

ÍNDICE

1. CONSULTAS DE LOS DATOS	3
ACTIVIDAD. CREAR TABLAS EMPLE Y DEPART	3
1.1. COMANDO SELECT	4
▪ SELECCIÓN DE COLUMNAS	4
ACTIVIDAD. SELECCIÓN DE COLUMNAS	5
▪ CLÁUSULA ORDER BY	5
ACTIVIDAD. ORDENACIÓN	6
▪ CLÁUSULA FROM	6
▪ SELECCIÓN DE FILAS: CLÁUSULA WHERE	7
▪ DIFERENCIA ENTRE SELECCIÓN DE FILAS Y SELECCIÓN DE COLUMNAS	7
▪ ALIAS DE COLUMNA	8
ACTIVIDAD. SELECT SIMPLE (TABLA EMPLE)	8
ACTIVIDAD. SELECT SIMPLE (TABLA ALUM22)	8
2. OPERADORES	9
2.1. OPERADORES ARITMÉTICOS	9
ACTIVIDAD. OPERADORES ARITMÉTICOS	9
2.2. OPERADORES DE COMPARACIÓN Y LÓGICOS	9
ACTIVIDAD. OPERADORES DE COMPARACIÓN Y LÓGICOS	10
2.3. OPERADORES DE COMPARACIÓN DE CADENAS DE CARACTERES	11
ACTIVIDAD. OPERADORES DE COMPARACIÓN DE CADENAS DE CARACTERES	11
ACTIVIDAD. DIFERENCIA ENTRE CHAR Y VARCHAR2	11
2.4. IS NULL / IS NOT NULL	12
2.5. OPERADORES DE COMPARACIÓN DE CONJUNTOS DE VALORES	12
ACTIVIDAD. OPERADORES DE COMPARACIÓN DE CONJUNTOS DE VALORES	12
2.6. RESUMEN DE OPERADORES	13
3. FUNCIONES	14
3.1. ¿QUÉ ES UNA FUNCIÓN?	14
3.2. FUNCIONES ARITMÉTICAS	14
▪ ROUND	14
ACTIVIDAD. FUNCIÓN ROUND	14
▪ NVL	15
ACTIVIDAD. FUNCIÓN NVL	15
3.3. FUNCIONES DE AGREGADO	15
▪ AVG	16
ACTIVIDAD. FUNCIÓN AVG	16
▪ MAX	17
ACTIVIDAD. FUNCIÓN MAX	17

▪ MIN	17
ACTIVIDAD. FUNCIÓN MIN	17
▪ SUM	18
ACTIVIDAD. FUNCIÓN SUM	18
▪ COUNT	18
ACTIVIDAD. FUNCIÓN COUNT(*)	18
ACTIVIDAD. FUNCIÓN COUNT(CAMPO)	19

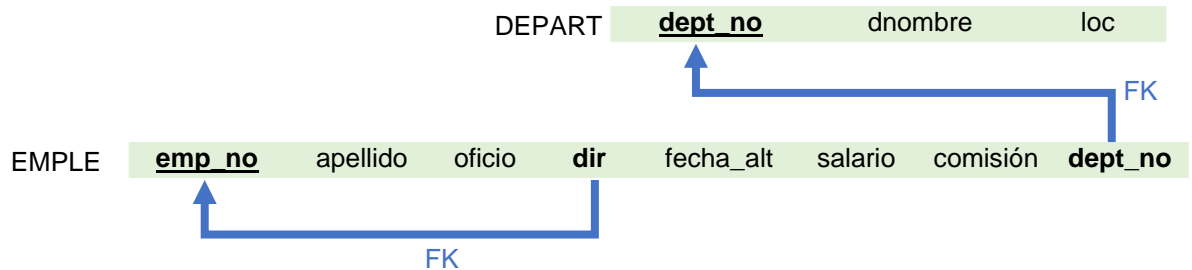
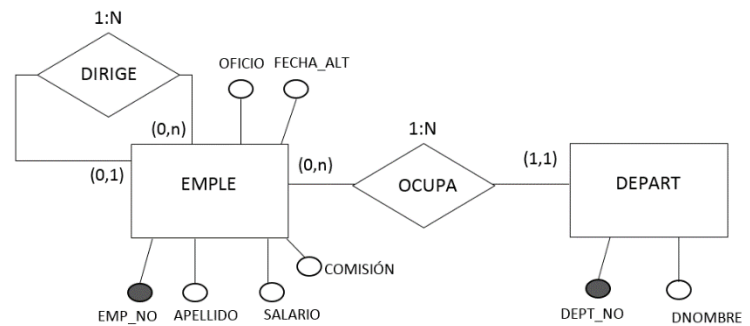
En este documento, a no ser que se indique lo contrario, se reflejan los comandos SQL referidos al Sistema Gestos de Base de Datos (SGBD) Oracle Database.

1. Consultas de los datos

Actividad. Crear tablas EMPLE y DEPART

Ejecuta los comandos para crear las tablas EMPLE y DEPART, que representan la base de datos de los empleados de una empresa.

