

Practice 9: Data selection and projection: single row functions

Jesús Manuel Juárez Pasillas

October 14, 2021

1 Introduction

The functions in Oracle are very useful when making queries, since you will want to consult the data and manipulate it so that it appears in a special way to what you want. Without the functions it would be very difficult to do all this since it would have to be done by a person data by data and it would be too time consuming and could contain possible errors.

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. Which statements regarding single-row functions are true?
 - A. They may return more than one result.
 - B. **They execute once for each record processed.**
 - C. They may have zero or more input parameters.
 - D. **They must have at least one mandatory parameter.**

Single-line functions run once for each item processed, and they must have at least one required parameter, otherwise the function would do nothing.

2. Which of these are single-row character-case conversion functions?
 - A. **LOWER**
 - B. SMALLER
 - C. INITCASE
 - D. **INITCAP**

The lower function converts the string to lowercase and the init-cap function makes the initial letters of each word uppercase.

3. What value is returned after executing the following statement:
SELECT LENGTH('How_long_is_a_piece_of_string?') FROM DUAL;
- A. 29
 - B. **30**
 - C. 24
 - D. None of the above

Returns 30 since it is the number of characters that the string contains.

4. What value is returned after executing the following statement:
SELECT SUBSTR('How_long_is_a_piece_of_string?', 5,4) FROM DUAL;
- A. **long**
 - B. _long
 - C. string?
 - D. None of the above

Returns "long" since it will extract 4 characters starting from character 5.

5. What value is returned after executing the following statement?
SELECT INSTR('How_long_is_a_piece_of_string?', '_',5,3) FROM DUAL;
- A. 4
 - B. **14**
 - C. 12
 - D. None of the above

It gets 14 as it evaluates the string starting from character 5 looking for the third underscore it finds, returning the position it is in.

6. What value is returned after executing the following statement?
SELECT REPLACE('How_long_is_a_piece_of_string?', '_','') FROM DUAL;
- A. How long is a piece of string?
 - B. How_long_is_a_piece_of_string?
 - C. **Howlongisapieceofstring?**
 - D. None of the above

It gets the string, but without the hyphens since I replace them with nothing, since nothing was put in single quotes in the function argument.

7. What value is returned after executing the following statement?
SELECT MOD(14,3) FROM DUAL;

A. 3
B. 42
C. 2
D. None of the above

Returns 2 which is the remainder of the division of 14 by 3 without decimals.

8. Assuming SYSDATE=07-JUN-1996 12:05pm, what value is returned after executing the following statement?
SELECT ADD_MONTHS(SYSDATE,-1) FROM DUAL;

A. 07-MAY-1996 12:05pm
B. 06-JUN-1996 12:05pm
C. 07-JUL-1996 12:05pm
D. None of the above

The result will be the same day, year and time as the original, only the month that will be one before the original will change by -1 in the function.

9. What value is returned after executing the following statement? Take note that 01-JAN-2009 occurs on a Thursday.
SELECT NEXT_DAY('01-JAN-2009','wed') FROM DUAL;

A. 07-JAN-2009
B. 31-JAN-2009
C. Wednesday
D. None of the above

When you want to find out which day is the following Wednesday from that date, the days are counted and they are increased to the given date.

10. Assuming SYSDATE=30-DEC-2007, what value is returned after executing the following statement?
SELECT TRUNC(SYSDATE,'YEAR') FROM DUAL;

A. 31-DEC-2007

- B. 01-JAN-2008
- C. 01-JAN-2007
- D. None of the above

As the year is truncated, the date is made as if it had just begun that year, that is, 01-Jan-2007.

