Practice 10: Using Conversion Functions and Conditional Expressions

Jesús Manuel Juárez Pasillas

October 21, 2021

1 Introduction

The functions in Oracle allow us to carry out a great variety of operations with the data that we have, they also allow us to modify, represent them in another format, among other things. With these functions we can perform queries and obtain the results in some desired format, they also work for depending on the data that is consulted, the function does something and does not always return the same. This allows us to have a great variety of options with which to consult data.

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. What type of conversion is performed by the following statement?
   SELECT LENGTH(3.14285) FROM DUAL;

   A. Explicit conversion

   B. Implicit conversion

   C. TO_NUMBER function conversion

   D. **None of the above**

   It does not make any conversion, if not what it does is obtain the length of the number that was entered in this case it would return 7.

2. Choose any incorrect statements regarding conversion functions.

   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.**

The to_date function doesn't convert a date into a character string, that's what the to_char function is for.

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

A. 1234.49

B. 001234.5

C. 1234.5

D. **None of the above**

The instruction would give an error since the number is not in the indicated format.

4. What value is returned after executing the following statement?
SELECT TO_CHAR(1234.49, '999999.9') FROM DUAL;

A. 1234.49

B. 001234.5

C. **1234.5**

D. None of the above

It converts the number into a string of characters, in addition to rounding its decimals to a single decimal as indicated by the format, it also adds blank spaces to the left to complete the established format.

5. If SYSDATE returns 12-JUL-2009, what is returned by the following statement?
SELECT TO_CHAR(SYSDATE, 'fmMONTH, YEAR') FROM DUAL;

A. JUL, 2009

B. **JULY, TWO THOUSAND NINE**

C. JUL-09

D. None of the above

This returns the full name of the month followed by the name of the year in words, this due to the format in which it was requested to convert.

6. If SYSDATE returns 12-JUL-2009, what is returned by the following statement?
SELECT TO_CHAR(SYSDATE, 'fmDDth MONTH') FROM DUAL;

A. **12TH JULY**

B. 12th July

C. TWELFTH JULY

D. None of the above

Returns the number of the day of the month followed by the th format option and finally the full name of the month.

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;

   A. 2009

   B. **TWO THOUSAND NINE**

   C. 12-JUL-2009

   D. None of the above

   Returns the year in which it is, since when converting the date into a string, only the day is extracted, when converting it back into a date, only the day is taken, the rest is based on the current date (month and year) and the get the year, the current year is extracted, in this case 2009, obtaining its full name.

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

   A. **SPANIEL**

   B. TERRIER

   C. NULL

   D. None of the above

   It returns SPANIEL since the nullif function returns CODA as it is not equal to the other string, and the nvl2 function obtains the second parameter since the first 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;

   A. NO

   B. NULL

   C. **YES I AM**

   D. None of the above

As the substr function returns a null since there is no position 10 in the evaluated string, the function returns the second parameter of the nvl function, which in this case is "YES I AM".

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;

    A. TAX DUE

    B. **PARTY**

    C. 02

    D. None of the above

    Since the month of the date is not 02, the decode function returns the default option, since there is no option that is equal to the one returned.

2.2 Activity 2:

Propose an answer to the following issues:

- 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?
  - **If with the to_char function and with the 'dd/mm' format, you can extract the day and month from the column and the current date and thus compare them.**
- 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?
  - **Yes, with the function to_char (amount, '$999,999,999.99pr'), the pr places the number between <> if the amount is negative.**
- 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?
  - **Yes, converting the data to date type data with the format to_date (year, 'yyyy'), and it must also be taken into account that the current date is January of the current year since as only the year is extracted, the month is extracted from the current date and the day if it takes the value of 1.**

- Are nested functions evaluated from the outermost level to the innermost level?
  - **No, they are evaluated from the internal to the external level, since the output of the innermost function is the input of the function that nests it.**
- Must all functions in a nested expression return the same data type?
  - **No, it should only return the type of data that the function that nests it needs, since there are conversion functions which change the type of data that is being handled.**
- Is there a simpler way to display the SALARY information from the EMPLOYEES table in the form $19,000 without using the following statement?
  SELECT '$'|| SUBSTR(SALARY,1, MOD(LENGTH(SALARY),3))||','||
  SUBSTR(SALARY, MOD (LENGTH(SALARY),3)+1)
  - **If with the to_char (salary, 'fm$999,999') function it returns the same value as the long confusing statement used.**

2.3 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.

