



Usando Funciones SQL de una Fila de Conversión y Validar Valores Nulos

MDY2131

Experiencia de Aprendizaje y Competencia Asociada



ESCUELA DE
INFORMÁTICA Y
TELECOMUNICACIONES

Experiencia	Nombre	Unidad de Competencia Especialidad – Nivel de la Competencia de Empleabilidad
Nº 1	Construyendo sentencias simples de recuperación de datos	Desarrolla operaciones sobre la base de datos que permitan la obtención, actualización, inserción y eliminación de información estableciendo una conexión con una base de datos para cumplir con los requerimientos de la organización.
		Resolución de Problemas (N1)

- Cómo convertir en forma explícita una fecha a un formato caracter en una sentencia SQL.
- Cómo convertir en forma explícita un número a un formato caracter en una sentencia SQL.
- Cómo convertir en forma explícita un caracter a una fecha en una sentencia SQL.
- Cómo convertir en forma explícita un caracter a un número en una sentencia SQL.
- Cómo definir un valor cuando alguna expresión es NULA en una sentencia SQL.
- Cómo trabajar con Funciones Anidadas en una sentencia SQL.
- Cómo implementar una condición lógica IF-THEN-ELSE en una sentencia SQL.

Funciones de Conversión de Tipos de Datos

*Las sentencias de los ejemplos usan las tablas del esquema HR de la Base de Datos Oracle



Conversión de Tipos de Datos Implícita

DE	A
VARCHAR2 o CHAR	NUMBER
VARCHAR2 o CHAR	DATE
NUMBER	VARCHAR2
DATE	VARCHAR2

- Ejemplo:

```
SELECT employee_id, hire_date
FROM employees
WHERE hire_date > '05/03/2008'
ORDER BY hire_date;
```

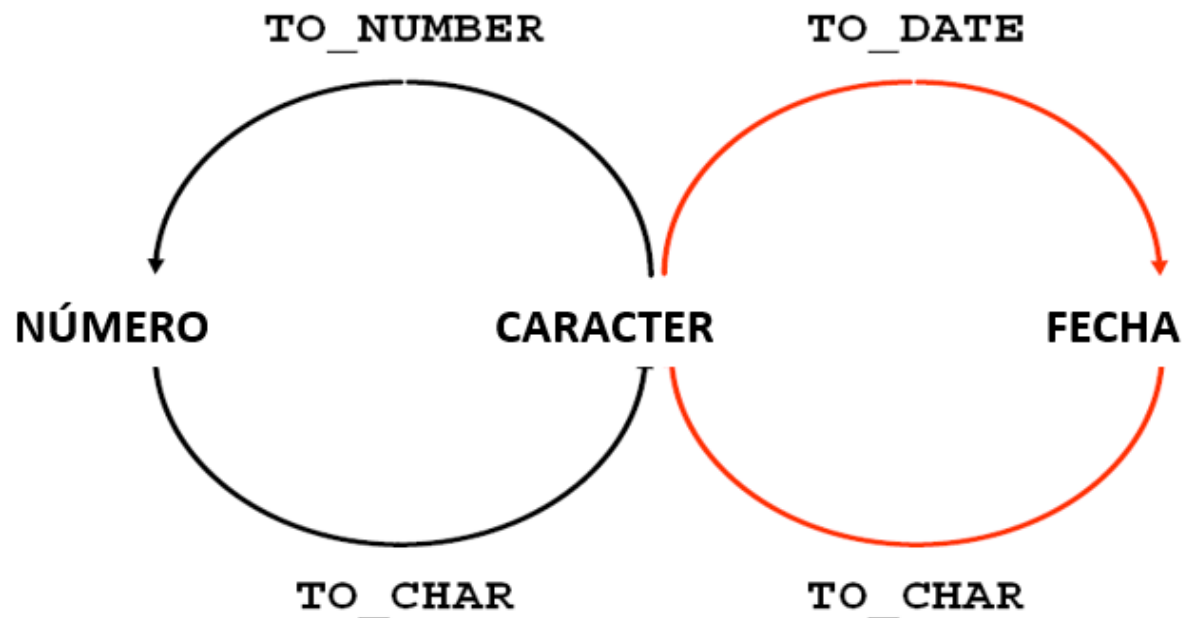
	EMPLOYEE_ID	HIRE_DATE
1	128	08/03/2008
2	166	24/03/2008
3	167	21/04/2008
4	173	21/04/2008

