

EJEMPLOS

SELECT y WHERE

Tenemos la siguiente tabla, con los siguientes registros:

```
create table agenda(  
    apellido varchar(30),  
    nombre varchar(30),  
    domicilio varchar(30),  
    telefono varchar(11)  
);  
  
insert into agenda(apellido,nombre,domicilio,telefono) values  
    ('Acosta', 'Ana', 'Colon 123', '4234567');  
insert into agenda(apellido,nombre,domicilio,telefono) values  
    ('Bustamante', 'Betina', 'Avellaneda 135', '4458787');  
insert into agenda(apellido,nombre,domicilio,telefono) values  
    ('Lopez', 'Hector', 'Salta 545', '4887788');  
insert into agenda(apellido,nombre,domicilio,telefono) values  
    ('Lopez', 'Luis', 'Urquiza 333', '4545454');  
insert into agenda(apellido,nombre,domicilio,telefono) values  
    ('Lopez', 'Marisa', 'Urquiza 333', '4545454');
```

1. Realiza un "select" mostrando solamente el apellido y teléfono de todos los registros de agenda

```
SELECT apellido, telefono FROM agenda;
```

2. Muestra el nombre y apellido de de todos los registros de agenda.

```
SELECT nombre, apellido FROM agenda;
```

3. Selecciona todos los registros de la tabla

```
SELECT * FROM agenda;
```

4. Selecciona el registro cuyo nombre sea "Marisa"

```
SELECT * FROM agenda WHERE nombre='Marisa';
```

5. Selecciona los nombres y domicilios de quienes tengan apellido igual a "Lopez"

```
SELECT nombre, domicilio FROM agenda WHERE apellido='Lopez'
```

6. Selecciona los nombres y domicilios de quienes tengan apellido igual a "lopez" (en minúsculas). No aparece ningún registro, ya que la cadena "Lopez" no es igual a la cadena "lopez".

```
SELECT nombre, domicilio FROM agenda WHERE apellido='lopez'
```

7. Muestra el nombre de quienes tengan el teléfono "4545454"

```
SELECT nombre FROM agenda WHERE telefono='4545454'
```

ORDER BY

Tenemos la siguiente tabla con los siguientes registros:

```
create table libros(  
    titulo varchar(40) not null,  
    autor varchar(20) default 'Desconocido',  
    editorial varchar(20),  
    edicion date,  
    precio numeric(6,2)
```

```
);
```

```
insert into libros values('El aleph','Borges','Emece','10/10/1980',25.33);
insert into libros values('Java en 10 minutos','Mario Molina','Siglo XXI','05/12/2005',50.65);
insert into libros values('Alicia en el pais de las maravillas','Lewis Carroll','Emece','29/11/2000',19.95);
insert into libros values('Alicia en el pais de las maravillas','Lewis Carroll','Planeta','27/11/2004',15);
```

1. Recuperamos los registros ordenados por el título:

```
select * from libros order by titulo;
```

2. Ordenamos los registros por el campo "precio", referenciando el campo por su posición en la lista de selección:

```
select titulo,autor,precio from libros order by 3;
```

3. Los ordenamos por "editorial", de mayor a menor empleando "desc":

```
select * from libros order by editorial desc;
```

4. Ordenamos por dos campos:

```
select * from libros order by titulo,editorial;
```

5. Ordenamos en distintos sentidos:

```
select * from libros order by titulo asc, editorial desc;
```

6. Podemos ordenar por un campo que no se lista en la selección:

```
select titulo, autor from libros order by precio;
```

7. Está permitido ordenar por valores calculados, lo hacemos:

```
select titulo, editorial,precio+(precio*0.1) as "precio con descuento" from
libros order by 3;
```

8. Mostramos el título y año de edición de todos los libros, ordenados por año de edición:

```
select titulo, extract (year from edicion) as edicion from libros order by 2;
```

OPERADORES ARITMÉTICOS Y DE CONCATENACIÓN

Trabajamos con la tabla "libros" de una librería.

```
create table libros(
codigo numeric(5),
titulo varchar(40) not null,
autor varchar(20) default 'Desconocido',
editorial varchar(20),
precio numeric(6,2),
cantidad numeric(4) default 0,
primary key (codigo)
);
```

```
insert into libros (codigo,titulo,autor,editorial,precio)
values(1,'El aleph','Borges','Emece',25);
insert into libros
values(2,'Java en 10 minutos','Mario Molina','Siglo XXI',50.40,100);
insert into libros (codigo,titulo,autor,editorial,precio,cantidad)
values(3,'Alicia en el pais de las maravillas','Lewis Carroll','Emece',15,50);
```

