

## COMANDOS BÁSICOS

select \*  
from tabla

select columna as Apellidos  
select columna “Nombre y apellidos”  
select distinct columna  
and  
or

not

where

## OPERADORES DE COMPARACIÓN

between

not between

in

not in

any

all

like % muchos caracteres  
like \_ un único carácter

**select:** selecciona las columnas según su nombre, si se pone \* selecciona todas.

**from:** hace referencia a la tabla de la cual se extraen los datos. Renombra el encabezado de la columna a la palabra después de as, si necesitamos incluir espacios utilizamos las comillas. Hace que no se muestren los valores duplicados.

Devuelve verdadero si las expresiones son verdaderas.

Devuelve verdadera si cualquiera de las expresiones es verdadera.

Invierte la lógica de la expresión, de falso/verdadero y viceversa.

Se utiliza para incluir una condición.

Nos permite obtener datos que se encuentren en un rango.

Nos permite obtener datos que se encuentren fuera de un rango.

Permite obtener registros cuyos valores estén en una lista de valores.

Distinto a cualquiera de los miembros entre paréntesis.

Compara con cualquier registro de la subconsulta. La instrucción es válida si hay un registro en la subconsulta que permite que la comparación sea cierta.

Compara con todos los registros de la consulta. La instrucción resulta cierta si es cierta toda la comparación con los registros de la subconsulta.

Permite obtener registros cuyo valor en un campo cumpla una condición textual.

**FUNCIONES DE CADENA DE CARACTERES**

lower(texto)	Convierte el texto en minúsculas (caracteres españoles)
upper(texto)	Convierte el texto a mayúsculas.
initcap(texto)	Coloca la primera letra de cada palabra en mayúsculas.
rtrim(texto)	Elimina los espacios a la derecha del texto.
ltrim(texto)	Elimina los espacios a la izquierda del texto.
trim(texto)	Elimina los espacios en blanco del texto en ambos lados y los espacios dobles del interior.
trim(caracteres from texto)	Elimina del texto los caracteres indicados.
substr(texto,n,m)	Obtiene los m siguientes caracteres del texto a partir de la posición n 8si m no se indica se cogen desde n hasta el final).
length(texto)	Obtiene el tamaño del texto.
instr(texto,textobuscado,posinicial,naparición)	Obtiene la posición en la que se encuentra el texto buscado en el texto inicial.
replace(texto, textoabuscar,textoreemplazo)	Si no lo encuentra devuelve 0. Busca el texto a buscar en un determinado texto y lo cambia por el indicado como texto de reemplazo. Si no se indica texto de reemplazo, entonces está función elimina el texto a buscar.
translate(texto,caractresacambiar,caractressustiutivos)	Función que permite transformar caracteres.
lpad(texto,anchuramáxima,caracterderrelleno)	Rellena el texto a la izquierda o derehca con el carácter indicado para ocupar la anchura indicada.
rpadd(texto,anchuramáxima,caracterderrelleno)	Si el texto es más grande que la anchura hincada, el texto se recorta. Si no se indica carácter de relleno se rellenará el espacio marcado con espacios en blanco.
reverse(texto)	Invierte el texto.
ascii(carácter)	Devuelve el código ASCII del carácter indicado.
chr(número)	Devuelve el carácter correspondiente al código ASCII indicado.
select tipo, modelo, tipo    ‘-‘    modelo	Concatena.

