, Year.') "START_DATE"
FROM EMPLOYEES WHERE UPPER(TO_CHAR(HIRE_DATE, 'fmDay')) = 'LUNES';
```

Todas las Filas Recuperadas: 10 en 0.005 segundos

```
   FIRST_NAME LAST_NAME  START_DATE
 1 Bruce      Ernst      Lunes, the 21st of Mayo, Two Thousand Seven.
 2 Shanta     Vollman    Lunes, the 10th of Octubre, Two Thousand Five.
 3 Jason      Mallin     Lunes, the 14th of Junio, Two Thousand Four.
 4 Renske     Ladwig     Lunes, the 14th of Julio, Two Thousand Three.
 5 Gerald     Cambrault  Lunes, the 15th of Octubre, Two Thousand Seven.
 6 Danielle   Greene     Lunes, the 19th of Marzo, Two Thousand Seven.
 7 Sundar     Ande       Lunes, the 24th of Marzo, Two Thousand Eight.
 8 Amit       Banda      Lunes, the 21st of Abril, Two Thousand Eight.
 9 Sundita    Kumar      Lunes, the 21st of Abril, Two Thousand Eight.
10 Alana      Walsh      Lunes, the 24th of Abril, Two Thousand Six.
```

Figure 2: *SELECT statement, data projection.*

- You are required to return a set of rows from the EMPLOYEES table with DEPARTMENT_ID values of 100. The set must also contain FIRST_NAME and LAST_NAME values and an expression aliased as NAME_LENGTHS. This expression must return the string 'Different Length' if the length of the FIRST_NAME differs from that of the LAST_NAME, else the string 'Same Length' must be returned, **see figure 3**.



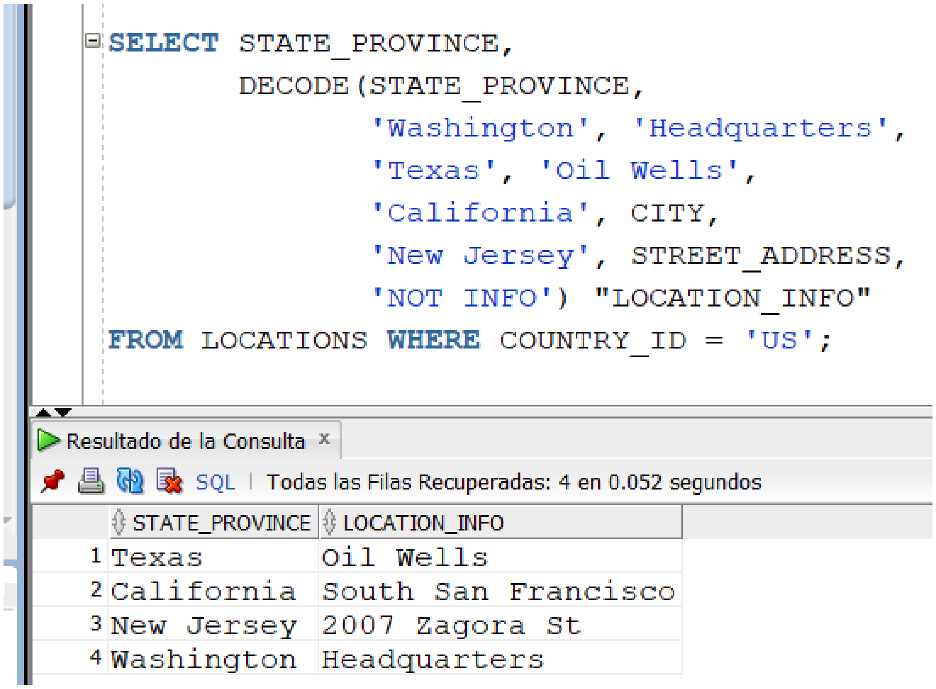
```
SELECT FIRST_NAME, LAST_NAME, LENGTH(FIRST_NAME), LENGTH(LAST_NAME),
       CASE LENGTH(FIRST_NAME) WHEN LENGTH(LAST_NAME) THEN 'Same Length'
                               ELSE 'Different Length'
       END "NAME_LENGTHS"
FROM EMPLOYEES WHERE DEPARTMENT_ID = 100;
```

Todas las Filas Recuperadas: 6 en 0 segundos

	FIRST_NAME	LAST_NAME	LENGTH(FIRST_NAME)	LENGTH(LAST_NAME)	NAME_LENGTHS
1	Nancy	Greenberg	5	9	Different Length
2	Daniel	Faviet	6	6	Same Length
3	John	Chen	4	4	Same Length
4	Ismael	Sciarra	6	7	Different Length
5	Jose Manuel	Urman	11	5	Different Length
6	Luis	Popp	4	4	Same Length

Figure 3: *SELECT statement, data projection.*

- You are requested to query the LOCATIONS table for rows with the value US in the COUNTRY_ID column. An expression aliased as LOCATION_INFO is required to evaluate the STATE_PROVINCE column values and returns different information as per the following table. Sort the output based on the LOCATION_INFO expression. Use the decode function., **see figure 4**.