2.2 Activity 2:

Propose an answer to the following issues:

- You would like to search for a character string stored in the database. The case in which it is stored is unknown and there are potentially leading and trailing spaces surrounding the string. Can such a search be performed?
 - **Making comparisons with a like in the condition, where the string to search is between % signs so that it only evaluates if the string is inside or not.**
- You have been asked to extract the last three characters from the LAST_NAME column in the EMPLOYEES table. Can such a query be performed without using the LENGTH function?
 - **You cannot, since to get the last characters you have to know the length of the string, since not all surnames have the same length.**
- You would like to extract a consistent 11-character string based on the SALARY column in the EMPLOYEES table. If the SALARY value is less than 11 characters long, zeros must be added to the left of the value to yield a 11-character string. Is this possible?
 - **Yes, with the LPAD function it is possible to fill the string up to 11 characters always with the same character provided (0 in this case).**
- You wish to retrieve the duration of employment in days for each employee. Is it possible to perform such a calculation?
 - **Yes, with a simple subtraction between the hire date and the termination date, this returns the number of days difference between both dates.**
- You are tasked with identifying the date the end of year staff bonus will be paid. Bonuses are usually paid on the last Friday in December. Can the bonus date be computed using the NEXT_DAY function?
 - **Yes, positioning in the last 7 days of the month of December (25-Dec-Year) and obtaining the following Friday from that date.**

- Employees working in the IT department have moved to new offices and, although the last four digits of their phone numbers are the same, the set of the three digits 324 is changed to 326. A typical phone number of an IT staff member is 140-324-3489. You are required to provide a list of employees' names with their old and new phone numbers. Can this list be provided?
 - If, comparing with a like in the resulting string being the condition: (phone_number like '%324_____'), this evaluates that the string contains anything starting, a 324 followed by 5 characters which 1 would be the hyphen, then the 4 latest issues.

2.3 Activity 3:

Connect to the OE schema and complete the following tasks.

- Several quotations were requested for prices on color printers. The supplier information is not available from the usual source, but you know that the supplier identification number is embedded in the CATALOG_URL column from the PRODUCT_INFORMATION table. You are required to retrieve the PRODUCT_NAME and CATALOG_URL values and to extract the supplier number from the CATALOG_URL column for all products which have both the words COLOR and PRINTER in the PRODUCT_DESCRIPTION column stored in any case.
 - SELECT PRODUCT_NAME, CATALOG_URL,
SUBSTR(CATALOG_URL, 17, 6) AS SUPPLIER_ID FROM
PRODUCT_INFORMATION WHERE
LOWER(PRODUCT_DESCRIPTION) LIKE '%color%' AND
LOWER(PRODUCT_DESCRIPTION) LIKE '%printer%';

	PRODUCT_NAME	CATALOG_URL	SUPPLIER_ID
1	Inkjet C/8/HQ	http://www.supp-102094.com/cat/hw/pl797.html	102094
2	Industrial 600/DQ	http://www.supp-102088.com/cat/hw/pl792.html	102088
3	Inkjet C/4	http://www.supp-102090.com/cat/hw/p2453.html	102090

Figure 1: Consult the oe scheme.

2.4 Activity 4:

This exercise must be performed using HR schema.

- Retrieve a list of all FIRST_NAME and LAST_NAME values from the EMPLOYEES table where FIRST_NAME contains the character string "li." 1. Start SQL Developer and connect to the HR schema. The data

filter must compare the FIRST_NAME column values with a pattern of characters containing all possible case combinations of the string “li.” Therefore, if the FIRST_NAME contains the character strings “LI,” “Li,” “lI,” or “li,” that row must be retrieved. The LIKE operator is used for character matching, and four combinations can be extracted with four WHERE clauses separated by the OR keyword. However, the case conversion functions can simplify the condition.

- Select first_name, last_name from Employees
where lower(first_name) like '%li%';

	FIRST_NAME	LAST_NAME
1	Shelli	Baida
2	Elizabeth	Bates
3	Julia	Dellinger
4	William	Gietz
5	Julia	Nayer
6	Lisa	Ozer
7	Valli	Pataballa
8	Lindsey	Smith
9	William	Smith
10	Oliver	Tuvault

Figure 2: FIRST NAME contains the character string “li”.

- Envelope printing restricts the addressee field to 16 characters. Ideally, the addressee field contains employees’ FIRST_NAME and LAST_NAME values separated by a single space. When the combined length of an employee’s FIRST_NAME and LAST_NAME exceeds 15 characters, the addressee field should contain their formal name. An employee’s formal name is made up of the first letter of their FIRST_NAME and the first 14 characters of their LAST_NAME. You are required to retrieve a list of FIRST_NAME and LAST_NAME values and formal names for employees where the combined length of FIRST_NAME and LAST_NAME exceeds 15 characters.