Conversión de Tipos de Datos Explícita

TO_CHAR(*número* | *fecha* [, 'formato_conversión '])

TO_NUMBER(*caracter* [, 'formato_conversión '])

TO_DATE(*caracter* [, 'formato_conversión '])



Uso Función TO_CHAR con Fechas

TO_CHAR puede convertir la fecha desde su formato por defecto a un formato especificado por el usuario

- Sintaxis:

```
TO_CHAR(fecha, 'formato_conversión')
```

- Principales elementos de fechas válidos para los formatos de fechas:

YYYY	YEAR	MM	MONTH	MON	DY	DAY	DD	Q
------	------	----	-------	-----	----	-----	----	---

- Principales elementos de horas válidos para los formatos de fechas:

AM o PM	HH	HH12	HH24	MI	SS	/ . ,	"caracteres_a_visualizar"
------------	----	------	------	----	----	-------	---------------------------

Uso Función TO_CHAR con Fechas

- Ejemplo:

```
SELECT last_name "Apellido", TO_CHAR(hire_date, 'dd/mm/yyyy') "Formato Fecha 1",
      TO_CHAR(hire_date, 'DD Month YYYY') "Formato Fecha 2",
      TO_CHAR(SYSDATE, 'dd "de" MONTH "del" yyyy hh24:mi:ss') "Fecha/Hora Sistema 1",
      TO_CHAR(SYSDATE, 'dd "de" MONTH "del" yyyy hh12:mi:ss AM') "Fecha/Hora Sistema 2"
FROM employees;
```

Apellido	Formato Fecha 1	Formato Fecha 2	Fecha/Hora Sistema 1	Fecha/Hora Sistema 2
1 King	17/06/2003	17 Junio 2003	11 de OCTUBRE del 2019 15:21:44	11 de OCTUBRE del 2019 03:21:44 PM
2 Kochhar	21/09/2005	21 Septiembre 2005	11 de OCTUBRE del 2019 15:21:44	11 de OCTUBRE del 2019 03:21:44 PM
3 De Haan	13/01/2001	13 Enero 2001	11 de OCTUBRE del 2019 15:21:44	11 de OCTUBRE del 2019 03:21:44 PM
4 Hunold	03/01/2006	03 Enero 2006	11 de OCTUBRE del 2019 15:21:44	11 de OCTUBRE del 2019 03:21:44 PM
5 Ernst	21/05/2007	21 Mayo 2007	11 de OCTUBRE del 2019 15:21:44	11 de OCTUBRE del 2019 03:21:44 PM
6 Austin	25/06/2005	25 Junio 2005	11 de OCTUBRE del 2019 15:21:44	11 de OCTUBRE del 2019 03:21:44 PM
7 Pataballa	05/02/2006	05 Febrero 2006	11 de OCTUBRE del 2019 15:21:44	11 de OCTUBRE del 2019 03:21:44 PM
8 Lorentz	07/02/2007	07 Febrero 2007	11 de OCTUBRE del 2019 15:21:44	11 de OCTUBRE del 2019 03:21:44 PM
9 Greenberg	17/08/2002	17 Agosto 2002	11 de OCTUBRE del 2019 15:21:44	11 de OCTUBRE del 2019 03:21:44 PM
10 Faviet	16/08/2002	16 Agosto 2002	11 de OCTUBRE del 2019 15:21:44	11 de OCTUBRE del 2019 03:21:44 PM
.....				
104 Mavris	07/06/2002	07 Junio 2002	11 de OCTUBRE del 2019 15:21:44	11 de OCTUBRE del 2019 03:21:44 PM
105 Baer	07/06/2002	07 Junio 2002	11 de OCTUBRE del 2019 15:21:44	11 de OCTUBRE del 2019 03:21:44 PM
106 Higgins	07/06/2002	07 Junio 2002	11 de OCTUBRE del 2019 15:21:44	11 de OCTUBRE del 2019 03:21:44 PM
107 Gietz	07/06/2002	07 Junio 2002	11 de OCTUBRE del 2019 15:21:44	11 de OCTUBRE del 2019 03:21:44 PM