## FUNCIONES DE FECHA Y CONVERSIÓN

sysdate	Obtiene la fecha y hora actuales.
systimestamp	Obtiene la fecha y hora actuales en formato timestamp.
add_months(fecha,n)	Añade a la fecha el número de meses indicado por n.
months_between(fecha1,fecha2)	Obtiene la diferencia en meses entre las dos fechas (puede ser decimal).
next_day(fecha,dia)	Indica cual es el día correspondiente a añadir a la fecha el día indicado. Dia = (1=lunes,2=martes...)
last_day(fecha)	Obtiene el último día del mes al que pertenece la fecha.
extract(valro from fecha)	Devuelve un valor DATE.
round(fecha,'formato')	Extrae un valor de una fecha concreta. El valor puede ser day, month, year, etc.
	Redondea la fecha al valor de aplicar el formato a la fecha.
	Formatos:
	‘year’ Hace que la fecha refleje el año completo.
	‘month’ Hace que la fecha refleje el mes completo más cercano a la fecha.
	‘hh24’ Redonde la hora a las 00:00 más cercanas
	‘day’ Redondea el día más cercano.
trunc(fecha formato)	Igual que el anterior pero trunca la fecha en lugar de redondearla.
greatest(fecha1,fecha2,...)	Devuelve la fecha más moderna de la lista.
least(fecha1,fecha2,...)	Devuelve la fecha más antigua de la lista.
to_char (fecha,'formato')	Obtiene un texto a partir de un número o una fecha.
to_number (cadena,'formato')	Convierte textos en números. Se indica el formato de la conversión.
to_date	Convierte textos en fechas. Como segundo parámetro se utilizan los códigos de formato de fechas comentados anteriormente.
YY	Año en formato de dos cifras
YYYY	Año en formato de cuatro cifras
MM	Mes en formato de dos cifras
MON	Las tres primeras letras del mes
MONTH	Nombre completo del mes
DY	Día de la semana en tres letras
DAY	Día completo de la semana
D	Día de la semana (del 1 al 7)
DD	Día en formato de dos cifras
DDD	Día del año
Q	Semestre
WW	Semana del año
AM	Indicador AM
PM	Indicador PM
HH12	Hora de 1 a 12
HH24	Hora de 0 a 23
MI	Minutos (0 a 59)
SS	Segundos (0 a 59)
SSSS	Segundos desde medianoche
/ . , : ; ’	Posición de los separadores, donde se pongan estos símbolos aparecerán en el resultado

## FUNCIONES DE CÁLCULO CON GRUPOS

count(\*)

Cuenta los elementos de un grupo. Se utiliza el asterisco para no tener que indicar un nombre de columna concreto, el resultado es el mismo para cualquier columna.

sum(expresión)

Suma los valores de la expresión

avg(expresión)

Calcula la media aritmética sobre la expresión indicada.

min(expresión)

Mínimo valor que toma la expresión indicada.

max(expresión)

Máximo valor que toma la expresión indicada.

## OTRAS FUNCIONES

minus

Resta una tabla con otra.

intersect

Compara dos tablas y quita aquellas filas que no tengan la relación marcada.

is null

Selecciona a la gente que no tiene teléfono.

order by titulo asc

Selecciona a la gente que tiene teléfono.

order by titulo desc

Ordena de forma ascendente.

round(n,decimales)

Ordena de forma descendente.

trunc(n,decimales)

Redondea el número al siguiente número con el número de decimales indicado más cercano.

mod(n1,n2)

Los decimales del número se cortan para que sólo aparezca el número de decimales indicado.

power(valor,exponente)

Devuelve el resto resultado de dividir n1 entre n2.

sqrt(n)

Eleve el valor al exponente indicado

sign(n)

Calcula la raíz cuadrada de n.

abs(n)

Devuelve 1 si n es positivo, cero si es cero y -1 si es negativo.

exp(n)

Calcula el valor absoluto de n.

ln(n)

Calcula en, es decir el exponente en base e del número n.

log(n)

Logaritmo neperiano de n.

sin(n)

Logaritmo en base 10 de n.

cos(n)

Calcula el seno de n (n tiene que estar en radianes)

tan(n)

Calcula el cose de n (n tiene que estar en radianes)

acos(n)

Calcula la tangente de n (n tiene que estar en radianes)

asin(n)

Devuelve en radianes el arco coseno de n.

atan(n)

