6 - Recuperar algunos registros (where)

Hemos aprendido a seleccionar algunos campos de una tabla.

También es posible recuperar algunos registros.

Existe una cláusula, "where" con la cual podemos especificar condiciones para una consulta "select". Es decir, podemos recuperar algunos registros, sólo los que cumplan con ciertas condiciones indicadas con la cláusula "where". Por ejemplo, queremos ver el usuario cuyo nombre es "Marcelo", para ello utilizamos "where" y luego de ella, la condición:

```
select nombre, clave
  from usuarios
  where nombre='Marcelo';
```

La sintaxis básica y general es la siguiente:

```
 select NOMBRECAMPO1, ..., NOMBRECAMPOn
  from NOMBRETABLA
  where CONDICION;
```

Para las condiciones se utilizan operadores relacionales (tema que trataremos más adelante en detalle). El signo igual(=) es un operador relacional. Para la siguiente selección de registros especificamos una condición que solicita los usuarios cuya clave es igual a "River":

```
 select nombre,clave
  from usuarios
  where clave='River';
```

Si ningún registro cumple la condición establecida con el "where", no aparecerá ningún registro.

Entonces, con "where" establecemos condiciones para recuperar algunos registros.

Para recuperar algunos campos de algunos registros combinamos en la consulta la lista de campos y la cláusula "where":

```
 select nombre
  from usuarios
  where clave='River';
```

En la consulta anterior solicitamos el nombre de todos los usuarios cuya clave sea igual a "River".

Primer problema:

Trabaje con la tabla "agenda" en la que registra los datos de sus amigos.

1- Elimine "agenda"

2- Cree la tabla, con los siguientes campos: apellido (cadena de 30), nombre (cadena de 20), domicilio (cadena de 30) y telefono (cadena de 11):

```
create table agenda(
 apellido varchar2(30),
 nombre varchar2(30),
 domicilio varchar2(30),
 telefono varchar2(11)
);
```

3- Visualice la estructura de la tabla "agenda" (4 campos)
4- Ingrese los siguientes registros ("insert into"):

```
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');
```

5- Seleccione todos los registros de la tabla (5 registros)
6- Seleccione el registro cuyo nombre sea "Marisa" (1 registro)

7- Seleccione los nombres y domicilios de quienes tengan apellido igual a "Lopez" (3 registros)

8- Seleccione los nombres y domicilios de quienes tengan apellido igual a "lopez" (en minúsculas)

No aparece ningún registro, ya que la cadena "Lopez" noe s igual a la cadena "lopez".

9- Muestre el nombre de quienes tengan el teléfono "4545454" (2 registros)

7 - Operadores relacionales

Los operadores son símbolos que permiten realizar operaciones matemáticas, concatenar cadenas, hacer comparaciones.

Oracle reconoce de 4 tipos de operadores:
1) relacionales (o de comparación)
2) aritméticos
3) de concatenación
4) lógicos
Por ahora veremos solamente los primeros.

Los operadores relacionales (o de comparación) nos permiten comparar dos expresiones, que pueden ser variables, valores de campos, etc.

Hemos aprendido a especificar condiciones de igualdad para seleccionar registros de una tabla; por ejemplo:

```
 select *from libros
  where autor='Borges';
```

Utilizamos el operador relacional de igualdad.

Los operadores relacionales vinculan un campo con un valor para que Oracle compare cada registro (el campo especificado) con el valor dado.

Los operadores relacionales son los siguientes:

=	igual
<>	distinto
>	mayor
<	menor
>=	mayor o igual
<=	menor o igual

Podemos seleccionar los registros cuyo autor sea diferente de "Borges", para ello usamos la condición:

```
 select * from libros
  where autor<>'Borges';
```

Podemos comparar valores numéricos. Por ejemplo, queremos mostrar los títulos y precios de los libros cuyo precio sea mayor a 20 pesos:

```
 select titulo, precio
  from libros
  where precio>20;
```

Queremos seleccionar los libros cuyo precio sea menor o igual a 30:

```
 select *from libros
  where precio<=30;
```

Los operadores relacionales comparan valores del mismo tipo. Se emplean para comprobar si un campo cumple con una condición.
No son los únicos, existen otros que veremos más adelante.