Uso Función TO_CHAR con Fechas

- Ejemplo:

```
SELECT last_name "Apellido", TO_CHAR(hire_date, 'dd/mm/yyyy') "Formato Fecha 1",
      TO_CHAR(hire_date, 'DD Month YYYY') "Formato Fecha 2",
      TO_CHAR(hire_date, 'Q') "Trimestre",
      TO_CHAR(SYSDATE, 'dd "de" MONTH "del" yyyy hh24:mi:ss') "Fecha/Hora Sistema 1",
      TO_CHAR(SYSDATE, 'dd "de" MONTH "del" yyyy hh12:mi:ss AM') "Fecha/Hora Sistema 2"
FROM employees;
```

Apellido	Formato Fecha 1	Formato Fecha 2	Trimestre	Fecha/Hora Sistema 1	Fecha/Hora Sistema 2
1 King	17/06/2003	17 Junio 2003	2	16 de OCTUBRE del 2019 18:37:26	16 de OCTUBRE del 2019 06:37:26 PM
2 Kochhar	21/09/2005	21 Septiembre 2005	3	16 de OCTUBRE del 2019 18:37:26	16 de OCTUBRE del 2019 06:37:26 PM
3 De Haan	13/01/2001	13 Enero 2001	1	16 de OCTUBRE del 2019 18:37:26	16 de OCTUBRE del 2019 06:37:26 PM
4 Hunold	03/01/2006	03 Enero 2006	1	16 de OCTUBRE del 2019 18:37:26	16 de OCTUBRE del 2019 06:37:26 PM
36 Gee	12/12/2007	12 Diciembre 2007	4	16 de OCTUBRE del 2019 18:37:26	16 de OCTUBRE del 2019 06:37:26 PM
37 Philtanker	06/02/2008	06 Febrero 2008	1	16 de OCTUBRE del 2019 18:37:26	16 de OCTUBRE del 2019 06:37:26 PM
38 Ladwig	14/07/2003	14 Julio 2003	3	16 de OCTUBRE del 2019 18:37:26	16 de OCTUBRE del 2019 06:37:26 PM
39 Stiles	26/10/2005	26 Octubre 2005	4	16 de OCTUBRE del 2019 18:37:26	16 de OCTUBRE del 2019 06:37:26 PM
104 Mavris	07/06/2002	07 Junio 2002	2	16 de OCTUBRE del 2019 18:37:26	16 de OCTUBRE del 2019 06:37:26 PM
105 Baer	07/06/2002	07 Junio 2002	2	16 de OCTUBRE del 2019 18:37:26	16 de OCTUBRE del 2019 06:37:26 PM
106 Higgins	07/06/2002	07 Junio 2002	2	16 de OCTUBRE del 2019 18:37:26	16 de OCTUBRE del 2019 06:37:26 PM
107 Gietz	07/06/2002	07 Junio 2002	2	16 de OCTUBRE del 2019 18:37:26	16 de OCTUBRE del 2019 06:37:26 PM

Uso Función TO_CHAR con Números

TO_CHAR puede convertir los valores numéricos a una cadena de caracteres en un formato especificado por el usuario

- Sintaxis:

TO_CHAR(*número*, 'formato_conversión')

- Principales elementos de formatos de números válidos:

