

Gestión de bases de datos

Diseño físico de bases
de datos

SELECT

Es la sentencia para realizar consultas a una base de datos.

Formato:

```
SELECT [ALL | DISTINCT] col1, col2, ...  
FROM tabla1, table2, ... WHERE cond1,  
cond2, ... ORDER BY colOrd;
```

SELECT

¿Qué significa cada opción?

- ALL: es opcional e indica que se quieren obtener todas las filas (aunque estén repetidas). Es el valor por defecto por lo que no se pone.
- DISTINCT: es opcional e indica que se quieren obtener todas las filas pero que no ESTÉN REPETIDAS (las que se repitan se obviarán).

Ejercicio 1

- Obtén el nombre de todos los futbolistas de la base de datos.
- Ahora queremos saber la posición que ocupan estos futbolistas (aunque se repitan los valores).
- Se quiere saber las posiciones existentes por lo que se requiere la posición de los futbolistas sin que se repitan los valores.

Solución Ejercicio 1

- SELECT NOMBRE FROM FUTBOLISTAS;

	 NOMBRE	
1	PEDRO LUIS	
2	LUIS	
3	JESÚS	
4	DIEGO	
5	PABLO	
6	ESTEBAN	
7	ENRIQUE	

Solución Ejercicio 1

- SELECT POSICION FROM FUTBOLISTAS;

	⚡ POSICION	
1	DEFENSA	
2	DELANTERO	
3	MEDIOCENTRO	
4	PORTERO	
5	DEFENSA	
6	DEFENSA	
7	MEDIOCENTRO	

Solución Ejercicio 1

- SELECT DISTINCT POSICION FROM FUTBOLISTAS;

	 POSICION
1	DEFENSA
2	PORTERO
3	MEDIOCENTRO
4	DELANTERO

SELECT

- Col1, col2, ... son las columnas que se quieren seleccionar. Además:
 - Se separan por comas.
 - Se puede indicar el nombre de la tabla a la que referencian (ej.: futbolistas.nombre).
- Para añadir todos los campos se usa * (Ej.: SELECT * FROM...)
- Se pueden poner ALIAS a los nombres de las columnas.

SELECT

- ¿Qué son los alias de los nombres de la columnas?

Cuando se seleccionan columnas, se muestran los resultados bajo una titulares que por defecto son los nombres de las columnas. Si se quiere ver otro texto (alias) se indica a continuación entre comillas (Ej.:
SELECT nombre “Nombre futbolista”
FROM FUTBOLISTAS;

Ejercicio 2

- Obtén todas las posiciones posibles (diferentes) de los futbolistas bajo el encabezado “Demarcaciones”.
- Muestra todos los datos almacenados de todos los futbolistas.

Solución Ejercicio 2

- SELECT DISTINCT POSICION
“Demarcación” FROM FUTBOLISTAS;

	 Demarcación
1	DEFENSA
2	PORTERO
3	MEDIOCENTRO
4	DELANTERO

Solución Ejercicio 2

- SELECT * FROM FUTBOLISTAS;

	ID	NOMBRE	APELLIDOS	FECHA_NACIMIENTO	POSICION	SALARIO	ID_EQUIPO
1	F00220	PEDRO LUIS	GÓMEZ	20/04/85	DEFENSA	110000	1
2	F00320	LUIS	GÁLVEZ	12/01/90	DELANTERO	80000	1
3	F00520	JESÚS	FERNÁNDEZ	02/01/95	MEDIOCENTRO	140000	2
4	F00620	DIEGO	DÍAZ	05/04/87	PORTERO	160000	2
5	F00720	PABLO	CARO	27/01/88	DEFENSA	200000	3
6	F00820	ESTEBAN	BENÍTEZ	29/09/00	DEFENSA	300000	3
7	F00920	ENRIQUE	ÁLVAREZ	30/09/01	MEDIOCENTRO	120000	1

SELECT

- FROM tabla1, tabla2...:
 - Se definen los nombres de las tablas de las que proceden las columnas seleccionadas.
 - Si hay más de una tabla se denomina JOIN (consulta combinada) y requiere de una condición de combinación (WHERE).

SELECT

- FROM tabla1, tabla2...:
 - Se puede indicar el usuario propietario de la tabla (Ej.: usuario.tabla).
 - Para asociar un alias (como en el caso de las columnas) no se necesitan comillas (Ej.: SELECT * FROM FUTBOLISTAS FUT;).

SELECT

- WHERE cond1, cond2...:
 - Sirven para restringir las filas a seleccionar.
 - Se pueden emplear operadores, funciones y expresiones en las condiciones.

Ejercicio 3

- Devuelve los apellidos de los futbolistas que su posición sea DEFENSA. Aplica un alias a apellidos para que se muestre “Futbolista” y asigna el alias F a la tabla futbolistas.

