



Unidad 8

Aplicación de funciones en consultas de selección.




Bases de datos 2011-12




Introducción

- Las funciones se usan dentro de expresiones y actúan con los valores de las columnas, variables o constantes.




Bases de datos 2011-12




Introducción

- Generalmente generan dos tipos diferentes de resultados: unas producen un resultado que es una modificación de la información original (Ej. poner en minúscula una cadena que está en mayúscula), el resultado de otras indica alguna cosa sobre la información (Ej. El número de caracteres que tiene una cadena)




Bases de datos 2011-12




Introducción

- Se utilizan en: cláusulas SELECT, cláusulas WHERE, cláusulas ORDER BY.
- Es posible en anidamiento de funciones. Existen cinco tipos de funciones: aritméticas, de cadenas de caracteres, de manejo de fechas, de conversión, otras varias




Bases de datos 2011-12




1. Funciones aritméticas

- Trabajan con datos de tipo numérico NUMBER. Este tipo incluye los dígitos de 0 a 9, el punto decimal y el signo menos, si es necesario.
- Para probar algunas de estas funciones usaremos la tabla DUAL
- DUAL es una tabla pequeña, pero muy útil, creada para probar funciones o para hacer cálculos rápidos. Es una tabla de trabajo




Bases de datos 2011-12




Tipos de funciones aritméticas

- **Funciones de valores simples**
- **Funciones de grupos de valores**
- **Funciones de listas**




Bases de datos 2011-12




1.1 Funciones de valores simples

- Trabajan con valores simples (un número, una variable, una columna)



Bases de datos 2011-12




ABS(n)

- Devuelve el valor absoluto de "n". El valor absoluto es siempre un número positivo.

```
select abs(-125) from dual
```

Results	Explain	Describe	Saved SQL	Hi
ABS(-125)				
125				



Bases de datos 2011-12



CEIL(n)

- Obtiene el valor entero inmediatamente superior o igual a "n"
- Para números positivos, el valor superior de un número decimal es el siguiente entero y el valor superior de un número entero es él mismo.



Bases de datos 2011-12



Ejemplo

```
SQL> SELECT CEIL(20.7), CEIL(20.2), CEIL(16) FROM DUAL;
```

```
CEIL(20.7) CEIL(20.2) CEIL(16)
```

```
-----
```

```
21      21      16
```

Para números negativos el resultado es:

```
SQL> SELECT CEIL(-20.7), CEIL(-20.2), CEIL(-17) FROM DUAL;
```

```
CEIL(-20.7) CEIL(-20.2) CEIL(-17)
```

```
-----
```

```
-20     -20     -17
```



Bases de datos 2011-12



FLOOR(n)

- Es lo opuesto a CEIL: devuelve el valor entero inmediatamente inferior o igual a "n"
- Para números positivos, el valor inferior de un número decimal es el entero anterior, y el valor inferior de un número entero es él mismo:



Bases de datos 2011-12



Ejemplo

```
SQL> SELECT FLOOR(20.7), FLOOR(20.2), FLOOR(17) FROM DUAL;
```

```
FLOOR(20.7) FLOOR(20.2) FLOOR(17)
```

```
-----
```

```
20      20      17
```

Para números negativos

```
SQL> SELECT FLOOR(-20.7), FLOOR(-20.2), FLOOR(-17) FROM DUAL
```

```
FLOOR(-20.7) FLOOR(-20.2) FLOOR(-17)
```

```
-----
```

```
-21     -21     -17
```



Bases de datos 2011-12



MOD(m,n)

- Devuelve el resto resultante de dividir "m" entre "n". Tanto "m" como "n" pueden ser números reales.

SQL> SELECT MOD(11,4), MOD(11,0), MOD(-10,3), MOD(10.4,4.5) FROM DUAL

MOD(11,4)	MOD(11,0)	MOD(-10,3)	MOD(10.4,4.5)
3	11	-1	1,4



Bases de datos 2011-12



NVL(valor, expresión)

- Se utiliza para sustituir un valor nulo por otro valor. Si "valor" es NULL es sustituido por la "expresión"; si no lo es, la función devuelve "valor".
- NVL se puede usar con cualquier tipo de datos: numéricos, carácter, tipo fecha, pero "valor" y "expresión" deben ser del mismo tipo, aunque admiten tipos diferentes.