9	0 (cero)	\$	L	G
. (punto)	, (coma)	D	V	

Uso Función TO_CHAR con Números

- Ejemplo:

```
SELECT salary, TO_CHAR(salary, '9999999') Formato1,  
             TO_CHAR(salary, '0000000') Formato2, TO_CHAR(salary, '$999G999') Formato3,  
             TO_CHAR(salary, 'L99,999.00') Formato4  
FROM employees  
WHERE last_name = 'Ernst';
```

SALARY	FORMATO1	FORMATO2	FORMATO3	FORMATO4
6000	6000	0006000	\$6.000	\$6,000.00

- Ejemplo:

```
SELECT salary, TO_CHAR(salary, '$0099,999.00') Formato1,  
             TO_CHAR(salary, '$99G999D00') Formato2, O_CHAR(salary, '99999V900') Formato3  
FROM employees  
WHERE last_name = 'King';
```

SALARY	FORMATO1	FORMATO2	FORMATO3
10000	\$0010,000.00	\$10.000,00	10000000
24000	\$0024,000.00	\$24.000,00	24000000

Uso de Función TO_NUMBER con Caracteres

TO_NUMBER puede convertir un caracter a un formato número

- Sintaxis:

TO_NUMBER(*caracter*, [' formato_conversión'])

- Ejemplo

**SELECT '1234,56' caracter, TO_NUMBER('1234,56') numero
FROM dual;**

⚡ CHARACTER	⚡ NUMERO
1234,56	1234,56

Uso de Función TO_DATE con Caracteres

TO_DATE puede convertir un caracter a un formato de fecha

- Sintaxis:

```
TO_DATE(caracter, [' formato_conversión'])
```

- Ejemplo


```
SELECT to_date('2007/05/31') FROM dual;
```

ORA-01861: el literal no coincide con la cadena de formato
01861. 00000 - "literal does not match format string"

*Cause: Literals in the input must be the same length as literals in the format string (with the exception of leading whitespace). If the "FX" modifier has been toggled on, the literal must match exactly, with no extra whitespace.

*Action: Correct the format string to match the literal.

```
SELECT to_date('2007/05/31','YYYY/MM/DD') FROM dual;
```

 TO_DATE('2007/05/31','YYYY/MM/DD')
31/05/2007

Uso de Función TO_DATE con Caracteres

- Ejemplo:

```
SELECT TO_CHAR('01/08/2019','YYYY') "USO TO_DATE CON CARACTERES"  
FROM dual;
```

```
ORA-01722: número no válido  
01722. 00000 - "invalid number"  
*Cause: The specified number was invalid.  
*Action: Specify a valid number.
```

```
SELECT TO_CHAR(TO_DATE('01/08/2019'),'YYYY') "USO TO_DATE CON CARACTERES"  
FROM dual;
```

```
USE TO_DATE CON CARACTERES  
2019
```


Funciones para validar valores Nulos



Trabajan con cualquier tipo de dato y permiten definir valores a utilizar en el caso de que las expresiones tengan valor nulo.

NVL(*expr1*, *expr2*)

NVL2(*expr1*, *expr2*, *expr3*)

NULLIF(*expr1*, *expr2*)

COALESCE(*expr1*, *expr2*, ..., *exprn*)

Si *expr1* es nula retorna el valor de *expr2*. Si *expr1* no es nula la función retorna el valor de *expr1*.

- Sintaxis:

NVL(*expr1*, *expr2*)

- Ejemplo:

```
SELECT last_name, manager_id,  
       NVL(manager_id,0) JEFE  
FROM employees;
```

	LAST_NAME	MANAGER_ID	JEFE
1	King	(null)	0
2	Kochhar	100	100
3	De Haan	100	100
4	Hunold	102	102
5	Ernst	103	103
6	Austin	103	103
7	Pataballa	103	103
8	Lorentz	103	103
9	Greenberg	101	101
10	Faviet	108	108
.....			
104	Mavris	101	101
105	Baer	101	101
106	Higgins	101	101
107	Gietz	205	205