Solución Ejercicio 3

- SELECT F.APELLIDOS “Futbolista”
FROM FUTBOLISTAS F WHERE
POSICION LIKE ‘DEFENSA’;

	Futbolista
1	GÓMEZ
2	CARO
3	BENÍTEZ

SELECT

- ORDER BY:
 - Se emplea para indicar el criterio de ordenación de los resultados.
 - Se realiza por cada columna (col1, col2, ...) y puede ser orden ascendente (por defecto si no se pone) o descendente (ASC | DESC).
 - Ej.: ... ORDER BY NOMBRE ASC;
 - El orden de las columnas indicadas especifica la prioridad.

SELECT

- ORDER BY:
 - También se puede indicar con números el campo a utilizar en el orden (Ej.: ... ORDER BY 2; siendo 2 col2).
 - Solo se puede ordenar por campos de tipo CARÁCTER, FECHA y NÚMERO.

Ejercicio 4

- Devuelve los apellidos de los futbolistas que su posición sea DEFENSA ordenador de la Z a la A.

Solución Ejercicio 4

- SELECT APELLIDOS FROM FUTBOLISTAS WHERE POSICION LIKE 'DEFENSA' ORDER BY APELLIDOS DESC;
- SELECT APELLIDOS FROM FUTBOLISTAS WHERE POSICION LIKE 'DEFENSA' ORDER BY 1 DESC;

APELLIDOS
1 GÓMEZ
2 CARO
3 BENÍTEZ

Ejercicio 5

- Devuelve todos los datos de futbolistas ordenados primero por posición y luego por apellidos.

Ejercicio 5

- SELECT * FROM FUTBOLISTAS
ORDER BY POSICION, APELLIDOS;

	ID	NOMBRE	APELLIDOS	FECHA_NACIMIENTO	POSICION	SALARIO	ID_EQUIPO
1	F00820	ESTEBAN	BENÍTEZ	29/09/00	DEFENSA	300000	3
2	F00720	PABLO	CARO	27/01/88	DEFENSA	200000	3
3	F00220	PEDRO LUIS	GÓMEZ	20/04/85	DEFENSA	110000	1
4	F00320	LUIS	GÁLVEZ	12/01/90	DELANTERO	80000	1
5	F00920	ENRIQUE	ÁLVAREZ	30/09/01	MEDIOCENTRO	120000	1
6	F00520	JESÚS	FERNÁNDEZ	02/01/95	MEDIOCENTRO	140000	2
7	F00620	DIEGO	DÍAZ	05/04/87	PORTERO	160000	2

Operadores

- Permiten realizar operaciones matemáticas, unir cadenas o comparar valores.
- Pueden ser:
 - Relacionales/comparación.
 - Aritméticos (sumas, restas, etc.).
 - Concatenación/unión cadenas.
 - Lógicos.

Operadores comparación

- Permiten comparar dos valores.
- Estos valores pueden ser de campos, variables, etc.
- Si la comparación tiene lugar, es decir, es correcta, se da como verdadera. En caso contrario, si no se cumple. se dará como falsa.

Operadores comparación

- Operadores y operación:
 - = igual que
 - != distinto que
 - < > distinto que
 - ^= distinto que
 - < menor que
 - <= menor o igual que
 - > mayor que
 - >= mayor o igual que

Operadores comparación

- Operadores y operación:
 - IN valor igual que uno de
 - NOT IN valor distinto de los de
 - BETWEEN valor dentro de
 - NOT BETWEEN valor fuera de
 - LIKE ‘_abc%’ comparado con...
 - El símbolo ‘_’ se usa para un carácter cualquiera.
 - El símbolo ‘%’ se usa para varios caracteres cualesquiera.

Operadores comparación

- Operadores y operación:
 - IS NULL TRUE si el campo es nulo.
 - IS NOT NULL TRUE si el campo NO es nulo.

NULL:

Se emplea para dar valor inexistente y en un orden ASC aparece antes que los valores distintos de NULL.

Ejercicio 6

- Obtén el nombre de los futbolistas que cobran más de 150.000 euros (SALARIO).
- Muestra el ID de los futbolistas cuyo apellido sea GÓMEZ.
- Indica los apellidos de los futbolistas que empiezan por G.
- Devuelve todos los datos de los futbolistas cuya posición sea DEFENSA o DELANTERO.

Ejercicio 6

- Obtén el nombre de los futbolistas que cobran entre 100.000 y 200.000 euros (SALARIO).