Familiarízate con estas tablas y su significado, ya que las emplearemos frecuentemente en los ejemplos de clase.



Campos de DEPART (tabla de departamentos):

- dept_no: número que identifica a un departamento de la empresa
- dnombre: nombre del departamento
- loc: localización del departamento (ciudad en la que se encuentra)

Campos de EMPLE (tabla de empleados):

- emp_no: número que identifica a un empleado de la empresa
- apellido: apellido del empleado
- oficio: oficio del empleado
- dir: director del empleado
- fecha_alt: fecha de alta del empleado en la empresa
- salario: salario que cobra el empleado
- comisión: comisión que recibe el empleado por ciertas tareas
- dept_no: departamento al que pertenece el empleado

Según las tablas, contesta mentalmente a las preguntas:

¿Puede un empleado pertenecer a dos departamentos?

¿Puede un departamento tener más de un empleado?

¿Puede un departamento no tener ningún empleado?

¿Puede un empleado ser director de muchos empleados?

¿Puede un empleado no ser director de nadie?

¿El salario de un empleado depende del departamento al que pertenece?

Dos empleados del mismo departamento y con el mismo oficio, ¿deben tener el mismo salario?

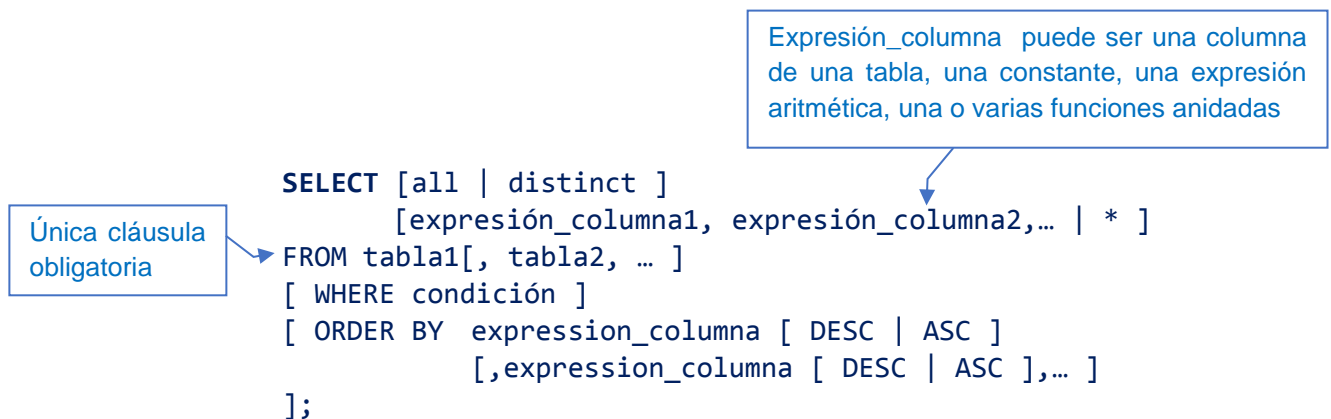
1.1. Comando Select

Para recuperar información de una base de datos relacional se emplea el comando **SELECT**.

Una característica fundamental de este comando es que el usuario indica qué quiere obtener, no dónde ni cómo (de esa complejidad se encarga el sistema gestor de la base de datos).

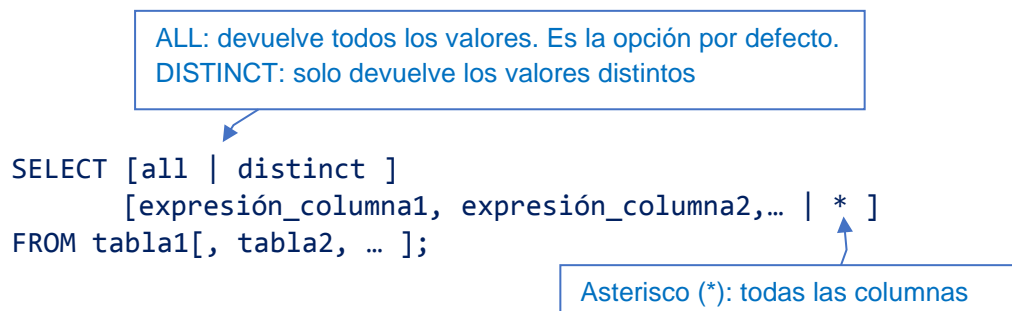
El comando SELECT es muy potente, permitiendo hacer búsquedas complejas de información.

La sintaxis básica es la siguiente (existen más cláusulas que las mostradas aquí).