```
SELECT STATE_PROVINCE,
       DECODE(STATE_PROVINCE,
              'Washington', 'Headquarters',
              'Texas', 'Oil Wells',
              'California', CITY,
              'New Jersey', STREET_ADDRESS,
              'NOT INFO') "LOCATION_INFO"
FROM LOCATIONS WHERE COUNTRY_ID = 'US';
```

Todas las Filas Recuperadas: 4 en 0.052 segundos

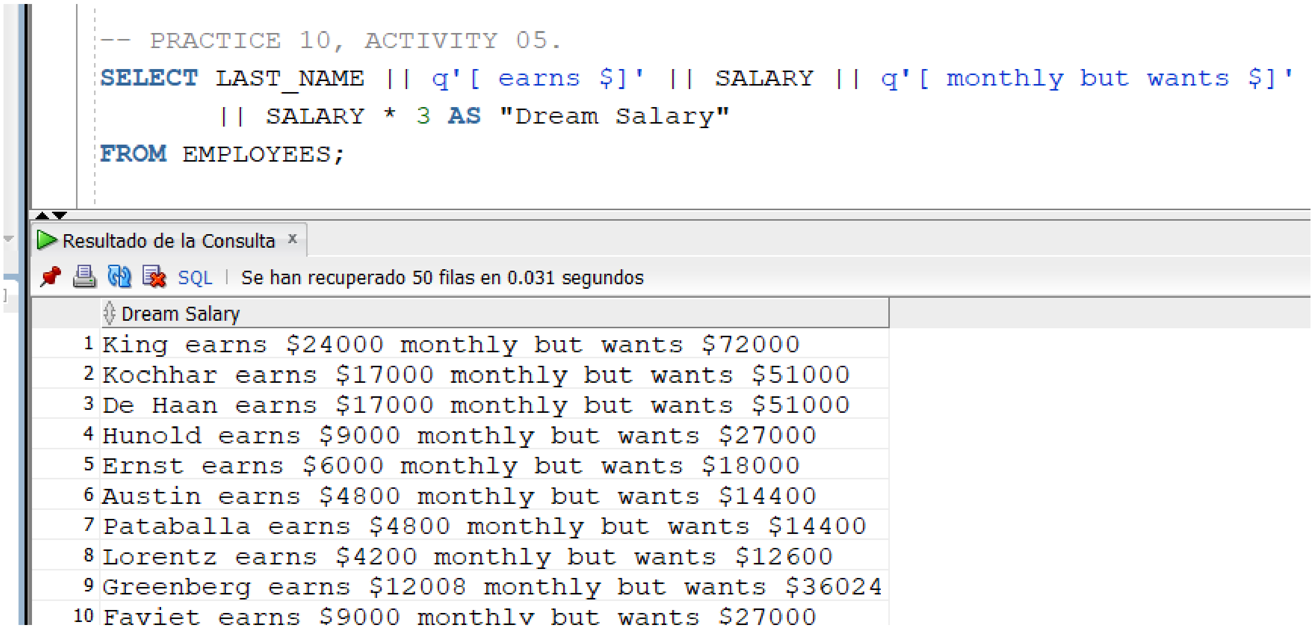
	STATE_PROVINCE	LOCATION_INFO
1	Texas	Oil Wells
2	California	South San Francisco
3	New Jersey	2007 Zagora St
4	Washington	Headquarters

Figure 4: *SELECT statement, data projection.*

Activity 5: This practice provides a variety of exercises using TO_CHAR and TO_DATE functions, and conditional expressions such as DECODE and CASE. Remember that for nested functions, the results are evaluated from the innermost function to the outermost function.