Solución Ejercicio 6

- SELECT NOMBRE FROM
FUTBOLISTAS WHERE SALARIO >
150000;

	NOMBRE
1	DIEGO
2	PABLO
3	ESTEBAN

Solución Ejercicio 6

- SELECT ID FROM FUTBOLISTAS
WHERE APELLIDOS = 'GÓMEZ';

	ID	
1	F00220	

Solución Ejercicio 6

- SELECT APELLIDOS FROM
FUTBOLISTAS WHERE APELLIDOS
LIKE 'G%' ORDER BY APELLIDOS;

	APELLIDOS
1	GÁLVEZ
2	GÓMEZ

Solución Ejercicio 6

- `SELECT * FROM FUTBOLISTAS
WHERE POSICION IN ('DEFENSA',
'DELANTERO');`

	ID	NOMBRE	APELLIDOS	FECHA_NACIMIENTO	POSICION	SALARIO	ID_EQUIPO
1	F00220	PEDRO LUIS	GÓMEZ	20/04/85	DEFENSA	110000	1
2	F00320	LUIS	GÁLVEZ	12/01/90	DELANTERO	80000	1
3	F00720	PABLO	CARO	27/01/88	DEFENSA	200000	3
4	F00820	ESTEBAN	BENÍTEZ	29/09/00	DEFENSA	300000	3

Solución Ejercicio 6

- SELECT NOMBRE FROM
FUTBOLISTAS WHERE SALARIO
BETWEEN 100000 AND 200000;

	NOMBRE
1	PEDRO LUIS
2	JESÚS
3	DIEGO
4	PABLO
5	ENRIQUE

Operadores aritméticos

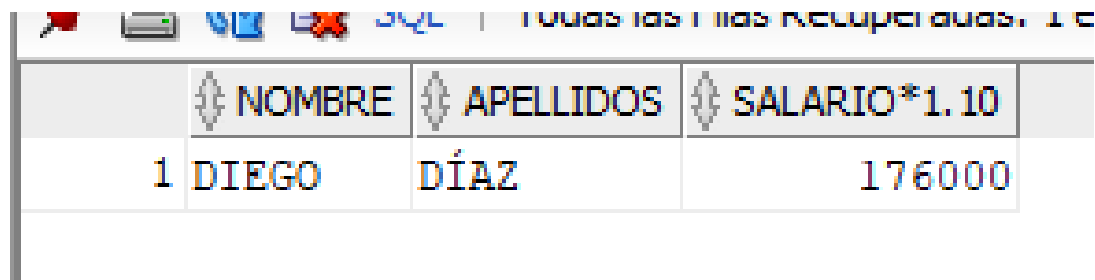
- Operaciones matemáticas:
 - + suma
 - - resta
 - * multiplicación
 - / división
- La salida puede ser por tanto el resultado de un cálculo y no un valor de un campo.
- Estas operaciones sobre NULL dan siempre como resultado NULL.

Ejercicio 7

- Se va a hacer un aumento de sueldo de los PORTEROS. Para ello obtén el nombre, apellido y el nuevo salario de los PORTEROS si se aumenta un 10% el que tienen actualmente. La lista debe estar ordenada alfabéticamente por los apellidos.

Solución Ejercicio 7

- SELECT NOMBRE, APELLIDOS, SALARIO*1.10 FROM FUTBOLISTAS WHERE POSICION LIKE 'PORTERO' ORDER BY APELLIDOS;



	NOMBRE	APELLIDOS	SALARIO*1.10
1	DIEGO	DÍAZ	176000

Operadores concatenación

- Unir valores, campos, etc:
 - || unir con
- Oracle puede convertir números en cadenas para hacer concatenaciones.

Ejercicio 8

- Devuelve en una única columna el nombre del futbolista seguido de la palabra “es” y a continuación la posición que ocupa. La columna debe llamarse “Posiciones”.

Solución Ejercicio 8

- SELECT NOMBRE || ' es ' || POSICION
“Posiciones” FROM FUTBOLISTAS;

	⚡ Posiciones
1	PEDRO LUIS es DEFENSA
2	LUIS es DELANTERO
3	JESÚS es MEDIOCENTRO
4	DIEGO es PORTERO
5	PABLO es DEFENSA
6	ESTEBAN es DEFENSA
7	ENRIQUE es MEDIOCENTRO

Operadores lógicos

- Se emplean para evaluar más de una condición, si cumple todas, alguna de ellas o ninguna.
- Operadores lógicos:
 - AND devuelve TRUE si ambas son correctas
 - OR devuelve TRUE si una es correcta
 - NOT invierte el resultado de la expresión.

Ejercicio 9

- Queremos saber todos los datos de los futbolistas que ganen más de 150.000 euros y sean defensas.