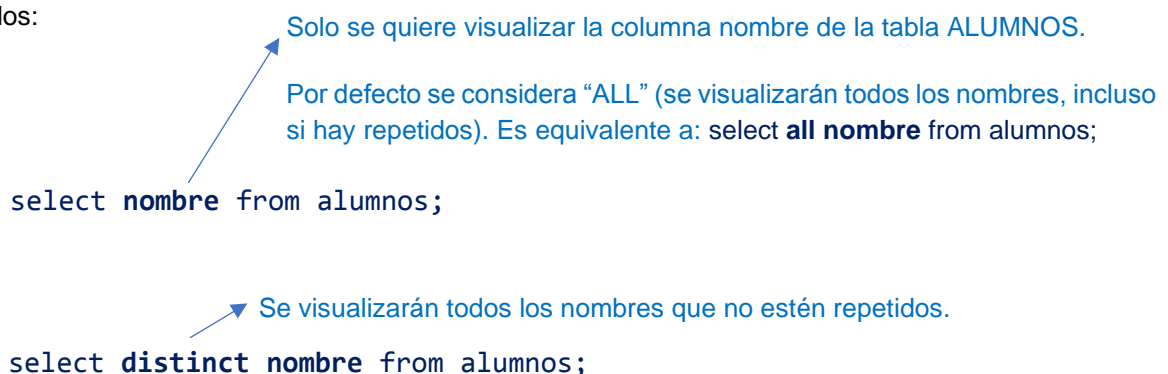


▪ Selección de columnas

Cuando se realiza una consulta a una o varias tablas, es posible indicar las columnas que se desean visualizar. El formato que permite seleccionar las columnas es el siguiente.



Ejemplos:



Listado de columnas que se quieren visualizar, separadas por comas.

```
select nombre, apellidos from alumnos;
```

Visualizar todas las columnas de la tabla

```
select * from alumnos;
```

Actividad. Selección de columnas

- En la tabla EMPLE, visualiza todos los empleados de la empresa (todos los datos de cada uno)
- Visualiza, para cada empleado, su número de empleado, apellido y el salario que cobra.
- Visualiza todos los apellidos de los empleados, sin que haya filas repetidas.

▪ Cláusula ORDER BY

Especifica el orden de visualización del listado resultante.

DESC: descendente
ASC: ascendente. Es la opción por defecto.

```
ORDER BY expression_columna [ DESC | ASC ]  
[,expression_columna [ DESC | ASC ],... ]
```

Se puede ordenar por varios campos

Ejemplos:

Se ordenará el listado por orden ascendente del nombre de los alumnos

```
select * from alumnos order by nombre;
```

Se ordenará el listado por orden ascendente de la columna 2

```
select dni, nombre from alumnos order by 2;
```

Se ordenará el listado por orden ascendente del nombre, y a nombres iguales se ordenará por el apellido


```
select * from alumnos order by nombre, apellido;
```

Actividad. Ordenación


- En la tabla EMPLE, visualiza todos los empleados de la empresa (todos los datos de cada uno), ordenados en orden ascendente por su apellido.
- Modifica el comando para que el listado sea descendente.
- Visualiza todos los apellidos de los empleados (solamente ese campo), ordenados descendientemente por su salario.
- Visualiza el número de empleado, apellido, oficio y salario de todos los empleados. El listado debe aparecer ordenado ascendentemente por oficios. Dentro de cada oficio, los empleados deben aparecer por orden descendente de su salario.

▪ Cláusula FROM

Única cláusula obligatoria del comando SELECT. Especifica la **tabla o lista de tablas de las que se recuperarán los datos**. Ejemplos:


`select nombre from alumnos;`


Si el usuario que realiza la consulta no es el propietario de la tabla que quiere consultar, debe usar el formato:
nombre_usuario.nombre_tabla (debe tener permisos para verla)


`select nombre from nbaezag.alumnos;`

En ocasiones se utilizan **alias de tabla**: se da un nuevo nombre a la tabla, que suele ser más corto (incluso una sola letra). Este alias solo tiene sentido dentro del propio comando SELECT, es decir, ese alias no puede usarse en ningún otro sitio.

Las utilidades de emplear un alias de tabla son:

- ▶ Reducir el nombre de la tabla cuando es largo
- ▶ Crear referencias a la misma tabla (se verá más adelante)



`select A.nombre from alumnos A;`


▪ Selección de filas: cláusula WHERE

Selecciona o filtra las filas que se quieren visualizar, en función de una condición.

La complejidad de la condición puede ser muy alta, anidando incluso distintas condiciones mediante operadores lógicos. Ejemplos:

`select nombre from alumnos where edad=18;`  Se visualizarán los nombres de los alumnos que tiene 18 años

`select nombre from alumnos where edad=18 and nacionalidad='española';`  Se visualizarán los nombres de los alumnos que tiene 18 años y son españoles (las dos cosas a la vez).

`select nombre from alumnos where edad=18 or nacionalidad='española';`  Se visualizarán los nombres de los alumnos que tiene 18 años o son españoles (con cumplir una condición es suficiente).

▪ Diferencia entre selección de filas y selección de columnas

- ▶ La selección de filas se realiza imponiendo una condición con la cláusula WHERE.
- ▶ La selección de columnas se realiza indicando las columnas que se quieren visualizar después de SELECT.