- Ejemplo:

```
SELECT last_name, salary, commission_pct "PORC. COMISION SIN NVL",
       (salary*12) + (salary*12*commission_pct) "SALARIO ANUAL SIN NVL" ,
       NVL(commission_pct, 0) "PORC. COMISION CON NVL",
       (salary*12) + (salary*12*NVL(commission_pct,0)) "SALARIO ANUAL CON NVL"
FROM employees;
```

	LAST_NAME	SALARY	PORC. COMISION SIN NVL	SALARIO ANUAL SIN NVL	PORC. COMISION CON NVL	SALARIO ANUAL CON NVL
1	King	24000	(null)	(null)	0	288000
2	Kochhar	17000	(null)	(null)	0	204000
3	De Haan	17000	(null)	(null)	0	204000
4	Hunold	9000	(null)	(null)	0	108000
.....						
54	Olsen	8000	0,2	115200	0,2	115200
55	Cambrault	7500	0,2	108000	0,2	108000
56	Tuvault	7000	0,15	96600	0,15	96600
57	King	10000	0,35	162000	0,35	162000
.....						
104	Mavris	6500	(null)	(null)	0	78000
105	Baer	10000	(null)	(null)	0	120000
106	Higgins	12008	(null)	(null)	0	144096
107	Gietz	8300	(null)	(null)	0	99600

Evalúa *expr1*. Si ésta no es Nula la función retorna *expr2*.
Si *expr1* es nula entonces la función retorna la *expr3*.

- Sintaxis: **NVL2(*expr1*, *expr2*, *expr3*)**

- Ejemplo:

```
SELECT employee_id, salary, commission_pct,  
       NVL2(commission_pct, 'SALARIO+COMISION', 'SOLO SALARIO')  
       AS "SALARIO MENSUAL CORRESPONDE A"  
FROM employees  
WHERE employee_id IN(100, 101, 114, 147,148, 149);
```

	EMPLOYEE_ID	SALARY	COMMISSION_PCT	SALARIO MENSUAL CORRESPONDE A
1	100	24000	(null)	SOLO SALARIO
2	101	17000	(null)	SOLO SALARIO
3	114	11000	(null)	SOLO SALARIO
4	147	12000	0,3	SALARIO+COMISION
5	148	11000	0,3	SALARIO+COMISION
6	149	10500	0,2	SALARIO+COMISION

Compara dos expresiones (*expr1* y *expr2*). Si son iguales, la función retorna Nulo

- Sintaxis:

NULLIF(*expr1*, *expr2*)

- Ejemplo:

```
SELECT first_name, salary, commission_pct,  
       salary*NVL(commission_pct,0) "VALOR COMISION",  
       NULLIF(salary*NVL(commission_pct,0),0) "RESULTADO FUNCION NULLIF"  
FROM employees  
WHERE employee_id IN(100, 104, 106, 110, 166, 167);
```

	FIRST_NAME	SALARY	COMMISSION_PCT	VALOR COMISION	RESULTADO FUNCION NULLIF
1	Steven	24000	(null)	0	(null)
2	Bruce	6000	(null)	0	(null)
3	Valli	4800	(null)	0	(null)
4	John	8200	(null)	0	(null)
5	Sundar	6400	0,1	640	640
6	Amit	6200	0,1	620	620

Retorna la primera expresión no nula de la lista

- Sintaxis: **COALESCE(expr1, expr2, ..., exprn)**

- Ejemplo:

```
SELECT last_name, commission_pct, manager_id,  
       COALESCE(commission_pct, manager_id, 9999) "RESULTADO FUNCION COALESCE"  
FROM employees  
WHERE department_id IN(10, 20, 90);
```

	LAST_NAME	COMMISSION_PCT	MANAGER_ID	RESULTADO FUNCION COALESCE
1	Whalen	(null)	101	101
2	Hartstein	(null)	100	100
3	Fay	(null)	201	201
4	King	(null)	(null)	9999
5	Kochhar	(null)	100	100
6	De Haan	(null)	100	100

Las funciones que operan sobre un fila se pueden anidar sin límites.
Se evalúan desde el nivel más interno hasta el nivel más externo.