Solución Ejercicio 9

- SELECT * FROM FUTBOLISTAS
WHERE SALARIO > 150000 AND
POSICION LIKE 'DEFENSA';

	ID	NOMBRE	APELLIDOS	FECHA_NACIMIENTO	POSICION	SALARIO	ID_EQUIPO
1	F00720	PABLO	CARO	27/01/88	DEFENSA	200000	3
2	F00820	ESTEBAN	BENÍTEZ	29/09/00	DEFENSA	300000	3

Orden evaluación SELECT

- Las expresiones utilizadas en un SELECT tienen un orden de evaluación si no se emplean paréntesis.

Orden evaluación SELECT

1. * y después / al mismo nivel
2. + y -
3. ||
4. Comparaciones (<, >, >=, ...)
5. IS NULL, IS NOT NULL, LIKE, BETWEEN
6. NOT
7. AND
8. OR

Ejercicio 10

En la siguiente consulta:

```
SELECT APELLIDOS  
FROM FUTBOLISTAS  
WHERE APELLIDOS LIKE 'A%S%';
```

¿QUÉ SE ESTARÍA SELECCIONANDO?

- a) Aquellos futbolistas cuyos apellidos contienen la letra A y la S.
- b) Aquellos futbolistas cuyos apellidos comienzan por la A y contienen la S.
- c) Aquellos futbolistas cuyos apellidos no contienen ni la A ni la S.
- d) Todos los apellidos de todos los futbolistas menos los que su apellido comienza por S.

Solución Ejercicio 10

¿QUÉ SE ESTARÍA SELECCIONANDO?

- a) Aquellos futbolistas cuyos apellidos contienen la letra A y la S.
- b) Aquellos futbolistas cuyos apellidos comienzan por la A y contienen la S.**
- c) Aquellos futbolistas cuyos apellidos no contienen ni la A ni la S.
- d) Todos los apellidos de todos los futbolistas menos los que su apellido comienza por S.

Consultas calculadas

Operaciones aritméticas sobre uno o varios campos para obtener cierta información.

Ejemplos: sumas de dos campos, valor extra de un campo para saber un precio de un descuento, etc.

Consultas calculadas

Sintaxis: SELECT EXPRESIÓN
FROM *tabla*

Estas consultas no alteran los valores originales de las tablas consultadas.

Ejercicio 11

Realiza una consulta que te devuelva los campos Nombre, Salario, Salario bruto terminado en la palabra “euros” y llamando a esa columna “Salario bruto”.

Nota: el salario bruto de un futbolista se obtiene añadiéndole el 50% de impuestos, es decir multiplicando por 1,5.

Solución Ejercicio 11

```
SELECT NOMBRE, SALARIO, SALARIO *  
1.5 || ' EUROS' "SALARIO BRUTO" FROM  
FUTBOLISTAS;
```

	NOMBRE	SALARIO	SALARIO BRUTO
1	PEDRO LUIS	110000	165000 EUROS
2	LUIS	80000	120000 EUROS
3	JESUS	140000	210000 EUROS
4	DIEGO	160000	240000 EUROS
5	PABLO	200000	300000 EUROS
6	ESTEBAN	300000	450000 EUROS
7	ENRIQUE	120000	180000 EUROS

Funciones

Son operaciones que se realizan sobre los datos obteniendo un resultado.

Para hacer esto necesitan unos datos de entrada denominados **parámetros** o **argumentos**.

Se realizan los cálculos con estos datos de entrada que suelen ir entre paréntesis.

Funciones

Las funciones se pueden incluir en SELECT, WHERE y ORDER BY.

Sintaxis: nombreFunción [(parámetro1, parámetro2, ...)]

Se pueden crear funciones dentro de otras funciones. A esto se le denomina **anidar funciones dentro de funciones**.

Funciones numéricas

Se refiere a cálculos con valores numéricos.

❑ ABS(n)

Obtiene el valor absoluto de un número n.

❑ EXP(n)

Obtiene el exponente en base e del número n (e^n).

Funciones numéricas

❑ CEIL(n)

Calcula el valor entero inmediatamente superior o igual al argumento n (redondeo superior).

❑ FLOOR(n)

Calcula el valor entero inmediatamente inferior o igual al parámetro n (redondeo inferior).

Funciones numéricas

❑ MOD(m, n)

Calcula el resto resultante de dividir m entre n.

❑ POWER(valor, exponente)

Eleva el valor al exponente indicado.

Funciones numéricas

❑ ROUND(n, decimales)

Redondea el número n al siguiente número con tantos decimales como se indica en decimales.

❑ SQRT(n)

Calcula la raíz cuadrada de n.

Funciones numéricas

❑ 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.

❑ SIGN(n)

Si n es positivo devuelve 1 y si es negativo devuelve -1. Si es 0 devuelve 0.