Primer problema:

Un comercio que vende artículos de computación registra los datos de sus artículos en una tabla con ese nombre.

1- Elimine "artículos"

2- Cree la tabla, con la siguiente estructura:

```
create table articulos(
 codigo number(5),
 nombre varchar2(20),
 descripcion varchar2(30),
 precio number(6,2),
 cantidad number(3)
);
```

3- Vea la estructura de la tabla.

4- Ingrese algunos registros:

```
insert into articulos (codigo, nombre, descripcion, precio,cantidad)
 values (1,'impresora','Epson Stylus C45',400.80,20);
insert into articulos (codigo, nombre, descripcion, precio,cantidad)
 values (2,'impresora','Epson Stylus C85',500,30);
insert into articulos (codigo, nombre, descripcion, precio,cantidad)
 values (3,'monitor','Samsung 14',800,10);
insert into articulos (codigo, nombre, descripcion, precio,cantidad)
 values (4,'teclado','ingles Biswal',100,50);
insert into articulos (codigo, nombre, descripcion, precio,cantidad)
 values (5,'teclado','español Biswal',90,50);
```

5- Seleccione los datos de las impresoras (2 registros)

6- Seleccione los artículos cuyo precio sea mayor o igual a 400 (3 registros)

7- Seleccione el código y nombre de los artículos cuya cantidad sea menor a 30 (2 registros)

8- Selecciones el nombre y descripción de los artículos que NO cuesten $100 (4 registros).

8 - Borrar registros (delete)

Para eliminar los registros de una tabla usamos el comando "delete".

Sintaxis básica:

```
 delete from NOMBRETABLA;
```

Se coloca el comando delete seguido de la palabra clave "from" y el nombre de la tabla de la cual queremos eliminar los registros. En el siguiente ejemplo se eliminan los registros de la tabla "usuarios":

```
delete from usuarios;
```

Luego, un mensaje indica la cantidad de registros que se han eliminado.

Si no queremos eliminar todos los registros, sino solamente algunos, debemos indicar cuál o cuáles; para ello utilizamos el comando "delete" junto con la clausula "where" con la cual establecemos la condición que deben cumplir los registros a borrar.

Por ejemplo, queremos eliminar aquel registro cuyo nombre de usuario es "Marcelo":

```
 delete from usuarios
 where nombre='Marcelo';
```

Si solicitamos el borrado de un registro que no existe, es decir, ningún registro cumple con la condición especificada, aparecerá un mensaje indicando que ningún registro fue eliminado, pues no encontró registros con ese dato.

Tenga en cuenta que si no colocamos una condición, se eliminan todos los registros de la tabla especificada.

Primer problema:
Trabaje con la tabla "agenda" que registra la información referente a sus amigos.

1- Elimine la tabla.

2- Cree la tabla con los siguientes campos: apellido (cadena de 30), nombre (cadena de 20), domicilio (cadena de 30) y telefono (cadena de 11):

```
create table agenda(
 apellido varchar2(30),
 nombre varchar2(20),
 domicilio varchar2(30),
 telefono varchar2(11)
);
```

3- Ingrese los siguientes registros (insert into):

```
insert into agenda(apellido,nombre,domicilio,telefono) values
  ('Alvarez','Alberto','Colon 123','4234567');
insert into agenda(apellido,nombre,domicilio,telefono) values
  ('Juarez','Juan','Avellaneda 135','4458787');
insert into agenda(apellido,nombre,domicilio,telefono) values
  ('Lopez','Maria','Urquiza 333','4545454');
insert into agenda(apellido,nombre,domicilio,telefono) values
  ('Lopez','Jose','Urquiza 333','4545454');
insert into agenda(apellido,nombre,domicilio,telefono) values
  ('Salas','Susana','Gral. Paz 1234','4123456');
```

4- Elimine el registro cuyo nombre sea "Juan" (1 registro)

5- Elimine los registros cuyo número telefónico sea igual a "4545454" (2 registros)

6- Elimine todos los registros (2 registros)

9 - Actualizar registros (update)

Decimos que actualizamos un registro cuando modificamos alguno de sus valores.