1. Queremos saber el total en dinero de cada libro:

```
select titulo, precio,cantidad, precio*cantidad from libros;
```

2. Queremos saber el precio de cada libro con un 10% de descuento:

```
select titulo,precio,precio-(precio*0.1) from libros;
```

3. Queremos una columna con el título y autor de cada libro:

```
select CONCAT(titulo, '-', autor) from libros;
```

4. Mostramos el título y precio de cada libro concatenados:

```
select titulo||' $'||precio from libros; (en Oracle)
```

```
select CONCAT(titulo, ' $', precio) from libros; (en MySQL)
```

OPERADORES AND, OR Y NOT

Trabajamos con la tabla "libros" de una librería.

```
create table libros(
  codigo numeric(5),
  titulo varchar(40) not null,
  autor varchar(20) default 'Desconocido',
  editorial varchar(20),
  precio numeric(6,2)
);
```

```
insert into libros
  values(1,'El aleph','Borges','Emece',15.90);
insert into libros
  values(2,'Antología poética','Borges','Planeta',39.50);
insert into libros
  values(3,'Java en 10 minutos','Mario Molina','Planeta',50.50);
insert into libros
  values(4,'Alicia en el pais de las maravillas','Lewis Carroll','Emece',19.90);
insert into libros
  values(5,'Martin Fierro','Jose Hernandez','Emece',25.90);
insert into libros
  values(6,'Martin Fierro','Jose Hernandez','Paidos',16.80);
insert into libros
  values(7,'Aprenda PHP','Mario Molina','Emece',19.50);
insert into libros
  values(8,'Cervantes y el quijote','Borges','Paidos',18.40);
```

1. Recuperamos los libros cuyo autor sea igual a "Borges" y cuyo precio no supere los 20 euros:

```
select * from libros where (autor='Borges') and (precio<=20);
```

2. Seleccionamos los libros cuyo autor es "Borges" y/o cuya editorial es "Planeta":

```
select * from libros where autor='Borges' or editorial='Planeta';
```

Nota que aparecen todos los libros de "Borges" y todos los libros de "Planeta", algunos cumplen ambas condiciones.

3. Recuperamos los libros cuya editorial NO es "Planeta":

```
select * from libros where not editorial='Planeta';
```

4. Veamos un ejemplo de cómo el empleo de paréntesis permite a Oracle evaluar en forma diferente ciertas consultas aparentemente iguales:

```
select * from libros where (autor='Borges') or(editorial='Paidos' and precio<20);
```

```
select * from libros where (autor='Borges' or editorial='Paidos') and(precio<20);
```

Nota que el primer resultado devuelve todos los libros de "Borges" (primera condición) y todos los libros de "Paidos" con precio inferior a 20 (segunda condición) (registros 1,2,6 y 8); la segunda recupera todos los libros de "Borges" o de "Paidos" (primera condición) cuyo precio sea inferior a 20 (segunda condición) (registros 1,6 y 8).

OPERADOR BETWEEN

Trabajamos con la tabla "libros" de una librería.

```
create table libros(
  codigo numeric(5) not null,
  titulo varchar(40) not null,
  autor varchar(20) default 'Desconocido',
  editorial varchar(20),
  edicion date,
  precio numeric(6,2)
);

insert into libros
  values(1,'El aleph','Borges','Emece','15/01/2000',15.90);
insert into libros
  values(2,'Cervantes y el quijote','Borges','Paidos',null,null);
insert into libros
  values(3,'Alicia en el pais de las maravillas','Lewis
Carroll',null,'25/03/2000',19.90);
insert into libros
  values(4,'Martin Fierro','Jose Hernandez','Emece','18/05/2000',25.90);
insert into libros (codigo,titulo,autor,edicion,precio)
  values(5,'Antología poética','Borges','25/08/2000',32);
insert into libros (codigo,titulo,autor,edicion,precio)
  values(6,'Java en 10 minutos','Mario Molina','11/02/2007',45.80);
insert into libros (codigo,titulo,autor,edicion,precio)
  values(7,'Martin Fierro','Jose Hernandez','23/11/2006',40);
insert into libros (codigo,titulo,autor,edicion,precio)
  values(8,'Aprenda PHP','Mario Molina','01/06/2007',56.50);
```

1. Recuperamos los registros cuyo precio esté entre 20 y 40 empleando "between":

```
select * from libros where precio between 20 and 40;
```

Nota que si el campo tiene el valor "null", no aparece en la selección.