Ejemplo. Se tiene la tabla PERSONAS:

NOMBRE	EDAD	PESO
Alba	35	65
Fernando	48	98
Ismael	19	75

- 1) Selección de columnas: se quiere visualizar el nombre y la edad.

`select nombre, edad from personas;`

NOMBRE	EDAD	PESO
Alba	35	65
Fernando	48	98
Ismael	19	75

- 2) Selección de filas: se quieren visualizar solo personas con cierto peso.

`select * from personas where peso>70;`

NOMBRE	EDAD	PESO
Alba	35	65
Fernando	48	98
Ismael	19	75

- 3) Selección de filas y columnas.

`select nombre, edad from personas where peso>70;`

NOMBRE	EDAD	PESO
Alba	35	65
Fernando	48	98
Ismael	19	75

Los datos en azul serán el resultado. Observa que no se visualiza "peso", que es el criterio por el que se han filtrado las filas.

▪ Alias de columna

Por defecto al hacer una selección de datos aparece como título de la columna el nombre del campo de la tabla. Para mejorar el formato de presentación existe la posibilidad de modificar esos títulos en el listado empleando [alias de columna](#). No hay que confundirlo con los [alias de tabla](#).

Los alias de columna se indican entre **comillas dobles** después del nombre del campo.

Ejecuta el siguiente comando:

```
select dnombre "Departamento", dept_no "Número de departamento" from depart;
```

Actividad. Select simple (tabla EMPLE)

- a) Visualiza todos los empleados del departamento 20. El listado debe aparecer ordenado por la columna "apellido". Los campos que aparecerán son: número de empleado, apellido, oficio y número de departamento. En el listado el número de departamento aparecerá con el título "departamento".
- b) Consulta los empleados cuyo oficio sea "ANALISTA" ordenado por el número de empleado.
- c) Selecciona los empleados del departamento 20 que sean analistas. La consulta se ha de ordenar descendientemente por apellido y descendientemente por número de empleado.

Actividad. Select simple (tabla ALUM22)

- a) La tabla ALUM22 contiene los alumnos matriculados para el curso 2022 en un centro de enseñanza. Visualiza todos los alumnos.
- b) Consulta el dni, nombre, apellidos, curso, nivel y clase de todos los alumnos, ordenados por nombre y apellidos.
- c) Obtén todos los datos de los alumnos de la provincia de Guadalajara.
- d) Obtén el nombre, apellidos y fecha de nacimiento de los alumnos de Guadalajara ordenados del más joven al más mayor.

Ejercicios: Select simple. Ejercicio 1.

2. Operadores

2.1. Operadores aritméticos

Operadores aritméticos	
+	suma
-	resta
*	multiplicación
/	división

Los operadores pueden emplearse en diversos sitios dentro de la instrucción SELECT, por ejemplo:

```
select col1*col2, col1-col2 from tabla where col1+col2=34;
```

Actividad. Operadores aritméticos

- La tabla NOTAS_ALUMNOS contiene notas trimestrales de alumnos. Carga la tabla y sus datos y visualiza todos los registros.
- Visualiza la nota media de cada alumno con el siguiente comando. Observa que hay un alias de columna.

```
select nombre_alumno, (nota1+nota2+nota3)/3 "Nota media" from notas_alumnos;
```
- Vamos a aplicar una función de redondeo a la nota media.

round (m,n) devuelve el número m redondeado a n decimales

Ejecuta el comando:

```
select nombre_alumno, round((nota1+nota2+nota3)/3,2) "Nota media" from notas_alumnos;
```

2.2. Operadores de comparación y lógicos

Operadores de comparación	
=	Igual a
>	Mayor que
>=	Mayor o igual que
<	Menor que
<=	Menos o igual que
!= <>	Distinto de

Operadores lógicos	
and	Devuelve TRUE si las dos condiciones son verdaderas
or	Devuelve TRUE si al menos una de las dos condiciones es verdadera
not	Devuelve TRUE si la condición es falsa y FALSE si es verdadera

Los operadores lógicos permiten realizar una condición compuesta de varias subcondiciones. Cada una de estas subcondiciones debe ser una condición válida y completa. Ejemplo:

```
select * from tabla where campo1 < 10 or > 50 ; ❌
```

```
select * from tabla where campo1 < 10 or campo1 > 50 ; ✅
```

Actividad. Operadores de comparación y lógicos

A partir de la tabla NOTAS_ALUMNOS, obtén los nombres de los alumnos que tengan un 7 en la primera evaluación (nota1) y cuya media sea mayor que 6.

Los operadores lógicos se pueden combinar de forma ilimitada, sin embargo hay que tener cuidado con su anidación. Se debe emplear paréntesis cuando sea necesario.

Orden de prioridad de los operadores lógicos:

1. NOT
2. AND
3. OR

Ejemplo:

A partir de la tabla EMPLE, se desea obtener el apellido, salario y número de departamento de los empleados cuyo salario sea mayor de 2000 y su departamento sea el 10 o el 20.