Ejercicio 12

Realiza las siguientes operaciones utilizando funciones numéricas:

1. Calcula el valor absoluto de -10.
2. Obtén el exponente en base e de 4.
3. Redondea el número 15,3 a 16.
4. Redondea el número anterior (15,3) a 15.
5. Calcula el resto de 15 entre 3 ($15/3$).
6. Eleva 15 al exponente 2 (15^2).

Solución Ejercicio 12

1. Calcula el valor absoluto de -10.

```
SELECT ABS(-10) FROM DUAL;
```

	ABS(-10)
1	10

2. Obtén el exponente en base e de 4.

```
SELECT EXP(4) FROM DUAL;
```

	EXP(4)
1	54,59815003314423907811026120286087840308

Solución Ejercicio 12

3. Redondea el número 15,3 a 16.

```
SELECT CEIL(15.3) FROM DUAL;
```

CEIL(15.3)	
1	16

4. Redondea el número anterior (15,3) a 15.

```
SELECT FLOOR(15.3) FROM DUAL;
```

FLOOR(15.3)	
1	15

Solución Ejercicio 12

5. Calcula el resto de 15 entre 3 ($15/3$).

```
SELECT MOD(15,3) FROM DUAL;
```

MOD(15,3)	
1	0

6. Eleva 15 al exponente 2 (15^2).

```
SELECT POWER(15,2) FROM DUAL;
```

POWER(15,2)	
1	225

Ejercicio 13

Realiza las siguientes operaciones utilizando funciones numéricas:

1. Redondea 15,789 con un decimal.
2. Obtén la raíz cuadrada de 128.
3. Trunca 15,789 a 1 decimal.
4. Trunca 15,789 para dejarlo sin decimales (15).
5. Trunca 157,89 para dejarlo en 100.
6. Obtén el signo de -15 (es decir -1).

Solución Ejercicio 13

1. SELECT ROUND(15.789, 1) FROM DUAL;

ROUND(15.789,1)	
1	15,8

2. SELECT SQRT(128) FROM DUAL;

SQRT(128)	
1	11,31370849898476039041350979367758462855

3. SELECT TRUNC(15.789, 1) FROM DUAL;

TRUNC(15.789,1)	
1	15,7

Solución Ejercicio 13

4. SELECT TRUNC(15.789) FROM DUAL;

TRUNC(15.789)	
1	15

5. SELECT TRUNC(157.89, -2) FROM DUAL;

TRUNC(157.89,-2)	
1	100

6. SELECT SIGN(-15) FROM DUAL;

SIGN(-15)	
1	-1

Funciones de cadenas de caracteres

❑ CHR(n)

Devuelve el carácter ASCII cuyo valor numérico es n.

❑ ASCII(n)

La inversa de CHR, devuelve el valor numérico ASCII de un carácter n.

Funciones de cadenas de caracteres

❑ CONCAT(*cadena1*, *cadena2*)

Devuelve la unión de las cadenas *cadena1* y *cadena2* (igual que ||).

❑ LOWER(*cad*)

Devuelve la cadena *cad* con todos sus caracteres en minúsculas.

❑ UPPER(*cad*)

Devuelve la cadena *cad* con todos sus caracteres en mayúsculas.

Funciones de cadenas de caracteres

❑ INITCAP(cad)

Devuelve la cadena cad con su primer carácter en mayúsculas.

❑ LPAD(cadena1, n, cadena2)

Devuelve la cadena cadena1 con una longitud de n caracteres, ajustada a la derecha, rellenándose la parte izquierda con la cadena2.

Funciones de cadenas de caracteres

❑ RPAD(cadena1, n, cadena2)

Devuelve la cadena cadena1 con una longitud de n caracteres, ajustada a la izquierda, rellenándose la parte derecha con la cadena2.

Funciones de cadenas de caracteres

❑ REPLACE(cadena, ant, nue)

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

❑ SUBSTR(cadena, m, n)

Devuelve cadena compuesta por n caracteres a partir de la posición m.

Funciones de cadenas de caracteres

❑ LENGTH(cadena)

Devuelve el tamaño o longitud de la cadena.

❑ TRIM(cadena)

Elimina los espacios en blanco a la izquierda y a la derecha de cadena, así como los espacios dobles de entre medio de cadena.

Funciones de cadenas de caracteres

❑ LTRIM(cadena)

Elimina los espacios a la izquierda que tenga cadena.

❑ RTRIM(cadena)

Elimina los espacios en blanco que tenga cadena a la derecha.

Funciones de cadenas de caracteres

- ❑ INSTR(cad, cadBus [,posIni [,nApar]])
Obtiene la posición en la que se encuentra la cadena buscada cadBus dentro de la cadena cad. Se puede iniciar la búsqueda en la posición posIni (opcional) de cad, o indicar el número de apariciones que se quiera localizar de cadBus en cad con el parámetro opcional nApar. Si no se encuentra nada devuelve cero.