1. Create a report that produces the following for each employee: (employee last name) earns (salary) monthly but wants (3 times salary). Label the column Dream Salaries, **see figure 5**.

2. Display each employee's last name, hire date, and salary review date, which is the first Monday after six months of service. Label the column REVIEW. Format the dates to appear in the format similar to "Monday, the Thirty-First of July, 2000.", **see figure 6**.

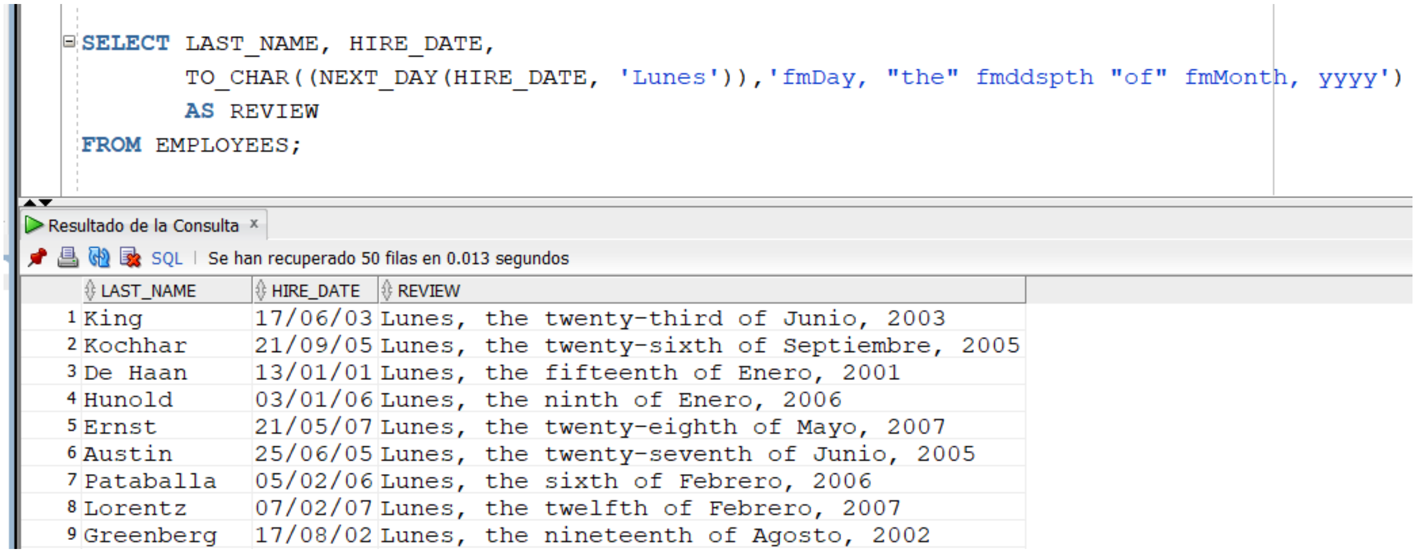


```
-- PRACTICE 10, ACTIVITY 05.