¿Cuál de los siguientes comandos es el correcto?

```
select apellido, salario, dept_no from emple where
salario > 2000 and (dept_no = 20 or dept_no = 30);
```

⚡ APELLIDO	⚡ SALARIO	⚡ DEPT_NO
JIMENEZ	2900	20
FERNANDEZ	3000	20
NEGRO	3005	30
GIL	3000	20

```
select apellido,salario,dept_no from emple where
salario > 2000 and dept_no = 20 or dept_no = 30;
```

⚡ APELLIDO	⚡ SALARIO	⚡ DEPT_NO
JIMENEZ	2900	20
FERNANDEZ	3000	20
NEGRO	3005	30
ARROYO	1500	30
SALA	1625	30
MARTIN	1600	30
GIL	3000	20
TOVAR	1350	30
JIMENO	1335	30

2.3. Operadores de comparación de cadenas de caracteres

Para comparar caracteres (textos) puede emplearse el operador “igual a” (=).

Pero además de esta posibilidad, las cadenas de caracteres permiten utilizar caracteres especiales en las cadenas de comparación (_ y %). En este caso no se utilizará el “igual que” (=) sino el operador “like”.

Operadores de comparación de cadenas de caracteres	
like	Compara con un texto
_	Cualquier carácter
%	Cualquier cadena de caracteres

Algunos ejemplos:

<code>like 'comercial'</code>	Texto “comercial”. En este caso podría haberse usado <code>= 'comercial'</code>
<code>like 'S%'</code>	Cualquier texto que comience por S
<code>like '%S%'</code>	Cualquier texto que contenga una S
<code>like '___S'</code>	Cualquier texto de 3 caracteres que termine por S
<code>like 'S____'</code>	Cualquier texto de 4 caracteres que comience por S
<code>like '_S%'</code>	Cualquier texto cuyo segundo carácter sea una S

Hay que tener en cuenta que las cadenas de comparación son textos, por lo que deben ir entre comillas simples ('). Además dentro del texto las mayúsculas y minúsculas son significativas.

Actividad. Operadores de comparación de cadenas de caracteres

A partir de la tabla EMPLE, obtén los apellidos que:

- Comiencen por J
- Tengan una R en segunda posición
- Empiecen por A y contengan una O

Actividad. Diferencia entre CHAR y VARCHAR2.

- a) La tabla LIBRERIA contiene información sobre los ejemplares de libros almacenados en una librería. Para cada tema se indica el estante en el que se encuentra y el número de ejemplares. Carga la tabla y sus datos y visualiza todos los registros. Visualiza también la estructura de la tabla (comando DESCRIBE).
- b) Obtén los registros cuyo tema sea LABORES. Emplea el operador “igual que” (=). Repite la consulta empleando el operador “like”. Hazlo de forma que el resultado sea el mismo que en el caso anterior. Interpreta el resultado.
- c) A partir de la tabla ALUM22, obtén todos los alumnos que se llamen CONCHA. Emplea el operador “igual que” (=) y también en operador “like”.

2.4. Is Null / is not null

Operadores del valor null	
Is null	Comprueba si un valor es null
Is not null	Comprueba si un valor no es null (es distinto de null)

Estos operadores se utilizan para comprobar si el valor de un campo tiene el valor null (nulo) o no.

Ejemplo: `select apellido from emple where comision is not null;`

2.5. Operadores de comparación de conjuntos de valores

Operadores de conjuntos de valores	
in	Comprueba si un valor pertenece a un conjunto de valores (lista)
Between and	Comprueba si un valor está en un rango de valores

El operador “**in**” permite comprobar si una expresión pertenece o no (not) a un conjunto de valores (lista de posibilidades). Ejemplo:

```
select apellido from emple where dept_no in(10,30);
select apellido from emple where dept_no not in(10,30);
```

El operador “**between and**” permite comprobar si una expresión está contenida o no (not) en un rango de valores (dado un inicio y un final). Ejemplo:

```
select apellido from emple where salario between 1500 and 2000;
select apellido from emple where dept_no not between 1500 and 2000;
```

Los rangos de valores pueden ser de tipo numérico o texto (se considera el orden alfabético). Por ejemplo `between 'A' and 'D'` comprendería el rango: de A a D.

No hay que confundir el “and” del operador “`between ... and ...`” con el operador lógico “and”.

Actividad. Operadores de comparación de conjuntos de valores

- A partir de la tabla EMPLE, obtén los apellidos de los empleados cuyo oficio sea “vendedor”, “analista” o “desarrollador”. Emplea el operador OR.
- Repite la consulta anterior empleando un operador de conjunto de valores.
- Obtén los oficios que no sean ni “vendedor” ni “analista” ni “desarrollador”.
- Obtén todos los empleados cuyo salario está entre 2000 y 3000 euros.
- A partir de la tabla ALUM22, lista todos los nombres de alumnos comprendidos entre ANA y CONCHA.