- Select
case length(first_name||last_name) when 15 then first_name||' '||last_name
when 14 then first_name||' '||last_name
when 13 then first_name||' '||last_name
when 12 then first_name||' '||last_name
when 11 then first_name||' '||last_name
when 10 then first_name||' '||last_name
when 9 then first_name||' '||last_name
when 8 then first_name||' '||last_name
when 7 then first_name||' '||last_name

```

when 6 then first_name||' '||last_name
when 5 then first_name||' '||last_name
when 4 then first_name||' '||last_name
when 3 then first_name||' '||last_name
when 2 then first_name||' '||last_name
when 1 then first_name||' '||last_name
else substr(first_name,1,1)||' '||rpad(last_name,14,' ') end name
from Employees;

```

	NAME
1	Ellen Abel
2	Sundar Ande
3	Mozhe Atkinson
4	David Austin
5	Hermann Baer
6	Shelli Baida
7	Amit Banda
8	Elizabeth Bates
9	Sarah Bell
10	David Bernstein
11	Laura Bissot
12	Harrison Bloom

Figure 3: First name, last name and formal name.

- You are required to obtain a list of EMPLOYEE_ID, LAST_NAME, and HIRE_DATE values (add the MONTHS WORKED by the employees) for the employees who have worked more than 90 months between the date they were hired and 01-JAN-2000 (The year was changed to 2010, because there were no records with the hiring date before 2000).
 - Select employee_id, last_name, hire_date, to_char(months_between(sysdate, hire_date), '999,999.00') as months_worked from Employees where months_between('01/01/10', hire_date) >= 90;

	EMPLOYEE_ID	LAST_NAME	HIRE_DATE	MONTHS_WORKED
1	102	De Haan	13/01/01	249.00
2	203	Mavris	07/06/02	232.21
3	204	Baer	07/06/02	232.21
4	205	Higgins	07/06/02	232.21
5	206	Gietz	07/06/02	232.21

Figure 4: Months worked.

- You are required to display employee first names and last names joined together, the length of the employee last name, and the numeric position of the letter “a” in the employee last name for all employees whose last names end with the letter “n.” Use Oracle functions to perform the whole sentence.

- Select first_name||' '||last_name name, length(last_name) as
”Length last name”, instr(lower(last_name),’a’) as
”Position of the a” from Employees where last_name like '%n';

	NAME	Length last name	Position of the a
1	Mozhe Atkinson	8	1
2	David Austin	6	1
3	David Bernstein	9	0
4	John Chen	4	0
5	Lex De Haan	7	5
6	Louise Doran	5	4
7	Michael Hartstein	9	2
8	Alyssa Hutton	6	0
9	Charles Johnson	7	0
10	Jack Livingston	10	0

Figure 5: Employees whose last names end with the letter “n”.

- Display the employee number, hire date, number of months employed, six-month review date, first Friday after hire date, and the last day of the hire month for all employees who have been employed for fewer than 150 months.
 - Select employee_id, hire_date, to_char(months_between('01/01/10', hire_date), '999,999.00') as ”Months employed”,
add_months(hire_date,6) as ”Review date”,
next_day(hire_date,5) as ”First Friday”, last_day(hire_date) as
”Last day” from Employees where
months_between('01/01/10',hire_date) <= 150;

	EMPLOYEE_ID	HIRE_DATE	Months employed	Review date	First Friday	Last day
1	100	17/06/03	78.48	17/12/03	20/06/03	30/06/03
2	101	21/09/05	51.35	21/03/06	23/09/05	30/09/05
3	102	13/01/01	107.61	13/07/01	19/01/01	31/01/01
4	103	03/01/06	47.94	03/07/06	06/01/06	31/01/06
5	104	21/05/07	31.35	21/11/07	25/05/07	31/05/07
6	105	25/06/05	54.23	25/12/05	01/07/05	30/06/05
7	106	05/02/06	46.87	05/08/06	10/02/06	28/02/06
8	107	07/02/07	34.81	07/08/07	09/02/07	28/02/07
9	108	17/08/02	88.48	17/02/03	23/08/02	31/08/02
10	109	16/08/02	88.52	16/02/03	23/08/02	31/08/02

Figure 6: Employees who have been employed for fewer than 150 months.