SELECT LAST_NAME || q'[ earns $]' || SALARY || q'[ monthly but wants $]'
        || SALARY * 3 AS "Dream Salary"
FROM EMPLOYEES;
```

Se han recuperado 50 filas en 0.031 segundos

	Dream Salary
1	King earns $24000 monthly but wants $72000
2	Kochhar earns $17000 monthly but wants $51000
3	De Haan earns $17000 monthly but wants $51000
4	Hunold earns $9000 monthly but wants $27000
5	Ernst earns $6000 monthly but wants $18000
6	Austin earns $4800 monthly but wants $14400
7	Pataballa earns $4800 monthly but wants $14400
8	Lorentz earns $4200 monthly but wants $12600
9	Greenberg earns $12008 monthly but wants $36024
10	Faviet earns $9000 monthly but wants $27000

Figure 5: *SELECT statement, data projection.*



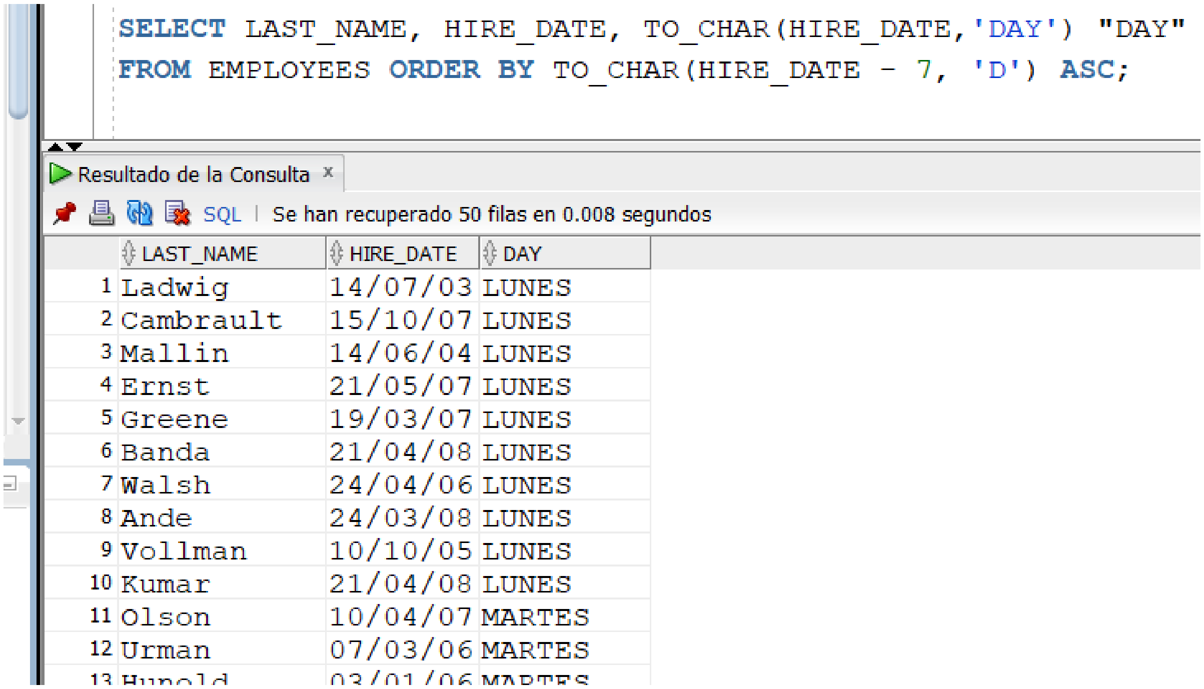
```
SELECT LAST_NAME, HIRE_DATE,
       TO_CHAR((NEXT_DAY(HIRE_DATE, 'Lunes')),'fmDay, "the" fmddspth "of" fmMonth, yyyy')
       AS REVIEW
FROM EMPLOYEES;
```

Se han recuperado 50 filas en 0.013 segundos

	LAST_NAME	HIRE_DATE	REVIEW
1	King	17/06/03	Lunes, the twenty-third of Junio, 2003
2	Kochhar	21/09/05	Lunes, the twenty-sixth of Septiembre, 2005
3	De Haan	13/01/01	Lunes, the fifteenth of Enero, 2001
4	Hunold	03/01/06	Lunes, the ninth of Enero, 2006
5	Ernst	21/05/07	Lunes, the twenty-eighth of Mayo, 2007
6	Austin	25/06/05	Lunes, the twenty-seventh of Junio, 2005
7	Pataballa	05/02/06	Lunes, the sixth of Febrero, 2006
8	Lorentz	07/02/07	Lunes, the twelfth of Febrero, 2007
9	Greenberg	17/08/02	Lunes, the nineteenth of Agosto, 2002

Figure 6: *SELECT statement, data projection.*

3. Display the last name, hire date, and day of the week on which the employee started. Label the column DAY. Order the results by the day of the week, starting with Monday, **see figure 7**.