Devuelve en radianes el arco seno de n.

sinh(n)

Devuelve en radianes el arco tangente de n.

cosh(n)

Devuelve el seno hiperbólico de n.

tanh(n)

Devuelve el cose hiperbólico de n.

soundex(texto)

Devuelve la tangente hiperbólica de n.

nv1(valor,sustituto)

Devuelve el valor fonético del texto. Es una función para buscar textos de los que no se sabe con exactitud su escritura.

nv12(valor,sustituto1,sustituto2)

Si el valor es NULL, devuelve el valor sustituto, de otro modo, devuelve el valor.

coalesce(listaexpresiones)

Devuelve el valor sustituto1 si valor no es nulo, si es nulo devuelve sustituto2.

nullif(valor1,valor2)

Devuelve la primera de las expresiones que no es nula.

decode (expresión,valor1,resultado1)

Devuelve nulo si valor 1 es igual a valor2, si no devuelve valor1.

Permite realizar condiciones en una consulta. Se evalúa una expresión y se colocan a continuación un valor, resultado de forma que si la expresión equivale al valor, se obtiene el resultado indicado.

## JOIN

### INNER JOIN

```
SELECT *  
FROM tabla1  
INNER JOIN tabla2  
ON tabla1.cod = tabla2.cod  
INNER JOIN tabla3  
ON tabla2.cod = tabla3.cod  
WHERE ...
```

```
SELECT *  
FROM tabla1, tabla2  
WHERE tabla1.cod = tabla2.cod
```

### OUTER JOIN

3 tpos de OUTER JOIN

- LEFT OUTER JOIN

```
SELECT *  
FROM empl  
LEFT OUTER JOIN dept  
ON empl.cod_dpt = dept.cod
```

- RIGHT OUTER JOIN

```
SELECT *  
FROM empl  
RIGHT OUTER JOIN dept  
ON empl.cod_dpt = dept.cod
```

- FULL OUTER JOIN

```
SELECT *  
FROM emp.  
FULL OUTER JOIN dept  
ON empl.cod_dpt = dept.cod
```

Une dos o más tablas por un campo.

Junta tablas que tengan un campo en común. Tiene que haber coincidencias entre las dos tablas

No necesariamente tiene que haber correspondencia entre el valor de la tabla izquierda y tabla derecha.

Retorna todos los valores de la tabla izquierda tengan o no correspondencia.

Retorna todos los valores de la tabla derecha tengan o no correspondencia.

Retorna todos los valores de la tabla derecha e izquierda tengan o no correspondencia.

## MANIPULACIÓN DE DATOS

INSERT INTO

INSERT INTO table (cols,...)  
VALUES (valores,...);  
OR  
INSERT INTO tabla  
VALUES (valores);  
INSERT INTO tabla (col1, col2..., valor1,...)  
SELECT col1, col2..., valor1,...  
FROM ....  
WHERE condiciones;  
UPDATE tabla  
SET col1 = valo1  
WHERE condición;

UPDATE

UPDATE CON  
SELECT

UPDATE nom\_tabla  
SET columna1=valor1, columna2=valor2,...  
WHERE columna3 = (SELECT ...);  
UPDATE nom\_tabla  
SET (col1, col2....) = (SELECT col1, col2....)  
WHERE condición;  
UPDATE nom\_tabla  
SET columna1=(SELECT col1 ...), columna2=(SELECT col2 ...)  
WHERE condición;

DELETE

DELETE [FROM] tabla [WHERE condición];

Inserta, actualiza y elimina datos  
Se puede especificar los campos en los cuales vamos a introducir datos o directamente introducimos los datos y dejamos null los que no sabemos.