- Compare the hire dates for all employees who started in 1997. Display the employee number and hire date (The year was changed from 1997 to 2005 because the records contain dates after 2000).
 - Select employee_id, hire_date from Employees where extract(year from hire_date) = 2005;

	EMPLOYEE_ID	HIRE_DATE
1	101	21/09/05
2	105	25/06/05
3	110	28/09/05
4	111	30/09/05
5	116	24/12/05
6	117	24/07/05
7	121	10/04/05
8	123	10/10/05
9	125	16/07/05
10	129	20/08/05

Figure 7: Employees who started in 2005.

2.5 Activity 5:

This activity provides a variety of exercises using different functions that are available for character, number, and date data types.

1. Write a query to display the system date. Label the column as Date.
Select sysdate as "Date" from Dual;

	Date
1	14/10/21

Figure 8: System date.

2. The HR department needs a report to display the employee number, last name, salary, and salary increased by 15.5% (expressed as a whole number) for each employee. Label the column New Salary. Save your SQL statement in a file named lab_9_02.sql.
The file is located inside the file "Plugins.zip"
3. Run your query in the lab_9_02.sql file.

	EMPLOYEE_ID	LAST_NAME	SALARY	New Salary
1	100	King	24000	27,720
2	101	Kochhar	17000	19,635
3	102	De Haan	17000	19,635
4	103	Hunold	9000	10,395
5	104	Ernst	6000	6,930
6	105	Austin	4800	5,544
7	106	Pataballa	4800	5,544
8	107	Lorentz	4200	4,851
9	108	Greenberg	12008	13,869
10	109	Faviet	9000	10,395

Figure 9: Run lab_9_02.sql file.

4. Modify your query lab_9_02.sql to add a column that subtracts the old salary from the new salary. Label the column Increase. Save the contents of the file as lab_9_04.sql. Run the revised query.
The file is located inside the file "Plugins.zip"

	EMPLOYEE_ID	LAST_NAME	SALARY	New Salary	Increase
1	100	King	24000	27,720	3720
2	101	Kochhar	17000	19,635	2635
3	102	De Haan	17000	19,635	2635
4	103	Hunold	9000	10,395	1395
5	104	Ernst	6000	6,930	930
6	105	Austin	4800	5,544	744
7	106	Pataballa	4800	5,544	744
8	107	Lorentz	4200	4,851	651
9	108	Greenberg	12008	13,869	1861.24
10	109	Faviet	9000	10,395	1395

Figure 10: Run lab_9.04.sql file.

- Write a query that displays the last name (with the first letter in uppercase and all the other letters in lowercase) and the length of the last name for all employees whose name starts with the letters “J,” “A,” or “M.” Give each column an appropriate label. Sort the results by the employees’ last names.

Select initcap(last_name) ”Last Name”, length(last_name)
”Length” **from** Employees **where** last_name like ’J%’
or last_name like ’A%’ **or** last_name like ’M%’;

	Last Name	Length
1	Abel	4
2	Ande	4
3	Atkinson	8
4	Austin	6
5	Johnson	7
6	Jones	5
7	Mallin	6
8	Markle	6
9	Marlow	6
10	Marvins	7

Figure 11: Employees whose name starts with “J,” “A,” or “M.”.

Rewrite the query so that the user is prompted to enter a letter that the last name starts with.

Select initcap(last_name) ”Last Name”, length(last_name) ”Length”
from Employees **where** last_name like ’&Start_Letter%’;

	Last Name	Length
1	Hall	4
2	Hartstein	9
3	Higgins	7
4	Himuro	6
5	Hunold	6
6	Hutton	6

Figure 12: User is prompted to enter a letter.

Modify the query such that the case of the entered letter does not affect the output. The entered letter must be capitalized before being processed by the SELECT query.