Funciones para tratar fechas

En Oracle existen dos tipos de datos para fechas:

- ❑ **DATE**. Guarda fecha e incluso hora.
- ❑ **TIMESTAMP**. Guarda instante (puede incluir hasta partes de un segundo).

Se pueden realizar operaciones numéricas con las fechas, como sumar o restar fechas (añadiendo o quitando días a una fecha).

Funciones para tratar fechas

Funciones:

- ❑ SYSDATE

Devuelve la fecha y hora actual.

- ❑ SYSTIMESTAMP

Devuelve la fecha y hora actual con formato ampliado.

- ❑ ADD_MONTHS(*fecha*, *n*)

Se le suma al parámetro *fecha*, *n* meses.

- ❑ MONTHS_BETWEEN(*fecha1*, *fecha2*)

Devuelve el número de meses existentes entre la *fecha1* y la *fecha2*.

Funciones para tratar fechas

Funciones:

❑ LAST_DAY(*fecha*)

Devuelve el último día del mes al que pertenece la *fecha*. Tendrá formato DATE.

❑ NEXT_DAY(*fecha*, *d*)

Devuelve el siguiente día que sea *d* (LUNES, MARTES, etc.) a partir de la *fecha*.

❑ EXTRACT(*val* FROM *fecha*)

Extrae un valor de una *fecha*. Este *val* puede ser DAY, MONTH, YEAR, HOURS, etc.

Ejercicio 14

Realiza las siguientes peticiones:

1. ¿Cuál es el carácter 80 en ASCII?
2. Obtén el valor ASCII de la letra H.
3. Devuelve el nombre y los apellidos en una única columna de los futbolistas cuya posición sea DEFENSA. El nombre de estar completamente en mayúsculas y los apellidos en minúsculas.

Ejercicio 14

Realiza las siguientes peticiones:

4. Obtén todas las posiciones posibles (no repetidas) con la primera letra en mayúsculas y el resto en minúscula) ordenadas alfabéticamente.
5. Devuelve los nombres de los futbolistas del EQUIPO A con el siguiente formato: --AAA, siendo AAA las tres primeras letras del nombre. Ej.: --PED, --LUI, etc.

Ejercicio 14

Realiza las siguientes peticiones:

6. Devuelve los nombres de los futbolistas del EQUIPO A con el siguiente formato: AAA--, siendo AAA las tres primeras letras del nombre. Ej.: PED--, LUI--, etc.
7. Devuelve el nombre de los futbolistas sustituyendo “LUIS” por “L.”.

Ejercicio 14

Realiza las siguientes peticiones:

8. Queremos los apellidos de los futbolistas con tan solo 3 caracteres.
9. Devuelve el nombre de los futbolistas sustituyendo “LUIS” por “L.” y en otra columna el tamaño de los apellidos (longitud de caract.).

Solución Ejercicio 14

1. SELECT CHR(80) FROM DUAL;

CHR(80)	
1	P

2. SELECT ASCII('H') FROM DUAL;

ASCII('H')	
1	72

3. SELECT CONCAT(UPPER(NOMBRE), ' ' || LOWER(APELLIDOS)) FROM FUTBOLISTAS WHERE POSICION LIKE 'DEFENSA';

CONCAT(UPPER(NOMBRE),' ' LOWER(APELLIDOS))	
1	PEDRO LUIS gomez
2	PABLO caro
3	ESTEBAN benitez