Bases de datos 2011-12



NVL(valor, expresión)

- Con esta función se evitan los valores nulos en expresiones aritméticas y funciones.
- Los valores nulos no intervienen en las funciones orientadas a grupos.



Bases de datos 2011-12




Ejemplo


- A partir de la tabla EMPLE obtener el SALARIO, la COMISION y la suma de ambos. Sin aplicar la función NVL. En este caso, cuando alguno de los dos sumandos sea nulo, la suma dará nulo
- SQL> SELECT salario, comision, salario+comision FROM emple;




Bases de datos 2011-12



SALARIO	COMISION	SALARIO+COMISION
104000	-	-
208000	39000	247000
162500	65000	227500
386750	-	-
162500	182000	344500
370500	-	-
318500	-	-
390000	-	-
650000	-	-
195000	0	195000
143000	-	-
123500	-	-
169000	-	-




Bases de datos 2011-12




Ejemplo


- Aplicar la función NVL al ejemplo anterior. Si es nula la comisión la sustituimos por el valor 0.
- SQL> SELECT salario, comision, salario+NVL(comision,0) FROM emple;




Bases de datos 2011-12



SALARIO	COMISION	SALARIO+NVL(COMISION,0)
104000	-	104000
208000	39000	247000
162500	65000	227500
386750	-	386750
162500	182000	344500
370500	-	370500
318500	-	318500
390000	-	390000
650000	-	650000
195000	0	195000
143000	-	143000
123500	-	123500
169000	-	169000



Bases de datos 2011-12




Ejemplo

Sustituir los valores nulos de la columna NOMBRE de la tabla NOMBRES por la cadena "Vacio".

SQL> SELECT NVL(nombre,'Vacio'), edad FROM nombres;

NVL(NOMBRE,'VAC	EDAD
PEDRO	17
JUAN	17
MARÍA	16
CLARA	14
Vacio	
Vacio	18
JESÚS	



Bases de datos 2011-12



POWER(m,exponente).

- Calcula la potencia de un número. Devuelve el valor de "m" elevado a un "exponente".
- SQL> SELECT POWER (3,4), POWER(3,-4), POWER(4.5,2.4), POWER(4.5,2) FROM DUAL
- POWER(3,4) POWER(3,-4) POWER(4.5,2.4) POWER(4.5,2)
- -----
- 81 ,01234568 36,958134 20,25



Bases de datos 2011-12



ROUND(número[,m]).

- Devuelve el valor de "número" redondeado a "m" decimales. Si "m" es negativo, el redondeo de dígitos se lleva a cabo a la izquierda del punto decimal. Si se omite "m", devuelve "número" con 0 decimales y redondeado.



Bases de datos 2011-12



Ejemplo

SQL> select round(1.5634,1),round(1.5634),round(1.2234,2),round(1.2676,3) from dual

ROUND(1.5634,1) ROUND(1.5634) ROUND(1.2234,2) ROUND(1.2676,3)

1,6 2 1,22 1,268

SQL> select round(145.5,-1),round(145.5,-2),round(145.5,-3),round(141,-1) from dual;

ROUND(145.5,-1) ROUND(145.5,-2) ROUND(145.5,-3) ROUND(141,-1)

150 100 0 140



Bases de datos 2011-12



SIGN(valor)

SIGN(valor). Indica el signo del "valor". Si "valor" es menor que 0, la función devuelve -1; y si "valor" es mayor que 0, la función devuelve 1

SQL> SELECT SIGN(-10), SIGN(10) FROM DUAL;

SIGN(-10) SIGN(10)

-1 1



Bases de datos 2011-12



SQRT(n).