## DDL CREATES Y CONSTRAINTS

TIPOS DE DATOS

Texto de anchura fija	char(n)
Texto de anchura variable	varchar2(n)
Texto de anchura fija para caracteres nacionales	nchar(n)
Texto de anchura variable para caracteres nacionales	nvarchar(n)
Enteros pequeños (2 bytes)	smallint
Enteros normales (4 bytes)	int
Enteros largos (8 bytes)	bigint
Enteros precisión decimal	number(2)
Decimal de coma variable	number
Decimal de cómo fija	number(m,d)
Fechas	date
Fecha y hora	timestamp
Intervalos	interval
Lógicos	bool
Binarios	bit / bit varying(n) / varbit(n)
Texto gran longitud	clob
Binario de gran longitud	blob

CREATE TABLE nom\_tabla (nom\_colum tipo\_dato (tamaño))

DROP TABLE nom\_tabla cascade constraints;

## MODIFICACIÓN DE TABLAS

ALTER TABLE nom\_viejo RENAME TO nom\_nuevo;

RENAME nom\_viejo TO nom\_nuevo;

TRUNCATE TABLE nom\_tabla

ALTER TABLE nom\_tabla ADD(nom\_columna tipo\_dato,...);

ALTER TABLE nom\_tabla DROP(nom\_columna, ...);

ALTER TABLE nom\_tabla ALTER COLUMN, tipo\_dato,...);

ALTER TABLE nom\_columna RENAME COLUMN nom\_viejo TO nom\_nuevo;

ALTER TABLE tabla ADD [CONSTRAINT nombre]

tipoDeRestricción(columnas);

CREATE TABLE nom\_tabla (nom\_colum tipo\_dato DEFAULT valor,...);

Primero borra las constraints de la tabla y luego las tablas

Cambia de nombre a una tabla

Borra contenido de tablas

Añade una columna o varias a la tabla

Borra columnas

Permite cambiar el tipo de datos y propiedades de una determinada columna.

Cambia el nombre de la columna

Añade constraints.

*tipoRestricción* es el texto CHECK, PRIMARY KEY o FOREIGN KEY. Las restricciones NOT NULL deben indicarse mediante ALTER TABLE .. MODIFY colocando NOT NULL en el campo que se modifica.

Le asigna un valor por defecto durante su creación

## RESTRICCIONES

```
CREATE TABLE nom_tabla (nom_columna tipo_dato CONSTRAINT
nom_constraint NOT NULL);
CREATE TABLE nom_tabla (nom_columna tipo_dato, nom_columna2
tipo_dato2, CONSTRAINT nom_constraint UNIQUE(columna, columna2));
CREATE TABLE nom_tabla ( nom_columna tipo_dato PRIMARY KEY);
CREATE TABLE nom_tabla ( nom_columna tipo_dato CONSTRAINT
nom_constraint PRIMARY KEY);
CREATE TABLE nom_tabla ( nom_columna tipo_dato, nom_columna1
tipo_dato1, CONSTRAINT nom_constraint PRIMARY KEY(columna,columna2));
ALTER TABLE tabla DISABLE CONSTRAINT nombre [CASCADE]
ALTER TABLE tabla ENABLE CONSTRAINT nombre [CASCADE]
CREATE TABLE tabla (
col1 tipodato CONSTRAINT nombre_constr REFERENCES tabla(col));
CREATE TABLE tabla(
Col1 tipodato,
...
CONSTRAINT nombre_constr FOREIGN KEY(col1,...)
REFERENCES tabla(clave)
[ON DELETE | ON UPDATE
[SET NULL | CASCADE | DEFAULT]]);
```

Prohíbe que el valor de esa columna sea nulo y le asigna un nombre a la constraint.

Obliga a que el contenido de las columnas no puedan ser valores repetidos

Declara la clave primaria de la tabla

Declara la clave primaria y coloca una constraint, esta constraint impide que el campo sea nulo

Declara la clave primaria de la tabla y le coloca una constraint para cuando hayan dos claves primaria

Desactivar constraints

Activar constraints

Declara las claves foráneas a nivel de fila

Declara las claves foráneas a nivel de tabla