- Select CUST_FIRST_NAME, CUST_LAST_NAME, CUST_EMAIL, DATE_OF_BIRTH,
  (decode (to_char(sysdate,'dd/mm'),
  to_char((date_of_birth+2),'dd/mm'), 'Day before yesterday',
  to_char((date_of_birth+1),'dd/mm'), 'Yesterday',
  to_char(date_of_birth,'dd/mm'), 'Today',
  to_char((date_of_birth-1),'dd/mm'), 'Tomorrow',
  to_char((date_of_birth-2),'dd/mm'), 'Day after tomorrow',
  'Later this week')) BIRTHDAY
  from Customers where to_char(date_of_birth,'dd') between
  to_char((sysdate-2),'dd') and to_char((sysdate+7),'dd')
  and to_char(date_of_birth,'mm') between to_char((sysdate-2),'mm') and
  to_char((sysdate+7),'mm') order by to_char(date_of_birth,'dd/mm') asc;

	CUST_FIRST_NAME	CUST_LAST_NAME	CUST_EMAIL	DATE_OF_BIRTH	BIRTHDAY
1	Ajay	Andrews	Ajay.Andrews@YELLOWTHROAT.COM	21/10/56	Today
2	Lauren	Dench	Lauren.Dench@LONGSPUR.COM	22/10/75	Tomorrow
3	Kevin	Goodman	Kevin.Goodman@WIGEON.COM	22/10/48	Tomorrow
4	Billy	Hershey	Billy.Hershey@BULBUL.COM	22/10/87	Tomorrow
5	Elia	Brando	Elia.Brando@JUNCO.COM	23/10/51	Day after tomorrow
6	Donald	Minnelli	Donald.Minnelli@SCAUP.COM	26/10/53	Later this week
7	Ingrid	Welles	Ingrid.Welles@TEAL.COM	27/10/39	Later this week
8	Alec	Idle	Alec.Idle@EIDER.COM	27/10/64	Later this week
9	Roy	Dunaway	Roy.Dunaway@WHIMBREL.COM	28/10/55	Later this week
10	Geraldine	Martin	Geraldine.Martin@SCOTER.COM	28/10/27	Later this week

Figure 1: Consult the oe scheme.

2.4 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.
  - **Select first_name, last_name, to_char(hire_date, 'fmDay, "the" ddth "of" Month, Year') Satrt_date from Employees where to_char(hire_date,'fmDay') = 'Sábado';**

	FIRST_NAME	LAST_NAME	SATRT_DATE
1	Lex	De Haan	Sábado, the 13th of Enero, Two Thousand One
2	David	Austin	Sábado, the 25th of Junio, Two Thousand Five
3	Nancy	Greenberg	Sábado, the 17th of Agosto, Two Thousand Two
4	Den	Raphaely	Sábado, the 7th of Diciembre, Two Thousand Two
5	Shelli	Baida	Sábado, the 24th of Diciembre, Two Thousand Five
6	Julia	Nayer	Sábado, the 16th of Julio, Two Thousand Five
7	Steven	Markle	Sábado, the 8th of Marzo, Two Thousand Eight
8	Laura	Bissot	Sábado, the 20th of Agosto, Two Thousand Five
9	Michael	Rogers	Sábado, the 26th of Agosto, Two Thousand Six
10	Curtis	Davies	Sábado, the 29th of Enero, Two Thousand Five

Figure 2: Employees hired on a Saturday.

- 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.

- **Select first_name, last_name,**
  decode(length(first_name),
  length(last_name), 'Same Length',
  'Different Length') name_lengths
  from Employees where department_id = 100;

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

Figure 3: Name Lengths.