Devuelve la raíz cuadrada de "n". El valor de "n" no puede ser negativo

```
SQL> SELECT SQRT(25),SQRT(25.6) FROM DUAL;
```

```
SQRT(25) SQRT(25.6)
```

```
-----
5      5,0596443|
```



Bases de datos 2011-12



TRUNC(número,[m])

- Devuelve "número" truncado a "m" decimales; "m" puede ser negativo; si lo es, trunca por la izquierda del punto decimal



Bases de datos 2011-12



Ejemplo

```
SQL> SELECT TRUNC(1.5634,1), TRUNC(1.1684,2),TRUNC(1.662) FROM DUAL;
```

```
TRUNC(1.5634,1) TRUNC(1.1684,2) TRUNC(1.662)
```

```
-----
1,5          1,16          1
```

```
SQL> SELECT TRUNC(187.98,-1), TRUNC(187.98,-2), TRUNC(187.98,-3) FROM DUAL;
```

```
TRUNC(187.98,-1) TRUNC(187.98,-2) TRUNC(187.98,-3)
```

```
-----
180             100             0
```



Bases de datos 2011-12



1.2 Funciones de grupos de valores

- Las funciones anteriores operaban con valores simples; no obstante hay funciones estadísticas que actúan sobre un grupo de filas para obtener un valor. Los valores nulos son ignorados por las funciones de grupos de valores, y los cálculos se realizan sin contar con ellos.



Bases de datos 2011-12

Funciones de grupos de valores	
Función	Propósito
AVG(n)	Calcula el valor medio de "n" ignorando los valores nulos
COUNT(* expresión)	Cuenta el número de veces que la expresión evalúa algún dato con valor no nulo. La opción "*" cuenta todas las filas seleccionadas
MAX(expresión)	Calcula el máximo valor de la expresión
MIN(expresión)	Calcula el mínimo valor de la expresión
SUM(expresión)	Obtiene la suma de valores de la expresión



Bases de datos 2011-12

Ejemplos

- Calcula el salario medio de los empleados del departamento 10 de la tabla EMPLE

```
SQL> SELECT AVG(salario) FROM emple
      WHERE dept_no=10;

      AVG(SALARIO)
      -----
      379166,67
```



Bases de datos 2011-12

Ejemplos

Calcula el número de filas de la tabla EMPLE

```
SQL> SELECT COUNT(*) FROM emple;

COUNT(*)
-----
14
```

Calcula el número de filas de la tabla EMPLE donde la COMISIÓN no es nula.

```
SQL> SELECT COUNT(comision) FROM emple;

COUNT(COMISION)
-----
4
```



Bases de datos 2011-12

Ejemplo


Obtener el apellido del empleado que tiene mayor salario

```
SQL> SELECT apellido, salario FROM emple
      2 WHERE salario=(SELECT MAX(salario) FROM emple);

APELLIDO  SALARIO
-----
REY       650000
```



Bases de datos 2011-12



Ejemplo

Calcula en número de oficios que hay en la tabla EMPLE, escribiendo el alias "nº oficios"


```
SQL> SELECT COUNT(oficio) "nº oficios" FROM emple;
```

```
nº oficios
-----
14
```


pero si quiero saber los distintos oficios que hay

```
SQL> SELECT COUNT(DISTINCT oficio) "nº oficios" FROM emple
```

```
nº oficios
-----
5
```




Bases de datos 2011-12




1.3 Funciones de listas

Las funciones de listas trabajan sobre un grupo de columnas dentro de una misma fila. Comparan los valores de cada una de las columnas en el interior de una fila para obtener el mayor o el menor valor de la lista

Función	Propósito
GREATEST (valor1,valor2,...)	Obtiene el mayor valor de la lista
LEAST (valor1, valor2, ...)	Obtiene el menor valor de la lista




Bases de datos 2011-12




Ejemplo

```
SQL> SELECT nombre_alumno, GREATEST(nota1,nota2,nota3) "Mayor",
LEAST(nota1,nota2,nota3) "Menor"
FROM notas_alumnos;
```

NOMBRE_ALUMNO	Mayor	Menor
Alcalde García, M. Luisa	5	5
Benito Martín, Luis	8	6
Casas Martínez, Manuel	7	5
Corregidor Sánchez, Ana	9	6
Díaz Sánchez, María		




Bases de datos 2011-12



2. Funciones de cadenas de caracteres

- Estas funciones trabajan con datos de tipo CHAR o VARCHAR2. Estos datos incluyen cualquier carácter alfanumérico: letras, número y caracteres especiales. Los literales se deben encerrar entre comillas simples.