2. Para seleccionar los libros cuyo precio NO esté entre un intervalo de valores antecede "not" al "between":

```
select * from libros where precio not between 20 and 40;
```

Nota que si el campo tiene el valor "null", no aparece en la selección.

3. Recuperamos los títulos y edición de los libros cuya fecha de edición se encuentre entre '01/05/2000' y '01/05/2007', ordenados por fecha de edición:

```
select titulo, edicion from libros where edicion between '01/05/2000' and
'01/05/2007' order by edicion;
```

OPERADOR IN

Trabajamos con la tabla "libros" de una librería.

```
create table libros(
  codigo numeric (5),
  titulo varchar(40) not null,
  autor varchar(20),
  editorial varchar(20),
  precio numeric(6,2)
);

insert into libros
  values(1,'El aleph','Borges','Emece',15.90);
insert into libros
  values(2,'Cervantes y el quijote','Borges','Paidos',null);
insert into libros
```

```

values(3,'Alicia en el pais de las maravillas','Lewis Carroll',null,19.90);
insert into libros
values(4,'Matematica estas ahi','Paenza','Siglo XXI',15);
insert into libros
values(5,'Antología poética',default,default,32);
insert into libros
values(6,'Martin Fierro','Jose Hernandez',default,40);
insert into libros
values(7,'Aprenda PHP','Mario Molina',default,56.50);

```

1. Recuperamos los libros cuyo autor es "Paenza" o "Borges":

```
select * from libros where autor in('Borges','Paenza');
```

Nota que los valores "null" no se consideran.

2. Recuperamos los libros cuyo autor NO es "Paenza" ni "Borges":

```
select * from libros where autor not in ('Borges','Paenza');
```

Nota que los valores "null" no se consideran.

3. Recuperamos los libros cuyo código se encuentre en la siguiente lista de valores (1,3,5,7,9):

```
select * from libros where codigo in (1,3,5,7,9);
```

OPERADOR LIKE

Trabajamos con la tabla "libros" de una librería.

```

create table libros(
  titulo varchar(40),
  autor varchar(20) default 'Desconocido',
  editorial varchar(20),
  edicion date,
  precio numeric(6,2)
);

insert into libros
values('El aleph','Borges','Emece','12/05/2005',15.90);
insert into libros
values('Antología poética','J. L. Borges','Planeta','16/08/2000',null);
insert into libros
values('Alicia en el pais de las maravillas','Lewis Carroll',null,'25/04/2000',19.90);
insert into libros
values('Matematica estas ahi','Paenza','Siglo XXI','21/12/2006',15);
insert into libros
values('Martin Fierro','Jose Hernandez',default,'22/09/2001',40);
insert into libros
values('Aprenda PHP','Mario Molina','Nuevo siglo','22/05/1999',56.50);
insert into libros
values(null,'Mario Molina','Nuevo siglo',null,45);

```

1. Recuperamos todos los libros que contengan en el campo "autor" la cadena "Borges":

```
select *from libros where autor like '%Borges%';
```

2. Seleccionamos los libros cuyos títulos comienzan con la letra "M":

```
select *from libros where titulo like 'M%';
```

Note que los valores nulos no se incluyen en la búsqueda.

3. Seleccionamos todos los títulos que NO comienzan con "M":

```
select *from libros where titulo not like 'M%';
```

4. Si queremos ver los libros de "Lewis Carroll" pero no recordamos si se escribe "Carroll" o "Carrolt", podemos emplear el comodín "_" (guión bajo) y establecer la siguiente condición:

```
select *from libros where autor like '%Carrol_';
```

5. Recuperamos todos los libros que contengan en el campo "edicion", en la parte correspondiente al mes, la cadena "05":

```
select titulo,edicion from libros where edicion like '__/05%';
```

Nota que especificamos que los 2 primeros caracteres (2 guiones bajos) pueden ser cualquier caracter, luego una barra seguida de "05" y que finalice con cualquier número de caracteres.