F3(F2(F1(col,arg1), arg2), arg3)



- Ejemplo:

```
SELECT last_name, manager_id, LPAD(NVL(TO_CHAR(manager_id),'NO POSEE JEFE'),13,' ') mensaje  
FROM employees;
```

	LAST_NAME	MANAGER_ID	MENSAJE
1	King	(null)	NO POSEE JEFE
2	Kochhar	100	100
3	De Haan	100	100
4	Hunold	102	102
.....			
104	Mavris	101	101
105	Baer	101	101
106	Higgins	101	101
107	Gietz	205	205

- Ejemplo:

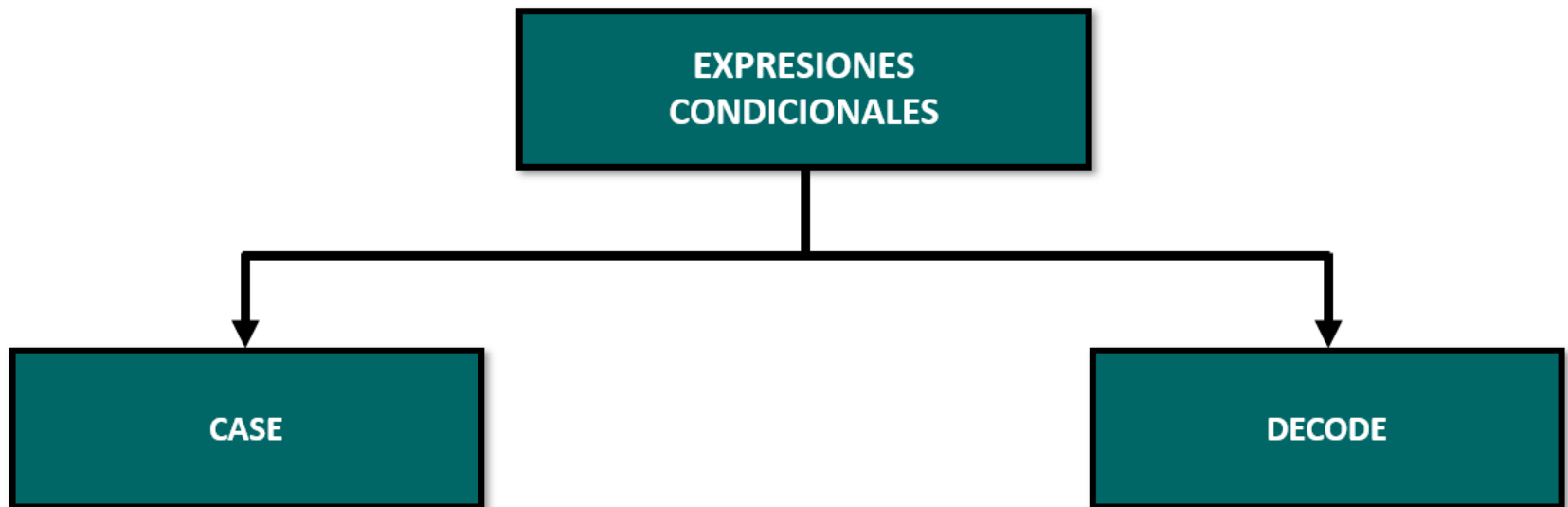
```
SELECT last_name, department_id,  
       UPPER(CONCAT(SUBSTR(last_name, 1, 6), '_chile')) resultado  
FROM employees  
WHERE department_id between 10 AND 40;
```

	LAST_NAME	DEPARTMENT_ID	RESULTADO
1	Whalen	10	WHALEN_CHILE
2	Hartstein	20	HARTST_CHILE
3	Fay	20	FAY_CHILE
4	Raphaely	30	RAPHAEL_CHILE
5	Khoo	30	KHOO_CHILE
6	Baida	30	BAIDA_CHILE
7	Tobias	30	TOBIAS_CHILE
8	Himuro	30	HIMURO_CHILE
9	Colmenares	30	COLMEN_CHILE
10	Mavris	40	MAVRIS_CHILE