Bases de datos 2011-12

2.1 Funciones que devuelven valores carácter

- Devuelven un carácter o un conjunto de caracteres

Función	Propósito
CHR(n)	Devuelve el carácter cuyo valor en binario es equivalente a "n"
CONCAT(cad1,cad2)	Devuelve "cad1" concatenada con "cad2"
LOWER(cad)	Devuelve la cadena "cad" con todas sus letras convertidas a minúsculas
UPPER(cad)	Devuelve la cadena "cad" con todas sus letras convertidas a mayúsculas
INITCAP(cad)	Convierte la cadena "cad" a tipo título
LPAD(cad1,n [,cad2])	Añade caracteres a la izquierda de la cadena, hasta que tiene una cierta longitud
RPAD(cad1,n [,cad2])	Añade caracteres a la derecha hasta conseguir una cierta longitud
LTRIM(cad [,set])	Suprime un conjunto de caracteres a la izquierda de la cadena
RTRIM(cad [,set])	Suprime un conjunto de caracteres a la derecha de la cadena
REPLACE(cad,cadena_búsqueda [,cadena_sustitución])	Sustituye un carácter o caracteres de una cadena con 0 o más caracteres
SUBSTR(cad,m [,n])	Obtiene parte de una cadena
TRANSLATE(cad1,cad2,cad3)	Convierte caracteres de una cadena en caracteres diferentes, según un plan de sustitución marcado por el usuario
SOUNDEX(cadena)	Devuelve una cadena de caracteres que contiene la representación fonética de "cadena"
TRIM ([leading trailing both] trim_carácter FROM trim_source)	Permite eliminar caracteres del principio, o del final o ambos de una cadena. Por defecto es both

Ejemplo

Obtener el apellido de los empleados de la tabla EMPLE de la siguiente manera:
El apellido es: APELLIDO

```
SQL> SELECT CONCAT('El apellido es: ', apellido)
2* FROM emple;
```

CONCAT('ELAPELIDOES:','APE

El apellido es: SÁNCHEZ
El apellido es: ARROYO
El apellido es: SALA
El apellido es: JIMÉNEZ
El apellido es: MARTÍN
El apellido es: NEGRO
El apellido es: CEREZO

El apellido es: GIL
El apellido es: REY
El apellido es: TOVAR
El apellido es: ALONSO
El apellido es: JIMENO
El apellido es: FERNÁNDEZ
El apellido es: MUÑOZ

14 filas seleccionadas.

Ejemplo

Obtener en una columna llamada CONCATENACIÓN el apellido y el oficio de cada uno de los empleados de la tabla EMPLE, de la siguiente manera: APELLIDO es OFICIO

```
SQL> SELECT CONCAT(CONCAT(apellido, ' es '), oficio) "Concatenación"
FROM emple;
```

Concatenación

SÁNCHEZ es EMPLEADO
ARROYO es VENDEDOR
SALA es VENDEDOR
JIMÉNEZ es DIRECTOR
MARTÍN es VENDEDOR
NEGRO es DIRECTOR
CEREZO es DIRECTOR

GIL es ANALISTA
REY es PRESIDENTE
TOVAR es VENDEDOR
ALONSO es EMPLEADO
JIMENO es EMPLEADO
FERNÁNDEZ es ANALISTA
MUÑOZ es EMPLEADO

14 filas seleccionadas.



Ejemplo

Para cada fila de la tabla NOTAS_ALUMNOS, obtener el nombre del alumno que tenga una longitud de 30 caracteres y rellenar por la izquierda con puntos.

```
SQL> SELECT LPAD(nombre_alumno, 30, '.') FROM notas_alumnos;
LPAD(NOMBRE_ALUMNO,30,'.')
```

```
.....Alcalde García, M. Luisa
.....Benito Martín, Luis
.....Casas Martínez, Manuel
.....Corregidor Sánchez, Ana
.....Díaz Sánchez, María
Ejemplo 66.
```



Bases de datos 2011-12



Ejemplo

A partir de la tabla MISTEXTOS, quitar los caracteres punto y comilla de la derecha de la columna TITULO y las comillas de la izquierda

```
SQL> SELECT LTRIM(RTRIM(titulo, '.')) FROM mistextos;
```

```
LTRIM(RTRIM(TITULO, '.'))
```

```
METODOLOGÍA DE LA PROGRAMACIÓN
INFORMÁTICA BÁSICA
SISTEMAS OPERATIVOS
SISTEMAS DIGITALES
MANUAL DE C
```