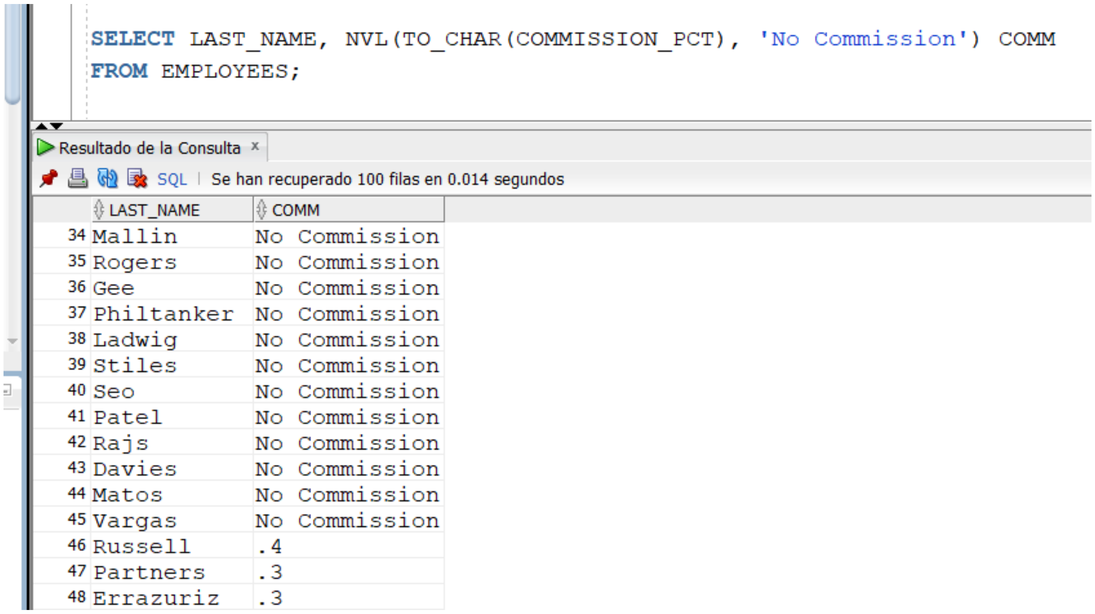
```
SELECT LAST_NAME, HIRE_DATE, TO_CHAR(HIRE_DATE,'DAY') "DAY"
FROM EMPLOYEES ORDER BY TO_CHAR(HIRE_DATE - 7, 'D') ASC;
```

Se han recuperado 50 filas en 0.008 segundos

	LAST_NAME	HIRE_DATE	DAY
1	Ladwig	14/07/03	LUNES
2	Cambrault	15/10/07	LUNES
3	Mallin	14/06/04	LUNES
4	Ernst	21/05/07	LUNES
5	Greene	19/03/07	LUNES
6	Banda	21/04/08	LUNES
7	Walsh	24/04/06	LUNES
8	Ande	24/03/08	LUNES
9	Vollman	10/10/05	LUNES
10	Kumar	21/04/08	LUNES
11	Olson	10/04/07	MARTES
12	Urman	07/03/06	MARTES
13	Hunold	03/01/06	MARTES

Figure 7: *SELECT statement, data projection.*

4. Create a query that displays the employees' last names and commission amounts. If an employee does not earn commission, show "No Commission." Label the column COMM, **see figure 8**.