Solución Ejercicio 14

4. SELECT DISTINCT INITCAP(POSICION) FROM
FUTBOLISTAS ORDER BY 1;

	INITCAP(POSICION)
1	Defensa
2	Delantero
3	Mediocentro
4	Portero

5. SELECT LPAD(SUBSTR(NOMBRE, 1, 3), 5, '--')
FROM FUTBOLISTAS;

	LPAD(SUBSTR(NOMBRE,0,3),5,'--')
1	--PED
2	--LUI
3	--JES
4	--DIE
5	--PAB
6	--EST
7	--ENR

Solución Ejercicio 14

6. SELECT RPAD(SUBSTR(NOMBRE, 1, 3), 5, '--')
FROM FUTBOLISTAS;

	RPAD(SUBSTR(NOMBRE,0,3),5,'--')
1	PED--
2	LUI--
3	JES--
4	DIE--
5	PAB--
6	EST--
7	ENR--

7. SELECT REPLACE(NOMBRE, 'LUIS', 'L.') FROM
FUTBOLISTAS;

	REPLACE(NOMBRE,'LUIS','L.')
1	PEDRO L.
2	L.
3	JESUS
4	DIEGO
5	PABLO
6	ESTEBAN
7	ENRIQUE

Solución Ejercicio 14

8. SELECT SUBSTR(APELLIDOS, 1, 3) FROM FUTBOLISTAS;

	⚡ SUBSTR(APELLIDOS,0,3)
1	GOM
2	GAL
3	FER
4	DIA
5	CAR
6	BEN
7	ALV

9. SELECT REPLACE(NOMBRE, 'LUIS', 'L.'),
LENGTH(APELLIDOS) FROM FUTBOLISTAS;

	⚡ REPLACE(NOMBRE,'LUIS','L.')	⚡ LENGTH(APELLIDOS)
1	PEDRO L.	5
2	L.	6
3	JESUS	9
4	DIEGO	4
5	PABLO	4
6	ESTEBAN	7
7	ENRIQUE	7

Ejercicio 15

1. Obtén en tres columnas independientes, el día, el mes y la fecha de hoy.
2. Muestra por pantalla la fecha que será dentro de 6 meses utilizando **ADD_MONTHS**.
3. Muestra el número de meses que hay entre hoy y la fecha de nacimiento de todos los futbolistas.
4. ¿Cuál es el último día de este mes? Muéstralo.
5. ¿Qué día será el próximo lunes? Ponlo por pantalla mediante una petición.
6. Muestra el día que era hace 4 días. ¿Y el día de mañana?

Solución ejercicio 15

1. SELECT EXTRACT(DAY FROM SYSDATE), EXTRACT(MONTH FROM SYSDATE), EXTRACT(YEAR FROM SYSDATE) FROM DUAL;

	EXTRACT(DAYFROMSYSDATE)	EXTRACT(MONTHFROMSYSDATE)	EXTRACT(YEARFROMSYSDATE)
1	2	2	2021

2. SELECT ADD_MONTHS(SYSDATE, 6) FROM DUAL;

	ADD_MONTHS(SYSDATE,6)
1	02/08/21

3. SELECT FLOOR(MONTHS_BETWEEN(SYSDATE, FECHA_NACIMIENTO)) FROM FUTBOLISTAS;

	FLOOR(MONTHS_BETWEEN(SYSDATE,FECHA_NACIMIENTO))
1	429
2	372
3	313
4	405
5	396
6	244
7	232

Solución ejercicio 15

4. SELECT LAST_DAY(SYSDATE) FROM DUAL;

	LAST_DAY(SYSDATE)
1	28/02/21

5. SELECT NEXT_DAY(SYSDATE, 'LUNES') FROM DUAL;

	NEXT_DAY(SYSDATE,'LUNES')
1	08/02/21

6. SELECT SYSDATE - 4 "HACE 4D", SYSDATE + 1 "MAÑANA"
FROM DUAL;

	HACE 4D	MAÑANA
1	29/01/21	03/02/21

Funciones de conversión

La conversión de datos es el paso de un tipo de dato a otro. En Oracle, se hace de forma automática de texto a número y a la inversa.

Igualmente se realiza de forma automática la conversión de texto a fecha, y viceversa, para que los resultados tengan sentido.

Si se quieren realizar estos tipos de conversiones de forma expresa, existen algunas funciones de conversión.

Funciones de conversión

❑ `TO_NUMBER(cadena [, formato_máscara][, nsl_language])`.

Convierte una cadena de texto en número, dándole el formato que se pasa por parámetro. La cadena debe contener un número con el formato pasado en *formato_máscara*.

Los atributos “formato_máscara” y “nsl_language” son opcionales.

- Formato_máscara: es el formato en que se quiere convertir *cadena*. Es opcional.
- Nsl_language: el idioma utilizado al convertir *cadena* a un número. Es opcional.

Devuelve un valor numérico (número)

Funciones de conversión

❑ TO_CHAR(d, formato).

Convierte un número o fecha d a una cadena de caracteres. Es habitual utilizarlo para convertir fechas en cadenas de caracteres. Se le pasa el formato por atributo.

❑ TO_DATE(cadena, formato).

Convierte texto a fecha con el formato indicado.

Funciones de conversión

❑ Formatos para números (TO_NUMBER para cadenas)

Símbolo	Significado
9	Posiciones numéricas. Debe coincidir la cantidad de números tanto en la cadena como en el formato_máscara.
\$	Antepone el dólar al número que debe aparecer como tal en la cadena y en esa misma posición.
L	Coloca en la posición donde se ponga, la moneda local configurada con el parámetro NSL_CURRENCY que debe aparecer en la cadena en esa posición.
S	Muestra el símbolo del signo que debe aparecer como tal en la cadena y en esa misma posición.
.	Posición del símbolo decimal.