Bases de datos 2011-12



2.2 Funciones que devuelven valores numéricos

Función	Propósito
ASCII(cad)	Devuelve el valor ASCII de la primera letra de la cadena "cad"
INSTR(cad1, cad2 [,comienzo [m]])	Busca un conjunto de caracteres en una cadena. Devuelve la posición de la m-esima ocurrencia de cad2 en cad1, empezando la búsqueda por la posición <i>comienzo</i>
LENGTH(cad)	Devuelve el número de caracteres de cad



Bases de datos 2011-12



Ejemplo

Encontrar la posición de la primera ocurrencia de la letra 'A' en la columna AUTOR de la tabla MISTEXTOS

```
SQL> SELECT autor, INSTR(autor, 'A') FROM mistextos;
```

```
AUTOR          INSTR(AUTOR, 'A')
.....
ALCALDE, GARCÍA      1
GARCÍA GARCERÁN      2
J.F. GARCÍA          7
M.A. RUÍZ            3
M.A. RUÍZ            3
```



Bases de datos 2011-12

3. Funciones para el manejo de fechas

- Oracle puede almacenar datos de tipo fecha (DATE). Su formato por omisión es 'DD-MM-YY', pero con la función TO_CHAR es posible mostrar las fechas de cualquier modo. Los literales de fecha deben encerrarse entre comillas simples.

```
SQL> SELECT * FROM NLS_DATABASE_PARAMETERS;
```

PARAMETER	VALUE
NLS_CSMIG_SCHEMA_VERSION	5
NLS_LANGUAGE	SPANISH
NLS_TERRITORY	SPAIN
NLS_CURRENCY	?
NLS_ISO_CURRENCY	SPAIN
NLS_NUMERIC_CHARACTERS	,
NLS_CHARACTERSET	WE8ISO8859P15
NLS_CALENDAR	GREGORIAN
NLS_DATE_FORMAT	DD/MM/RR
NLS_DATE_LANGUAGE	SPANISH
NLS_SORT	SPANISH
NLS_TIME_FORMAT	HH24:MI:SSXFF
NLS_TIMESTAMP_FORMAT	DD/MM/RR HH24:MI:SSXFF
NLS_TIME_TZ_FORMAT	HH24:MI:SSXFF TZR
NLS_TIMESTAMP_TZ_FORMAT	DD/MM/RR HH24:MI:SSXFF TZR
NLS_DUAL_CURRENCY	?
NLS_COMP	BINARY
NLS_LENGTH_SEMANTICS	BYTE
NLS_NCHAR_CHARACTERSET	AL16UTF16
NLS_NCHAR_CONV_EXCP	FALSE
NLS_RDBMS_VERSION	10.2.0.4.0

Función	Propósito
SYSDATE	Devuelve la fecha del sistema
ADD_MONTHS(fecha, n)	Devuelve la fecha "fecha" incrementada en "n" meses
LAST_DAY(fecha)	Devuelve la fecha del último día del mes que contiene "fecha"
MONTHS_BETWEEN(fecha1, fecha2)	Devuelve la diferencia en meses entre las fechas "fecha1" y "fecha2". Puede ser un número decimal
NEXT_DAY(fecha, cad)	Devuelve la fecha del primer día de la semana indicado por "cad" después de la fecha indicada por "fecha". Cad es: lunes, martes...


Ejemplo

Dada la tabla EMPLE, sumar dos meses a la fecha de alta

```
SQL> SELECT fecha_alt, ADD_MONTHS(fecha_alt,2) FROM emple;
```

FECHA_AL	ADD_MONT		
17/12/80	17/02/81	09/11/81	09/01/82
20/02/80	20/04/80	17/11/81	17/01/82
22/02/81	22/04/81	08/09/81	08/11/81
02/04/81	02/06/81	23/09/81	23/11/81
29/09/81	29/11/81	03/12/81	03/02/82
01/05/81	01/07/81	03/12/81	03/02/82
09/06/81	09/08/81	23/01/82	23/03/82

14 filas seleccionadas.