Select initcap(last_name) "Last Name", length(last_name) "Length"
from Employees where last_name like upper('&Start_Letter%');

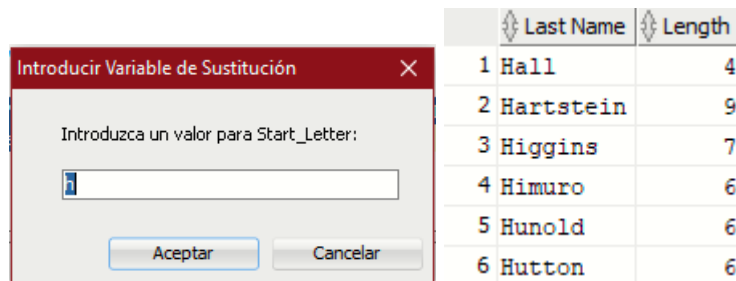


Figure 13: Case letter does not affect the output.

- The HR department wants to find the duration of employment for each employee. For each employee, display the last name and calculate the number of months between today and the date on which the employee was hired. Label the column as MONTHS_WORKED. Order your results by the number of months employed. Round the number of months up to the closest whole number.

Select last_name, round(months_between(sysdate, hire_date))
MONTHS_WORKED from Employees order by
months_between(sysdate,hire_date);

LAST_NAME	MONTHS_WORKED
1 Kumar	162
2 Banda	162
3 Ande	163
4 Markle	163
5 Lee	164
6 Philtanker	164
7 Geoni	164
8 Zlotkey	165
9 Marvins	165
10 Grant	165

Figure 14: Number of months between today and hire date.

- Create a query to display the last name and salary for all employees. Format the salary to be 15 characters long, left-padded with the \$ symbol. Label the column as SALARY.

Select last_name, lpad(salary,15,'\$') salary from Employees;

LAST_NAME	SALARY
1 King	\$\$\$\$\$\$\$\$\$24000
2 Kochhar	\$\$\$\$\$\$\$\$\$17000
3 De Haan	\$\$\$\$\$\$\$\$\$17000
4 Hunold	\$\$\$\$\$\$\$\$\$9000
5 Ernst	\$\$\$\$\$\$\$\$\$6000
6 Austin	\$\$\$\$\$\$\$\$\$4800
7 Pataballa	\$\$\$\$\$\$\$\$\$4800
8 Lorentz	\$\$\$\$\$\$\$\$\$4200
9 Greenberg	\$\$\$\$\$\$\$\$\$12008
10 Faviet	\$\$\$\$\$\$\$\$\$9000

Figure 15: Format salary to be 15 characters with the \$ symbol.

- Create a query that displays the first eight characters of the employees' last names and indicates the amounts of their salaries with asterisks. Each asterisk signifies a thousand dollars. Sort the data in descending order of salary. Label the column as EMPLOYEES_AND_THEIR_SALARIES.
Select (rpad(last_name, 8, ' ') || (rpad(' ', round(salary/1000), '*'))) as EMPLOYEES_AND_THEIR_SALARIES
from Employees order by salary desc;

	EMPLOYEES_AND_THEIR_SALARIES
1	King *****
2	Kochhar *****
3	De Haan *****
4	Russell *****
5	Partners *****
6	Hartstei *****
7	Greenber *****
8	Higgins *****
9	Errazuri *****
10	Ozer *****

Figure 16: Each asterisk signifies a thousand dollars.

- Create a query to display the last name and the number of weeks employed for all employees in department 90. Label the number of weeks column as TENURE. Truncate the number of weeks value to 0 decimal places. Show the records in descending order of the employee's tenure.
Select last_name, trunc((sysdate-hire_date)/7,0) tenencia from Employees where department_id = 80 order by (sysdate-hire_date)/7 desc;

	LAST_NAME	TENENCIA
1	King	923
2	Sully	919
3	Abel	909
4	McEwen	897
5	Russell	888
6	Partners	875
7	Tucker	871
8	Errazuriz	866
9	Smith	866
10	Ozer	865

Figure 17: Tenure.

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:

Many functions by themselves result in a lot of work for the person who makes the query, combining several functions can reduce the sentence so that it is much easier to understand, and much more efficient when executing it.

The entire structure of a select statement is very important to obtain reports that satisfy one or more needs, for this it is necessary to evaluate what is needed, also to evaluate what parts of the query are needed and what functions could be useful to improve the statement.