

SELECT

Formato:

SELECT [ALL/DISTINCT] col1,col2..

FROM tabla WHERE cond1...

ORDER BY colOrd;

Asc -> Incremento, **Desc** ->

Descenso.

OPERADORES

Todas las operaciones sobre null, dan null.

OPERADORES COMPARACIÓN

Permiten comparar dos valores.

= igual que

!= , <>, ^= distinto que

<= menor o igual que

>= mayor o igual que

IN valor igual que uno de

NOT IN valor distinto de los de

BETWEEN valor dentro del intervalo

NOT BETWEEN valor fuera del

intervalo

LIKE ' _abc%' comparado con...

' _ ' un carácter cualquiera

'%' varios caracteres

IS NULL campo nulo

IS NOT NULL campo NO nulo

OPERADORES CONCATENACIÓN

Unir valores, campos, etc..

|| unir con, se pueden unir dos frases inventadas o valores ya existentes.

OPERADORES LÓGICOS

AND devuelve TRUE si ambas son correctas

OR devuelve TRUE si una es correcta

NOT invierte el resultado de la expresión.

FUNCIONES

Formato: **nombreFunción**

[(parámetro1, parámetro2, ...)]

FUNCIONES NUMÉRICAS

ABS(n) -> Obtiene el valor absoluto de un número n, o sea el valor en positivo

EXP(n) -> Obtiene el exponente en base e del número n (e^n)

CEIL(n) -> Calcula el valor entero haciendo un redondeo superior o igual al argumento n

FLOOR(n) -> Calcula el valor entero haciendo un redondeo inferior o igual al argumento n

MOD(m,n) -> Calcula el resto resultante de dividir m entre n

POWER(valor,exponente) -> Eleva un número(valor) a otro número indicado(exponente)

ROUND(n,decimales) -> Redondea número a los decimales indicados.

SQRT(n) -> Raíz cuadrada de n

TRUNC(m,n) -> Trunca un número a la cantidad de decimales n. Si no se indica n se truncan todos los decimales. Si n es negativo el truncado es en la parte entera quitando números.

SIGN(n) -> Si n es positivo devuelve 1 y si es negativo devuelve -1. Si es 0 devuelve 0.

FUNCIONES CADENAS CARACTERES

CHR(n) -> Devuelve el carácter ASCII cuyo valor numérico es n (Una letra)

ASCII(n) -> Lo mismo de antes pero devuelve un número

CONCAT(cadena1,cadena2) ->

Unión de ambas cadenas

LOWER(cad) -> Minúsculas

UPPER(cad) -> Mayúsculas

INITCAP(cad) -> La primera en mayúsculas, demás minúsculas

LPAD(cadena1, n, cadena2) ->

Rellenamos por la izquierda
SELECT LPAD('palabraqueponemos',
numerodeletrasquedebetener,'con
loquellorrellenamos');

RPAD(cadena1, n, cadena2) ->

Rellenamos por la derecha. SELECT
RPAD('palabraqueponemos',
numerodeletrasquedebetener,'con
loquellorrellenamos');

REPLACE(cadena, antiguo, nuevo)

Devuelve cadena donde se ha sustituido la aparición de ant por la cadena nue en todos los casos.

SUBSTR(cadena,m,n) ->

Coge un trozo de la cadena
SUBSTR(cadena,desdondeseempeza,
letrasquesecogen)

LENGTH(cadena) -> Longitud de la cadena

TRIM(cadena) -> Elimina los espacios en blanco en ambos lados.

LTRIM(cadena) -> Elimina los espacios en blanco (izquierda)

RTRIM(cadena) -> Elimina los espacios en blanco (derecha)

INSTR(cad,cadBus[,posInic[,nApar]]) Obtiene la posición de la cadena buscada dentro de otra cadena.

FUNCIONES FECHAS

SYSDATE -> Fecha y hora actual

SYSTIMESTAMP -> Fecha y hora formato ampliado

ADD_MONTHS(fecha,n) -> n son los meses que queremos sumar
MONTHS_BETWEEN(fecha1, fecha2) -> Número de meses entre estas fechas

LAST_DAY(fecha) -> Último día del mes

NEXT_DAY(fecha,d) -> Devuelve la fecha del día siguiente que sea d ('lunes','martes',...)

EXTRACT(val from fecha) -> Extrae el valor de una fecha (dd,mm...)

FUNCIONES CONVERSIÓN

TO_NUMBER(cadena

[,formato_máscara][,nsl_language]
) Cadena de texto en número

TO_CHAR(d,formato) Convierte un número o fecha a una cadena

TO_DATE(cadena,formato)
Convierte de texto a fecha

TABLA SÍMBOLOS TO_NUMBER

9 -> Posiciones numéricas, debe tener la misma cantidad de números que la cadena.