Para modificar uno o varios datos de uno o varios registros utilizamos "update" (actualizar).

Sintaxis básica:

```
 update NOMBRETABLA set CAMPO=NUEVOVALOR;
```

Utilizamos "update" junto al nombre de la tabla y "set" junto con el campo a modificar y su nuevo valor.

El cambio afectará a todos los registros.

Por ejemplo, en nuestra tabla "usuarios", queremos cambiar los valores de todas las claves, por "RealMadrid":

```
 update usuarios set clave='RealMadrid';
```

Podemos modificar algunos registros, para ello debemos establecer condiciones de selección con "where".

Por ejemplo, queremos cambiar el valor correspondiente a la clave de nuestro usuario llamado "Federicolopez", queremos como nueva clave "Boca", necesitamos una condición "where" que afecte solamente a este registro:

```
 update usuarios set clave='Boca'
  where nombre='Federicolopez';
```

Si Oracle no encuentra registros que cumplan con la condición del "where", un mensaje indica que ningún registro fue modificado.

Las condiciones no son obligatorias, pero si omitimos la cláusula "where", la actualización afectará a todos los registros.

También podemos actualizar varios campos en una sola instrucción:

```
 update usuarios set nombre='Marceloduarte', clave='Marce'
  where nombre='Marcelo';
```

Para ello colocamos "update", el nombre de la tabla, "set" junto al nombre del campo y el nuevo valor y separado por coma, el otro nombre del campo con su nuevo valor.

Primer problema:

Trabaje con la tabla "agenda" que almacena los datos de sus amigos.

1- Elimine la tabla y créela con la siguiente estructura:

```
drop table agenda;

create table agenda(
 apellido varchar2(30),
 nombre varchar2(20),
 domicilio varchar2(30),
 telefono varchar2(11)
);
```

3- Ingrese los siguientes registros:

```
insert into agenda (apellido,nombre,domicilio,telefono)
 values ('Acosta','Alberto','Colon 123','4234567');
insert into agenda (apellido,nombre,domicilio,telefono)
 values ('Juarez','Juan','Avellaneda 135','4458787');
insert into agenda (apellido,nombre,domicilio,telefono)
 values ('Lopez','Maria','Urquiza 333','4545454');
insert into agenda (apellido,nombre,domicilio,telefono)
 values ('Lopez','Jose','Urquiza 333','4545454');
insert into agenda (apellido,nombre,domicilio,telefono)
 values ('Suarez','Susana','Gral. Paz 1234','4123456');
```

4- Modifique el registro cuyo nombre sea "Juan" por "Juan Jose" (1 registro actualizado)

5- Actualice los registros cuyo número telefónico sea igual a "4545454" por "4445566" (2 registros)

6- Actualice los registros que tengan en el campo "nombre" el valor "Juan" por "Juan Jose" (ningún registro afectado porque ninguno cumple con la condición del "where")

10 – Comentarios

Para aclarar algunas instrucciones, en ocasiones, necesitamos agregar comentarios.

Es posible ingresar comentarios en la línea de comandos, es decir, un texto que no se ejecuta; para ello se emplean dos guiones (--):

```
 select *from libros;--mostramos los registros de libros
```

en la línea anterior, todo lo que está luego de los guiones (hacia la derecha) no se ejecuta.

Para agregar varias líneas de comentarios, se coloca una barra seguida de un asterisco (/*) al comienzo del bloque de comentario y al finalizarlo, un asterisco seguido de una barra (*/)

```
 select titulo, autor
 /*mostramos títulos y
 nombres de los autores*/
 from libros;
```

todo lo que está entre los símbolos "/*" y "*/" no se ejecuta.