Operaciones aritméticas con fechas

Fecha + número } = fecha Agrega o resta una cantidad de días a la fecha
 Fecha - número }


Fecha - fecha = número de días Resta una fecha de otra
 Fecha + número / 24 = fecha Agrega una cantidad de horas (número) a la fecha

El formato por defecto en el sistema puede ser: DD/MM/YY ó DD/MON/YY Si queremos cambiar utilizaremos:

```
ALTER SESSION SET NLS_DATE_FORMAT='DD/MM/YYYY'
```




Bases de datos 2011-12



Funciones de transformación

Transforman un tipo de dato en otro

Función	Propósito
TO_CHAR	Transforma un tipo de DATE o NUMBER en una cadena de caracteres
TO_DATE	Transforma un tipo NUMBER o CHAR en date
TO_NUMBER	Transforma una cadena de caracteres en NUMBER



Bases de datos 2011-12




TO_CHAR(fecha, 'formato')

- Convierte una fecha de tipo DATE a tipo VARCHAR2 en el formato especificado, según máscara indicada en la tabla de control de formato de fechas siguiente:




Bases de datos 2011-12




Máscaras de formato numéricas


cc o scc	Valor del siglo
y, yyy o sy,yyy	Año con coma, con o sin signo
yyyy	Año sin signo
yyy	Últimos tres dígitos del año
yy	Últimos dos dígitos del año
y	Último dígito del año
q	Número del trimestre
ww	Número de semana del año
w	Número de semana del mes
mm	Número de mes
ddd	Número de día del año
dd	Número de día del mes
d	Número de día de la semana
hh o hh2	Hora (1-12)
hh24	Hora (1-24)
mi	Minutos
ss	Segundos
ssss	Segundos transcurridos desde medianoche
j	Juliano




Bases de datos 2011-12



Máscaras de formato de caracteres	
syear o year	Año en inglés
month	Nombre del mes
mon	Abreviatura de tres letras del nombre del mes
day	Nombre del día de la semana
dy	Abreviatura de tres letras del nombre del día
a.m. o p.m.	Muestra a.m. o p.m. dependiendo del momento del día
b.c. o a.d.	Indicador para el año (antes de Cristo o después de Cristo)



Bases de datos 2011-12



Ejemplo


A partir de la tabla EMPLE, obtener la fecha de alta, de manera que aparezca el nombre del mes con todas sus letras, el número del día del mes, y el año

```
SQL>SELECT TO_CHAR (fecha_alt, 'Month dd, yyyy') "NUEVA FECHA"
FROM emple;
```


NUEVA FECHA

Diciembre	17, 1980	Noviembre	17, 1981
Febrero	20, 1980	Septiembre	08, 1981
Febrero	22, 1981	Septiembre	23, 1981
Abril	02, 1981	Diciembre	03, 1981
Septiembre	29, 1981	Diciembre	03, 1981
Mayo	01, 1981	Enero	23, 1982
Junio	09, 1981		
Noviembre	09, 1981		

14 filas seleccionadas.



Bases de datos 2011-12



Ejemplo


Obtener la fecha anterior, de forma que aparezca el nombre del mes con tres letras, el número de día del año, el último dígito del año y los tres últimos dígitos del año

```
SQL> SELECT TO_CHAR (fecha_alt, 'mon ddd y yy')
2 FROM emple;
```


TO_CHAR(FECHA

dic 352 0 980	nov 313 1 981
feb 051 0 980	nov 321 1 981
feb 053 1 981	sep 251 1 981
abr 092 1 981	sep 266 1 981
sep 272 1 981	dic 337 1 981
may 121 1 981	dic 337 1 981
jun 160 1 981	ene 023 2 982

14 filas seleccionadas.



Bases de datos 2011-12




Ejemplo

Obtener la fecha de hoy con el siguiente formato:
Hoy es nombre_día, día_mes de nombre_mes de año


```
SQL> SELECT TO_CHAR(SYSDATE, 'Hoy es "day, dd " de " month " de "
yyyy') "Fecha"
2 FROM DUAL;
```

Fecha

Hoy es domingo ,15 de octubre de 2000




Bases de datos 2011-12




TO_CHAR (número,'formato').