\$ -> Antes del número indicado con 0

L -> Moneda local, detrás del número 9999.99

S -> Antes del número, con el \$

TABLA SÍMBOLOS TO_CHAR

YY -> Año con dos cifras

YYYY -> Año con cuatro cifras

MM -> Mes con dos cifras

MON -> 3 primeras letras del mes

MONTH -> Nombre completo

DY -> Día con 3 letras

DAY -> Día completo

DD -> Día con dos cifras

D -> Día semana (1 al 7)

TABLA SÍMBOLOS TO_CHAR

Q -> Semestre

WW -> Semana del año

AM -> Indicador

PM -> Indicador

HH12 -> Hora de 1 a 12

HH24 -> Hora de 0 a 23

MI -> Minutos de 0 a 59

SS -> Segundos dentro del min

SSSS -> Segundos dentro de las 0 horas

FUNCIÓN NVL Y DECODE

Formato: **NVL(valor, expresión)**

Si valor es NULL, se devuelve la expresión marcada.

Formato: **DECODE (expresión1, condición1, valor1 [, condición 2, valor2, ...], default)**

Si se van cumpliendo las condiciones, se pone el valor1, si no se cumple se pone el valor default.

EMPAREJAR CAMPOS

En el filtrado (WHERE) se indicarán las condiciones del emparejamiento, siendo una de las columnas clave principal en su tabla.

Si se quiere **combinar** una **tabla consigo misma** hay que poner un **alias** a uno de los nombres de la tabla.

Si un campo tiene el **mismo nombre** en las **distintas tablas**, hay que **especificarlo** indicando el nombre de la tabla junto a un punto y el nombre del campo.

FUNCIONES DE AGREGADO

SUM. Suma de valores

COUNT. Cuenta los elementos que le indiquemos como columnas o *

MIN. Valor mínimo

MAX. Valor máximo

AVG. Media de los valores

AGRUPAMIENTO

Para obtener tantas filas como posiciones diferentes haya ->

GROUP BY

Las columnas que aparezcan en el **SELECT** deben aparecer en **GROUP BY**

La cláusula **HAVING** se utiliza para indicar la condición que deben cumplir los grupos para que se realice la consulta. Se coloca después del **GROUP BY** y antes de **ORDER BY**

JOIN

Mejora en la sintaxis de creación de composiciones de tablas.

Formato:

SELECT

tabla1.col1,...tabla1.coln,tabla2.col1,...tabla2.coln FROM tabla1;

CROSS JOIN

Se comporta igual que el primer **SELECT** visto de varias tablas.

SELECT col1, col2,... coln FROM tabla1 CROSS JOIN tabla2;

NATURAL JOIN

Los campos se deben llamar igual en ambas tablas sobre la columna de unión

SELECT * FROM tabla1 NATURAL JOIN tabla2;

Nombre campo fk tabla1 = nombre campo pk tabla2

JOIN ... USING

Permite establecer con qué campos se van a relacionar las tablas. Solo se tendrá en cuenta la relación que se indique

JOIN... ON

Une las tablas en las que los nombres no coinciden entre las tablas.

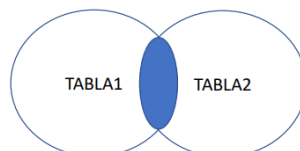
SELECT tabla1.col, tabla2.col, ... FROM tabla1 JOIN tabla2 ON tabla1.col = tabla2.col;

on = where

INNER JOIN

Devuelve los resultados que coinciden en ambas tablas.

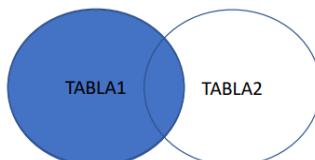
SELECT * FROM tabla1 INNER JOIN tabla2 ON campo relacionado entre tablas.



LEFT JOIN

Las filas de la tabla de la izquierda se devuelven, aunque no haya ninguna columna correspondiente en las tablas combinadas

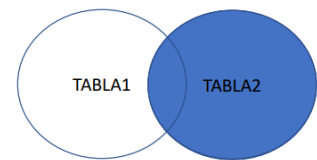
SELECT * FROM tabla1 LEFT JOIN tabla2 ON campo relacionado entre tablas



RIGHT JOIN

Las filas de la tabla de la derecha se devuelven, aunque no haya ninguna columna correspondiente en las tablas combinadas

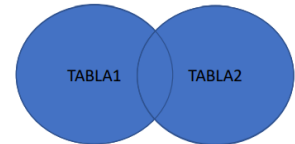
SELECT * FROM tabla1 RIGHT JOIN tabla2 ON campo relacionado entre tablas



FULL OUTER JOIN

Devolverán todas las tablas de ambas tablas y donde pondrá null también.

SELECT * FROM tabla1 FULL OUTER JOIN tabla2 ON campo relacionado entre ambas



UNION

Se suman las tablas sin mostrar lo duplicado

Formato: **SELECT... UNION SELECT...**

INTERSECT

Se muestra lo que tienen en común

Formato: **SELECT... INTERSECT**

SELECT...

MINUS

A la primera tabla le resto la segunda

Formato: **SELECT... MINUS SELECT...**