2.6. Resumen de operadores

Operadores aritméticos	
+	suma
-	resta
*	multiplicación
/	división

Operadores lógicos	
and	Devuelve TRUE si las dos condiciones son verdaderas
or	Devuelve TRUE si al menos una de las dos condiciones es verdadera
not	Devuelve TRUE si la condición es falsa y FALSE si es verdadera

Operadores de comparación	
=	Igual a
>	Mayor que
>=	Mayor o igual que
<	Menor que
<=	Menos o igual que
!= <>	Distinto de

Operadores del valor null	
Is null	Comprueba si un valor es null
Is not null	Comprueba si un valor no es null (es distinto de null)

Operadores de comparación de cadenas de caracteres	
like	Compara con un texto
_	Cualquier carácter
%	Cualquier cadena de caracteres

Ejemplos. "valor" es un campo o una expresión:

valor LIKE 'Hola' se cumple sólo en el caso de que el valor sea 'Hola'

valor LIKE 'Tela_' se cumple para valor: 'Tela1', 'Tela2', 'TelaX', etc

valor LIKE 'A%' se cumple para cualquier valor que comience con A, por ejemplo: 'Anastasia', 'Aleteo', 'Andrajoso', 'A123'

Operadores de conjuntos de valores	
in	Comprueba si un valor pertenece a un conjunto de valores (lista)
Between and	Comprueba si un valor está en un rango de valores

Ejemplos. "valor" es un campo o una expresión:

valor in (2, 3, 8) se cumple sólo en el caso de que valor sea 2, 3 u 8

valor between 3 and 8 se cumple para valor: 3, 4, 5, 6, 7 u 8

valor between 'A' and 'C' se cumple para valor: 'A', 'B' o 'C'

Ejercicios: Select con operadores. Ejercicios 2 y 3.

3. Funciones

3.1. ¿Qué es una función?

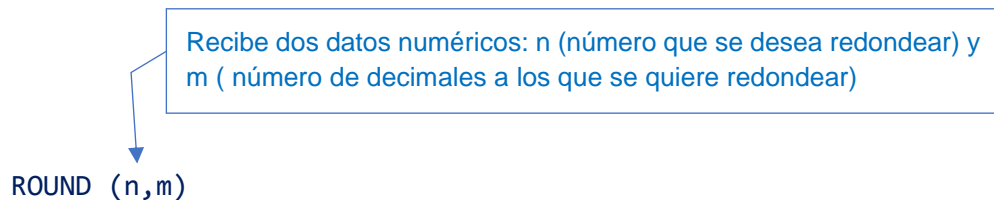
Las funciones son utilidades que reciben datos de entrada, los procesan y devuelven un resultado. Cuando se procesa la instrucción SQL automáticamente se sustituye la expresión de la función por su resultado.

Las funciones pueden **anidarse** (el resultado de una puede ser el dato de entrada de otra).

3.2. Funciones aritméticas

▪ ROUND

La función de redondeo **ROUND** es una función aritmética que tiene el formato:



Devuelve el número n redondeado a m decimales. Si m es un número negativo se realiza un redondeo a la unidad indicada (-1 redondea a las decenas, -2 redondea a las centenas, etc)

Ejemplos:

```
ROUND (226.28, 2) = 226.28
ROUND (226.28, 1) = 226.3
ROUND (226.28, 0) = 226
ROUND (226.28, -1) = 230
ROUND (226.28, -2) = 200
```

Algunos ejemplos de su uso:

```
select codigo_articulo, round(precio,2) from articulo;
select codigo_articulo from articulo where round(precio,0)=100;
select codigo_articulo from articulo order by round(precio,0);
```

Actividad. Función ROUND

- A partir de la tabla NOTAS_ALUMNOS, obtén la nota media de los alumnos redondeada a un decimal.
¿Por qué no se calcula la nota media en algunos alumnos?
- Muestra los nombres de alumnos cuya nota media redondeada a cero decimales sea mayor que 7. Muestra también, para cada uno, su nota media redondeada y sin redondear.

▪ NVL

La función **NVL** es una función que sustituye un valor nulo por otro valor. Es posible usarla con distintos tipos de datos (texto, numérico, fecha).

Recibe dos datos: si *n* tiene el valor **NULL**, la función devuelve *m*, en cualquier otro caso devuelve *n*.

NVL (*n*,*m*)

Hay que tener en cuenta que cuando un campo tiene un valor nulo, no se puede operar (el resultado será siempre **NULL**). Por ejemplo, **NULL + 1 = NULL**. Es interesante emplear la función **NVL** para dar un valor numérico a campos nulos, y de esta forma que se puedan realizar operaciones con ellos.