- Convierte un número de tipo NUMBER a tipo VARCHAR2 en el formato especificado. Los distintos formatos numéricos son los siguientes:




Bases de datos 2011-12



9	999	Devuelve el valor con el número especificado. Si es positivo, deja un espacio, si es negativo lo escribe, dejando a blancos los ceros no significativos, excepto si el valor es 0
0	9990	Muestra un 0 si el valor es 0
0	0999	Devuelve el valor dejando ceros al principio
\$	\$9999	Devuelve el valor con el signo \$ a la izquierda
B	B999	Muestra un espacio en blanco si el valor es 0. Es el formato por omisión
M	999M	Si el número es negativo, el signo menos sigue al número. Por omisión, el signo se pone a la izquierda
S	S999 999S	Escribe siempre el signo, en la posición indicada
D	99D99	Devuelve el carácter decimal en la posición especificada
G	9G999	Devuelve el carácter de los miles en la posición especificada
L	L999	Devuelve el símbolo de la moneda local en la posición indicada
.(COMA)	9,999	Devuelve la coma en la posición especificada
.(PUNTO)	99.99	Devuelve el punto decimal en la posición especificada
EEEE	9.9EEEE	Devuelve el valor usando la notación científica
RN	RN	Devuelve el valor en números romanos
FM	FM90.9	Devuelve el valor alineado a la izquierda




Bases de datos 2011-12



TO_NUMBER(cadena [,formato]).

- Convierte la cadena "cadena" a tipo NUMBER según el formato especificado. La cadena ha de contener números, el carácter decimal o el signo menos a la izquierda. No puede haber espacios entre los números, ni otros caracteres



Bases de datos 2011-12



TO_DATE (cad,'formato').

- Convierte "cad" de tipo VARCHAR2 o CHAR a un valor de tipo DATE. El formato de fecha elegido es "formato"
- Para cambiar el formato de fecha, colocando los 4 dígitos del año**
`ALTER SESSION SET NLS_DATE_FORMAT='DD/MM/YYYY';`
- Para cambiar el lenguaje**
`ALTER SESSION SET NLS_DATE_LANGUAGE=French;`



Bases de datos 2011-12



Ejemplo

Cambiar la cadena '01012001' a tipo DATE

```
SQL>SELECT TO_DATE('01012001') FROM dual;
```

```
TO_DATE(
```

```
-----
```

```
01/01/01
```

Si realizo: SQL>ALTER SESSION SET NLS_DATE_FORMAT='DD/MM/YYYY'

```
SQL>SELECT TO_DATE('01012001') FROM dual;
```

```
TO_DATE('0
```

```
-----
```

```
01/01/2001]
```



Bases de datos 2011-12



Función decode

Hace las veces de sentencias CASE o IF-THEN-ELSE, para facilitar consultas condicionales.

Formato:

```
DECODE (columna|expression, valor_buscado1, resultado1
                                             [valor_buscado2, resultado2,.....,]
                                             [, defecto])
```

La función DECODE descifra una expresión después de compararla con cada valor de búsqueda. Si la expresión es la misma que la búsqueda, se devuelve el resultado.

Si se omite el valor por defecto, se devolverá un valor nulo en el caso de que la búsqueda no coincida con ninguno de los valores resultantes.



Bases de datos 2011-12



Ejemplo

```
SELECT DECODE(COMISION, 0, 'CERO', NULL, 'ERES UN VAGO', COMISION) FROM EMPLE
```

```
DECODE(COMISION, 0, 'CERO', NULL, 'ERES UN VAGO
```

ERES UN VAGO	ERES UN VAGO
39000	ERES UN VAGO
65000	CERO
ERES UN VAGO	ERES UN VAGO
182000	ERES UN VAGO
ERES UN VAGO	ERES UN VAGO
ERES UN VAGO	ERES UN VAGO

Si no le pongo valor por defecto



Bases de datos 2011-12



Definición

Las subconsultas son aquellas sentencias SELECT que forman parte de una cláusula WHERE de una sentencia SELECT anterior.