Funciones de conversión

☐ Formatos para fechas (TO_CHAR con fechas)

Símbolo	Significado
YY	Año con dos cifras.
YYYY	Año con cuatro cifras.
MM	Mes con dos cifras.
MON	Las tres primeras letras del mes.
MONTH	Nombre completo del mes.
DY	Día de la semana con tres letras.
DAY	Día completo de la semana.
DD	Día con dos cifras.
D	Día de la semana (1 al 7).

Funciones de conversión

☐ Formatos para fechas (TO_CHAR con fechas)

Símbolo	Significado
Q	Semestre.
WW	Semana del año.
AM	Indicador de AM.
PM	Indicador de PM.
HH12	Hora de 1 a 12.
HH24	Hora de 0 a 23.
MI	Minutos de 0 a 59.
SS	Segundos dentro del minuto.
SSSS	Segundos dentro desde las 0 horas.

Ejercicio 16

1. Convierte '1000.45' a un número.
2. Convierte '1000.45€' a un número.
3. Convierte '-\$1000.45' a un número.
4. Muestra el día de hoy con los siguientes formatos:
 - 12/02/2021, 12/02/21, 12-feb-2021, 12 de febrero de 2021
5. Convierte el ID de la tabla FUTBOLISTAS en un número de tres cifras.
6. Muestra todos los años de nacimiento de los futbolistas (solo el año, ej.: 1990, 1991, etc.).
Ordénalos de menor a mayor año.

Solución ejercicio 16

1. SELECT TO_NUMBER('1000.45', '9999.99') FROM DUAL;

TO_NUMBER('1000.45','9999.99')	
1	1000,45

2. SELECT TO_NUMBER('1000.45€', '9999.99L') FROM DUAL;


TO_NUMBER('1000.45€','9999.99L')	
1	1000,45

3. SELECT TO_NUMBER('- \$1,000.45', 'S\$9999.99') FROM DUAL;

TO_NUMBER('- \$1,000.45','S\$9999.99')	
1	-1000,45

Solución ejercicio 16

```
4.  SELECT TO_CHAR(SYSDATE, 'DD/MM/YYYY') FROM DUAL;  
     SELECT TO_CHAR(SYSDATE, 'DD/MM/YY') FROM DUAL;  
     SELECT TO_CHAR(SYSDATE, 'DD-mon-YYYY') FROM DUAL;  
     SELECT TO_CHAR(SYSDATE, 'DD') || ' de ' ||  
           REPLACE(TO_CHAR(SYSDATE, 'month'), ' ', ',') || ' de ' ||  
           TO_CHAR(SYSDATE, 'YYYY') FROM DUAL;
```

 TO_CHAR(SYSDATE,'DD') 'DE' REPLACE(TO_CHAR(SYSDATE,'MONTH'),',','') 'DE' TO_CHAR(SYSDATE,'YYYY')
1 12 de febrero de 2021

Solución ejercicio 16

5. `SELECT TO_NUMBER(SUBSTR(ID,2,LENGTH(ID)-1),'99999')`
`FROM FUTBOLISTAS;`

	<code>TO_NUMBER(SUBSTR(ID,2,LENGTH(ID)-1),'99999')</code>
1	220
2	320
3	520
4	620
5	720
6	820
7	920

6. `SELECT TO_CHAR(FECHA_NACIMIENTO,'YYYY')` FROM
`FUTBOLISTAS ORDER BY 1;`

	<code>TO_CHAR(FECHA_NACIMIENTO,'YYYY')</code>
1	1985
2	1987
3	1988
4	1990
5	1995
6	2000
7	2001

Función NVL y DECODE

Sintaxis: NVL(valor, expresión)

Se evalúa *valor*, y si es NULL, entonces se devuelve *expresión*. Los parámetros valor y expresión tiene que ser del mismo tipo.

Puede ser útil si tenemos campos NULL y queremos mostrarlos con otro valor, como “ns/nc”, “sin valor”, “0”, etc.

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

La función evalúa la primera expresión (expresión1). Si se cumple la condición1, se devuelve el valor1; en caso contrario se evalúa la condición2 y si se cumple, se devuelve el valor2, y así sucesivamente hasta que se cumpla alguna condición. Si no se cumple ninguna, se devuelve default.

Ejercicio 17

1. Utiliza la función DECODE para mostrar, respecto de los EQUIPOS, el valor 1000k si el presupuesto es de 1000000, 2000k si el presupuesto es de 2000000, o 3000k si el presupuesto asciende a 3000000.

Solución Ejercicio 17

1. `SELECT DECODE(PRESUPUESTO, 1000000, '1000K', 2000000, '2000K', '3000K') FROM EQUIPOS;`

	DECODE(PRESUPUESTO,1000000,'1000K',2000000,'2000K','3000K')
1	1000K
2	2000K
3	3000K



CEU

*Centro de Estudios
Profesionales*

Fundación San Pablo Andalucía