- 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.

  - **Select state_province,**
    decode(state_province,
    'Washington', 'Headquarters',
    'Texas', 'Oil Wells',
    'California', city,
    'New Jersey', street_address)Location_info
    from Locations where country_id = 'US' order by location_info;

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

Figure 4: Location info.

2.5 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.

   - **Select (last_name||' earns '||to_char(salary,'fm$999,999')||' monthly but wants '||to_char(salary*3,'fm$999,999')) "Dream Salaries" from Employees;**

	Dream Salaries
1	King earns $24,000 monthly but wants $72,000
2	Kochhar earns $17,000 monthly but wants $51,000
3	De Haan earns $17,000 monthly but wants $51,000
4	Hunold earns $9,000 monthly but wants $27,000
5	Ernst earns $6,000 monthly but wants $18,000
6	Austin earns $4,800 monthly but wants $14,400
7	Pataballa earns $4,800 monthly but wants $14,400
8	Lorentz earns $4,200 monthly but wants $12,600
9	Greenberg earns $12,008 monthly but wants $36,024
10	Faviet earns $9,000 monthly but wants $27,000

Figure 5: Location info.

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."

   - **Select last_name, hire_date, to_char(next_day(add_months(hire_date,6),1), 'fmDay, "the" ddth "of" Month, yyyy') Review from Employees;**

	LAST_NAME	HIRE_DATE	REVIEW
1	King	17/06/03	Lunes, the 22nd of Diciembre, 2003
2	Kochhar	21/09/05	Lunes, the 27th of Marzo, 2006
3	De Haan	13/01/01	Lunes, the 16th of Julio, 2001
4	Hunold	03/01/06	Lunes, the 10th of Julio, 2006
5	Ernst	21/05/07	Lunes, the 26th of Noviembre, 2007
6	Austin	25/06/05	Lunes, the 26th of Diciembre, 2005
7	Pataballa	05/02/06	Lunes, the 7th of Agosto, 2006
8	Lorentz	07/02/07	Lunes, the 13th of Agosto, 2007
9	Greenberg	17/08/02	Lunes, the 24th of Febrero, 2003
10	Faviet	16/08/02	Lunes, the 17th of Febrero, 2003

Figure 6: Salary review.

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.

   - **Select last_name, hire_date, to_char(hire_date,'fmDay') "Day" from Employees order by mod(to_char(hire_date,'D')+6,7);**

	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
14	King	17/06/03	Martes
15	Hartstein	17/02/04	Martes

Figure 7: Day of the week on which the employee started.

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.

   - **Select last_name,**
     decode(commission_pct,
     null, 'No Commission',
     commission_pct) comm
     from Employees;

43	Davies	No Commission
44	Matos	No Commission
45	Vargas	No Commission
46	Russell	.4
47	Partners	.3
48	Errazuriz	.3
49	Cambrault	.3
50	Zlotkey	.2
51	Tucker	.3
52	Bernstein	.25
53	Hall	.25

Figure 8: Commission amounts.

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:

   - **Select job_id,**
     decode(job_id,
     'AD_PRES', 'A',
     'ST_MAN', 'B',
     'IT_PROG', 'C',
     'SA_REP', 'D',
     'ST_CLERK', 'E',
     0) Grade
     from Employees;

	JOB_ID	GRADE
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

Figure 9: Decode grade.

6. Rewrite the statement in the preceding exercise using the CASE syntax.

- **Select job_id,**
  case job_id
  when 'AD_PRES' then 'A'
  when 'ST_MAN' then 'B'
  when 'IT_PROG' then 'C'
  when 'SA_REP' then 'D'
  when 'ST_CLERK' then 'E'
  else '0' end Grade
  from Employees;

	JOB_ID	GRADE
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

Figure 10: Case grade.

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 functions give us a large number of options with which to manipulate the data that we have, such as the data that is stored and consulted, this allows us to have a great variety of options with which to be able to display the data in different ways. Also, that with these functions it is much easier to manipulate the data, such as wanting to change the data type, convert an element into something more detailed, etc.