Implementar Condición Lógica en Sentencias SQL

Permiten implementar una condición lógica IF-THEN-ELSE en una sentencia SQL



- Sintaxis 1:

```
CASE expr WHEN expr1_comparación THEN expr1_resultado  
      [ WHEN expr2_comparación THEN expr2_resultado  
        WHEN exprN_comparación THEN exprN_resultado  
        ELSE expr_else resultado ]  
END
```

- Sintaxis 2:

```
CASE WHEN expr1 expr1_comparación THEN expr1_resultado  
      [ WHEN expr2 expr2_comparación THEN expr2_resultado  
        WHEN exprN exprN_comparación THEN exprN_resultado  
        ELSE expr_else resultado ]  
END
```

- Ejemplo:

```
SELECT employee_id, job_id, department_id, salary,  
       CASE job_id WHEN 'PR_REP' THEN 1.15*salary  
                WHEN 'MK_MAN' THEN 1.20*salary  
                ELSE salary END "Salario Incrementado"  
FROM employees  
WHERE department_id IN(70, 20, 110);
```

	EMPLOYEE_ID	JOB_ID	DEPARTMENT_ID	SALARY	Salario Incrementado
1	201	MK_MAN	20	13000	15600
2	202	MK_REP	20	6000	6000
3	204	PR_REP	70	10000	11500
4	205	AC_MGR	110	12008	12008
5	206	AC_ACCOUNT	110	8300	8300

- Ejemplo:

```
SELECT last_name, salary,  
       (CASE WHEN salary <= 5000 THEN 'Bajo'  
             WHEN salary <= 10000 THEN 'Medio'  
             WHEN salary <= 20000 THEN 'Bueno'  
             ELSE 'Excelente' END) "Calificación del Salario"  
FROM employees  
WHERE department_id IN(60,90)  
ORDER BY salary DESC;
```

	LAST_NAME	SALARY	Calificación del Salario
1	King	24000	Excelente
2	De Haan	17000	Bueno
3	Kochhar	17000	Bueno
4	Hunold	9000	Medio
5	Ernst	6000	Medio
6	Austin	4800	Bajo
7	Pataballa	4800	Bajo
8	Lorentz	4200	Bajo

- Sintaxis:

```
DECODE(columna|expresión, búsqueda1, resultado1  
        [, búsqueda2, resultado2 ... , ]  
        [resultado por defecto])
```

- Ejemplo:

```
SELECT employee_id, job_id, department_id, salary,  
       DECODE (job_id, 'PR_REP', 1.15*salary,  
               'MK_MAN', 1.20*salary,  
               salary) "Salario Incrementado"  
FROM employees  
WHERE department_id IN(70, 20, 110);
```

	EMPLOYEE_ID	JOB_ID	DEPARTMENT_ID	SALARY	Salario Incrementado
1	201	MK_MAN	20	13000	15600
2	202	MK_REP	20	6000	6000
3	204	PR_REP	70	10000	11500
4	205	AC_MGR	110	12008	12008
5	206	AC_ACCOUNT	110	8300	8300

- Se explicó cómo convertir en forma explícita un tipo de dato numérico a carácter en una sentencia SQL.
- Se explicó cómo convertir en forma explícita un tipo de dato carácter a número en una sentencia SQL.
- Se explicó cómo convertir en forma explícita un tipo de dato fecha a carácter en una sentencia SQL.
- Se explicó cómo convertir en forma explícita un tipo de dato carácter a fecha en una sentencia SQL.
- Se explicó cómo definir un valor cuando alguna expresión es NULA en una sentencia SQL.
- Se explicó cómo trabajar con Funciones Anidadas en una sentencia SQL.
- Se explicó cómo implementar una condición lógica IF-THEN-ELSE en una sentencia SQL.