Ejecuta el siguiente comando para ver las notas de la primera evaluación de los alumnos en la tabla **NOTAS_ALUMNOS**:

```
select nombre_alumno, nota1 from notas_alumnos;
```

Vamos a hacer que si no hay nota para un alumno, aparecerá un 1. Observa el comando:

```
select nombre_alumno, nvl(nota1,1) from notas_alumnos;
```

Actividad. Función NVL

- A partir de la tabla **EMPLE**, muestra para cada empleado su número de empleado, apellido y el sueldo completo que percibe (el sueldo completo es la suma de salario y comisión).
¿Qué ocurre con los empleados que no tienen configurada la comisión?
- Añade al apartado anterior la función **NVL** para que, en los empleados sin comisión configurada, ésta se considere cero.
- A partir de la tabla **NOTAS_ALUMNOS**, calcula la nota media de todos los alumnos (si falta alguna nota, se considerará como 0 para la media)

3.3. Funciones de agregado

Estas funciones tienen como dato de entrada un campo de una tabla, realizando una operación sobre los valores de los registros para ese campo.

DNI	Nombre	Edad
123	Ana	35
345	Eva	22
876	Lara	18

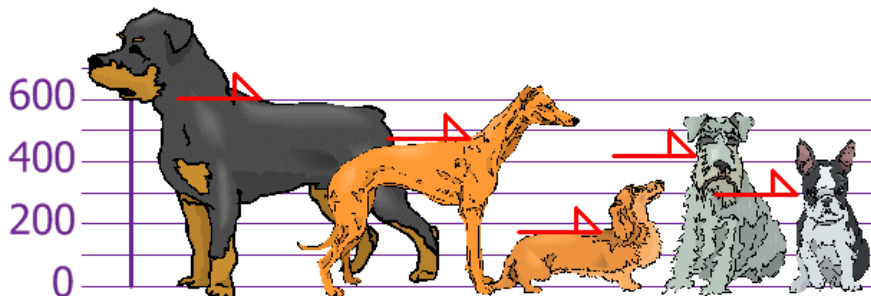
Actúan sobre una columna (todos sus valores o parte de ellos)

Vamos a estudiar estas funciones con un ejemplo. Tenemos una tabla con ejemplares de perro:

```
create table PERRO(
  nombre      varchar2(30) primary key,
  altura_mm   number(3)
);
insert into PERRO values ('Lassie', 600);
insert into PERRO values ('Socio', 470);
insert into PERRO values ('Canela', 170);
insert into PERRO values ('Lanas', 430);
insert into PERRO values ('Ron', 300);
```

NOMBRE	ALTURA_MM
Lassie	600
Socio	470
Canela	170
Lanas	430
Ron	300

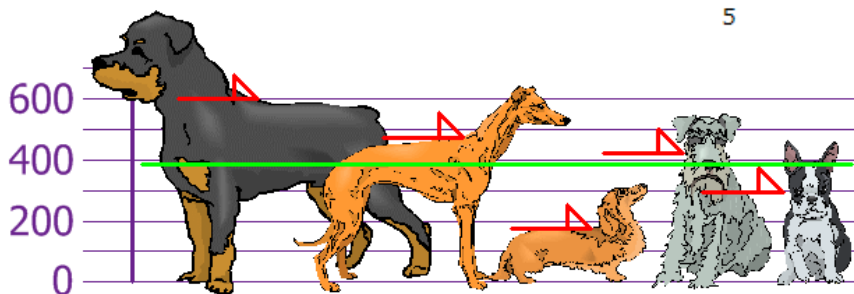
El campo altura_mm se refiere a la altura en mm desde los hombros del animal:



▪ AVG

La función **AVG** (Average) es la media aritmética, ignorando los valores nulos. En el ejemplo de la tabla PERRO:

$$\text{Media} = \frac{600 + 470 + 170 + 430 + 300}{5} = \frac{1970}{5} = 394$$



```
select avg(altura_mm) from perro;
```

```
SQL> select avg(altura_mm) from perro;

AVG(ALTURA_MM)
-----
              394
```

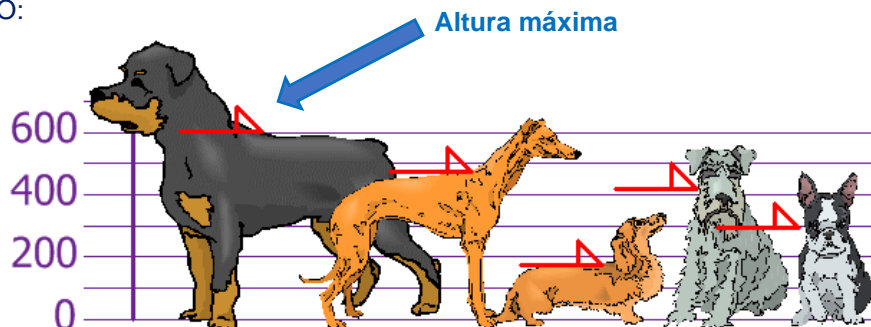
Actividad. Función AVG

- A partir de la tabla EMPLE, calcula el salario medio de los empleados del departamento 10.
- Muestra la media anterior redondeada sin decimales

MAX

La función **MAX** (MAXIMUM) devuelve el valor máximo.