```
SELECT LAST_NAME, NVL(TO_CHAR(COMMISSION_PCT), 'No Commission') COMM
FROM EMPLOYEES;
```

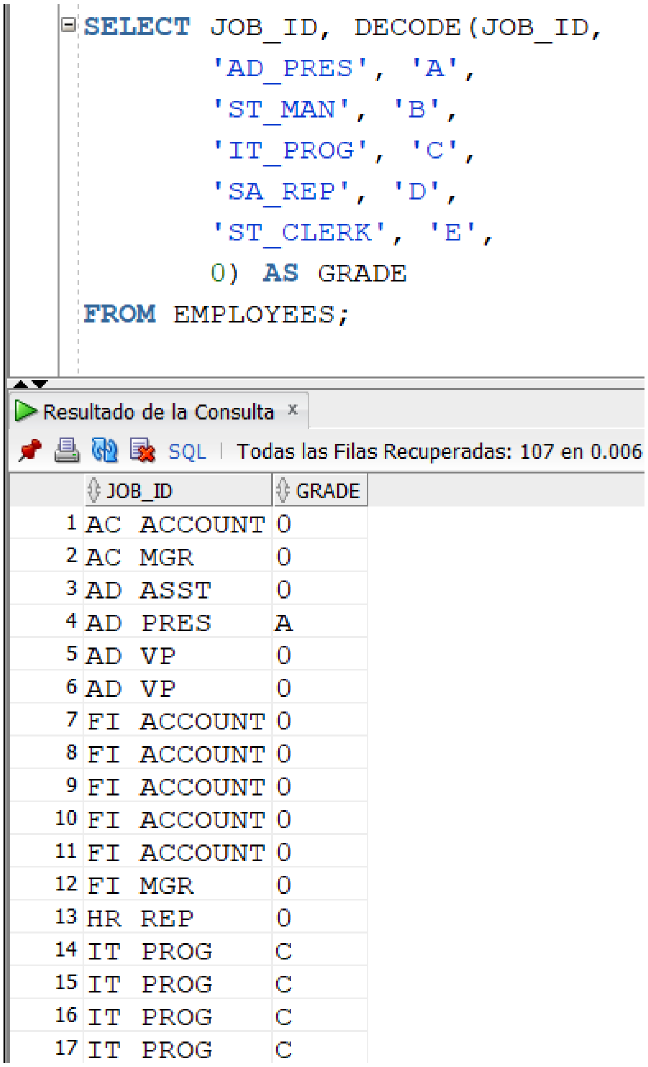
Se han recuperado 100 filas en 0.014 segundos

	LAST_NAME	COMM
34	Mallin	No Commission
35	Rogers	No Commission
36	Gee	No Commission
37	Philtanker	No Commission
38	Ladwig	No Commission
39	Stiles	No Commission
40	Seo	No Commission
41	Patel	No Commission
42	Rajs	No Commission
43	Davies	No Commission
44	Matos	No Commission
45	Vargas	No Commission
46	Russell	.4
47	Partners	.3
48	Errazuriz	.3

Figure 8: *SELECT statement, data projection.*

5. Using the DECODE function, write a query that displays the grade of all employees based on the value of the column JOB_ID, using the following data, **see figure 9**.