Una subconsulta consistirá en incluir una declaración SELECT como parte de una cláusula WHERE. El formato de una subconsulta es similar a éste:

```
SELECT.....
FROM .....
WHERE columna operador_comparativo (SELECT .....
                                     FROM ....
                                     WHERE .....);
```



Bases de datos 2011-12



Definición

La subconsulta (el comando SELECT entre paréntesis) se ejecutará primero y, posteriormente, el valor extraído es "introducido" en la consulta principal.



Bases de datos 2011-12



Ejemplo

Apellido de los empleados con el mismo oficio que GIL

```
select * from emple
where oficio =(select oficio
               from emple
               where apellido ='GIL')
```



Bases de datos 2011-12



Condiciones de búsqueda en subconsultas



A. Test de comparación de subconsultas (> ,<, <>, >=, <=, =)

Consiste en usar los operadores de comparación para comparar el valor de una expresión con un valor único producido por una subconsulta



Bases de datos 2011-12



Ejemplo

Obtener los datos de los empleados con el mismo oficio que GIL

```
select * from emple
where oficio =(select oficio
               from emple
               where apellido ='GIL')
```



Bases de datos 2011-12



Notas

- Con este tipo de test, la subconsulta solo puede producir un único valor (una fila y una columna), así, el siguiente ejemplo daría error. También fallaría la subconsulta devolviese más de una fila.

```
select * from emple
where oficio =(select oficio, salario
               from emple
               where apellido ='GIL')
```



Bases de datos 2011-12



B. Test de pertenencia a un conjunto devuelto por una subconsulta (IN).

Comprueba si el valor de una expresión coincide con uno del conjunto de valores producido por una subconsulta

Ejemplo: obtener los apellidos de los empleados cuyo oficio sea alguno de los oficios que hay en el departamento 20:

```
Select apellido from emple where oficio IN
(select oficio from emple where Dept_id=20)
```



Bases de datos 2011-12



C) Test de existencia (EXISTS, NOT EXISTS).

Examina si una subconsulta produce alguna fila de resultados. El test es TRUE si devuelve filas, si no devuelve filas, es FALSE.

Ejemplo: Listar los departamentos que tengan empleados

```
Select dnombre, dept_no from depart where
EXISTS (select * from emple where
emple.Dept_no=depart.dept_no)
```



Bases de datos 2011-12



Nota

- Para comprender la lógica de esta consulta se puede asumir que cada fila devuelta por la consulta principal, provoca la ejecución de la subconsulta, así, si la consulta principal devuelve 30 registros, se entenderá que se ejecutan 30 subconsultas, una por cada departamento que retorne la consulta principal



Bases de datos 2011-12



D) Test de comparación cuantificada (ANY, ALL) se usan en conjunción con los operadores (>, <, <>, >=, <=, =)

- **ANY** compara el valor de una expresión con cada uno de un conjunto de valores producido por una subconsulta, si alguna de las comparaciones individuales da como resultado TRUE, ANY devuelve TRUE, si la subconsulta no devuelve nada devolverá FALSE.



Bases de datos 2011-12



Ejemplo

Obtener los datos de los empleados cuyo salario sea igual a algún salario de los empleados del departamento 30:

```
Select * from emple where
salario=ANY(select salario from emple
where Dep_no=30);
```



Bases de datos 2011-12



ALL

Compara el valor de una expresión con cada uno del conjunto de valores producido por una subconsulta, si todas las comparaciones individuales dan como resultado TRUE, ALL devuelve TRUE, en caso contrario devuelve FALSE.



Bases de datos 2011-12



Ejemplo

- Obtener los datos de los empleados cuyo salario sea menor a cualquier salario de los empleados del departamento 30:
- `Select * from emple where salario < ALL (select salario from emple where Dept_no=30);`



Bases de datos 2011-12



Subconsultas anidadas

- Se puede usar una subconsulta para filtrar los resultados de otra subconsulta. De esta manera, se anidad subconsultas.



Bases de datos 2011-12



Ejemplo: Nombre del departamento donde trabaja el empleado que más cobra

El empleado que más cobra

X->Select max(salary) from employees

Y->Select department_id from employees
where salary=X

Select department_name from departments
where department_id=Y



Bases de datos 2011-12



Ejemplo

```
SELECT department_name
FROM departments
WHERE department_id=
(SELECT department_id FROM employees
WHERE salary =
(SELECT max(salary) FROM employees))
```



Bases de datos 2011-12