En el ejemplo de la tabla PERRO:



```
select max(altura_mm) from perro;
```

```
SQL> select max(altura_mm) from perro;  
  
MAX(ALTURA_MM)  
-----  
600
```

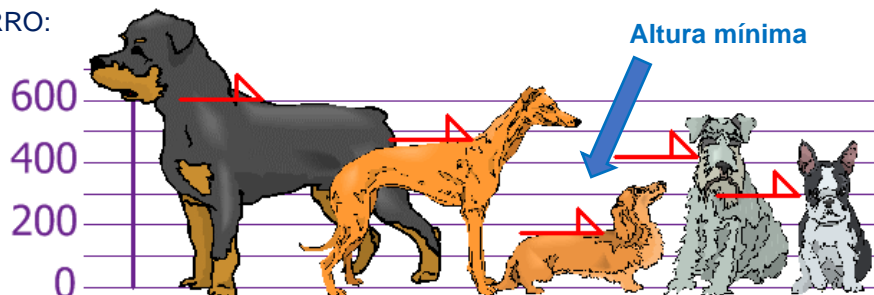
Actividad. Función MAX

A partir de la tabla EMPLE, muestra el máximo salario de la empresa.

MIN

La función **MIN** (MINIMUM) devuelve el valor mínimo.

En el ejemplo de la tabla PERRO:



```
select min(altura_mm) from perro;
```

```
SQL> select min(altura_mm) from perro;  
  
MIN(ALTURA_MM)  
-----  
170
```

Actividad. Función MIN

- A partir de la tabla EMPLE, muestra el menor salario de la empresa.
- Muestra el menor salario del departamento 30.

SUM

La función **SUM** (SUMA) devuelve la suma de valores.

En el ejemplo de la tabla PERRO:

$$600 + 470 + 170 + 430 + 300 = 1970$$

```
Select sum(altura_mm) from perro;
```

```
SQL> select sum(altura_mm) from perro;

SUM(ALTURA_MM)
-----
1970
```

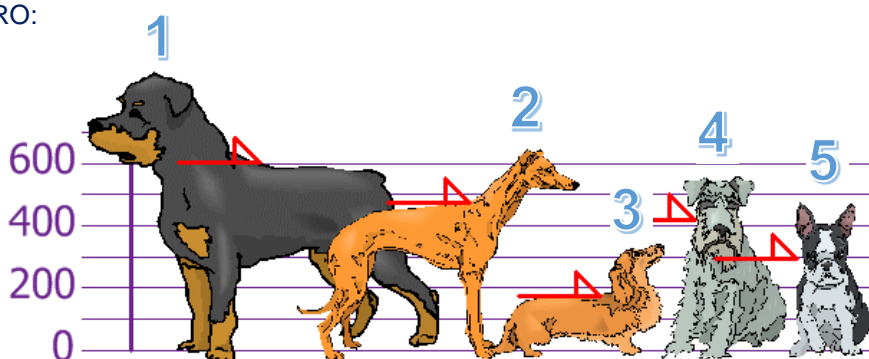
Actividad. Función SUM

- A partir de la tabla EMPLE, obtén la suma de todos los salarios de la empresa.
- Obtén la suma de las comisiones de los empleados del departamento 30.

COUNT

Esta función devuelve el número de registros, es decir, cuenta líneas.

En el ejemplo de la tabla PERRO:



```
Select count(*) from perro;
```

```
SQL> select count(*) from perro;

COUNT(*)
-----
5
```

Actividad. Función COUNT(*)

- A partir de la tabla EMPLE, obtén el número de empleados de la empresa.
- Obtén el número de empleados del departamento 30.

Utilización de COUNT

El formato de todas las posibilidades del uso de esta función es:

COUNT (* | [DISTINCT | ALL] campo)

- ▶ COUNT(*) cuenta todos los registros
- ▶ COUNT(campo) cuenta todos los registros en los que el valor del campo no es nulo
- ▶ COUNT (ALL campo) se cuentan todas las filas en las que el campo no es nulo. Alles la opción por defecto, por lo que es equivalente a COUNT(campo)
- ▶ COUNT (DISTINCT campo): se cuentan todas las filas en las que el campo no es nulo, y es distinto (si hay algún valor repetido se cuenta una única vez)

Actividad. Función COUNT(campo)

- a) Visualiza toda la tabla EMPLE ordenada por oficio, y averigua mentalmente cuántos oficios distintos hay.

```
select * from emple order by oficio;
```

Mediante una consulta, obtén el número de oficios que hay en la empresa y comprueba que coincide el resultado.

- b) Visualiza cuántos empleados tienen un apellido que comienza por la letra A.
c) Visualiza cuántos apellidos en la empresa comienzan por la letra A.
d) Compara y explica el resultado de la ejecución de los siguientes comandos:

```
select count(*) from emple;  
select count(apellido) from emple;  
select count (comision) from emple;
```

Ejercicios: Select con funciones. Ejercicios 4 y 5.