6. Rewrite the statement in the preceding exercise using the CASE syntax, **see figure 10**.

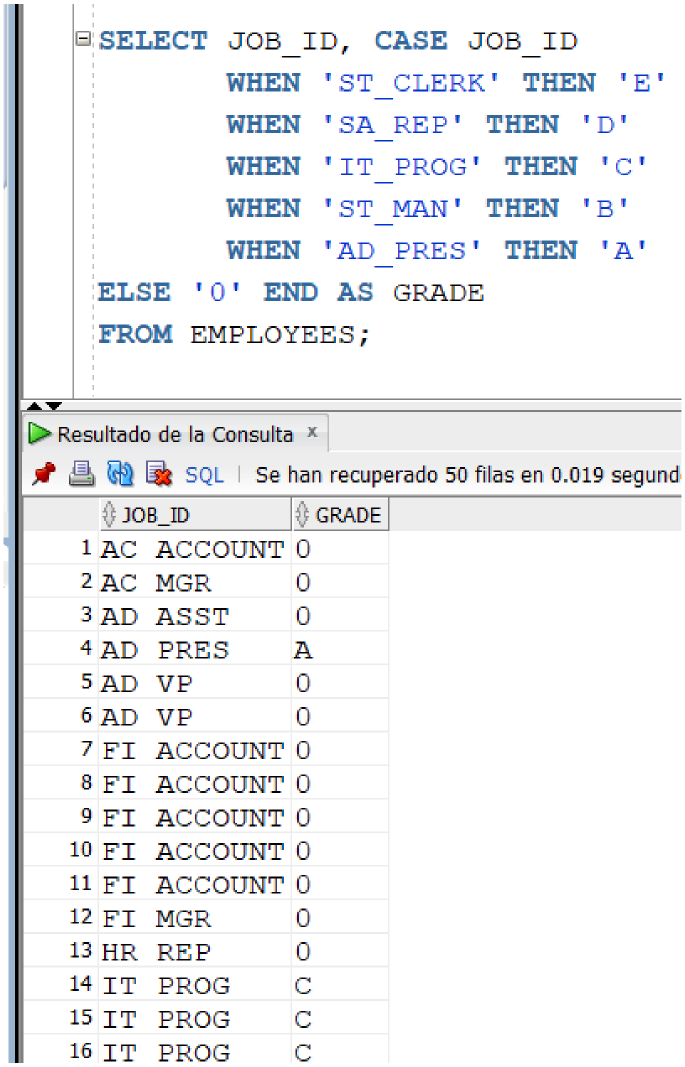


```
SELECT JOB_ID, DECODE(JOB_ID,
          'AD_PRES', 'A',
          'ST_MAN', 'B',
          'IT_PROG', 'C',
          'SA_REP', 'D',
          'ST_CLERK', 'E',
          0) AS GRADE
FROM EMPLOYEES;
```

Todas las Filas Recuperadas: 107 en 0.006

	JOB_ID	GRADE
1	AC ACCOUNT	0
2	AC MGR	0
3	AD ASST	0
4	AD PRES	A
5	AD VP	0
6	AD VP	0
7	FI ACCOUNT	0
8	FI ACCOUNT	0
9	FI ACCOUNT	0
10	FI ACCOUNT	0
11	FI ACCOUNT	0
12	FI MGR	0
13	HR REP	0
14	IT PROG	C
15	IT PROG	C
16	IT PROG	C
17	IT PROG	C

Figure 9: *SELECT statement, data projection.*



```
SELECT JOB_ID, CASE JOB_ID
        WHEN 'ST_CLERK' THEN 'E'
        WHEN 'SA_REP' THEN 'D'
        WHEN 'IT_PROG' THEN 'C'
        WHEN 'ST_MAN' THEN 'B'
        WHEN 'AD_PRES' THEN 'A'
ELSE '0' END AS GRADE
FROM EMPLOYEES;
```

Se han recuperado 50 filas en 0.019 segund

	JOB_ID	GRADE
1	AC ACCOUNT	0
2	AC MGR	0
3	AD ASST	0
4	AD PRES	A
5	AD VP	0
6	AD VP	0
7	FI ACCOUNT	0
8	FI ACCOUNT	0
9	FI ACCOUNT	0
10	FI ACCOUNT	0
11	FI ACCOUNT	0
12	FI MGR	0
13	HR REP	0
14	IT PROG	C
15	IT PROG	C
16	IT PROG	C

Figure 10: *SELECT statement, data projection.*

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 10 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.