

62 - Varias tablas (join)

Problema:

Una empresa tiene registrados sus clientes en una tabla llamada "clientes", también tiene una tabla "provincias" donde registra los nombres de las provincias.

1- Elimine la tabla "clientes" y "provincias", si existen:
drop table if exists clientes, provincias;

2- Créelas con las siguientes estructuras:

```
create table clientes (  
  codigo int unsigned auto_increment,  
  nombre varchar(30) not null,  
  domicilio varchar(30),  
  ciudad varchar(20),  
  codigoprovincia tinyint unsigned,  
  telefono varchar(11),  
  primary key(codigo)  
);  
  
create table provincias(  
  codigo tinyint unsigned auto_increment,  
  nombre varchar(20),  
  primary key (codigo)  
);
```

3- Ingrese algunos registros para ambas tablas:

```
insert into provincias (nombre) values('Cordoba');  
insert into provincias (nombre) values('Santa Fe');  
insert into provincias (nombre) values('Corrientes');  
insert into provincias (nombre) values('Misiones');  
insert into provincias (nombre) values('Salta');  
insert into provincias (nombre) values('Buenos Aires');  
insert into provincias (nombre) values('Neuquen');
```

```
insert into clientes (nombre,domicilio,ciudad,codigoProvincia,telefono)  
values ('Lopez Marcos', 'Colon 111', 'Córdoba',1,'null');  
insert into clientes (nombre,domicilio,ciudad,codigoProvincia,telefono)  
values ('Perez Ana', 'San Martin 222', 'Cruz del Eje',1,'4578585');  
insert into clientes (nombre,domicilio,ciudad,codigoProvincia,telefono)  
values ('Garcia Juan', 'Rivadavia 333', 'Villa Maria',1,'4578445');  
insert into clientes (nombre,domicilio,ciudad,codigoProvincia,telefono)  
values ('Perez Luis', 'Sarmiento 444', 'Rosario',2,null);  
insert into clientes (nombre,domicilio,ciudad,codigoProvincia,telefono)  
values ('Pereyra Lucas', 'San Martin 555', 'Cruz del Eje',1,'4253685');  
insert into clientes (nombre,domicilio,ciudad,codigoProvincia,telefono)  
values ('Gomez Ines', 'San Martin 666', 'Santa Fe',2,'0345252525');  
insert into clientes (nombre,domicilio,ciudad,codigoProvincia,telefono)  
values ('Torres Fabiola', 'Alem 777', 'Villa del Rosario',1,'4554455');  
insert into clientes (nombre,domicilio,ciudad,codigoProvincia,telefono)  
values ('Lopez Carlos', 'Irigoyen 888', 'Cruz del Eje',1,null);  
insert into clientes (nombre,domicilio,ciudad,codigoProvincia,telefono)  
values ('Ramos Betina', 'San Martin 999', 'Cordoba',1,'4223366');  
insert into clientes (nombre,domicilio,ciudad,codigoProvincia,telefono)  
values ('Lopez Lucas', 'San Martin 1010', 'Posadas',4,'0457858745');
```

4- Obtenga los datos de ambas tablas, use alias:

```
select c.nombre,c.domicilio,c.ciudad,p.nombre,c.telefono
```

```
from clientes as c
join provincias as p
on c.codigoProvincia=p.codigo;
```

5- Obtenga la misma información anterior pero ordenada por nombre del cliente:

```
select c.nombre,c.domicilio,c.ciudad,p.nombre,c.telefono
from clientes as c
join provincias as p
on c.codigoProvincia=p.codigo
order by c.nombre;
```

6- Omita la referencia a las tablas en la condición "on" para verificar que la sentencia no se ejecuta porque el nombre del campo "codigo" es ambiguo (ambas tablas lo tienen):

```
select c.nombre,c.domicilio,c.ciudad,p.nombre,c.telefono
from clientes as c
join provincias as p
on codigoProvincia=codigo
```

Ejecutar los comandos SQL

Borrar comando.

Finalizar

Otros problemas:

A) Un club dicta clases de distintos deportes. En una tabla llamada "socios" guarda los datos de sus socios y en una tabla denominada "inscriptos" almacena la información necesaria para las inscripciones de los socios a los distintos deportes.

1- Elimine las tablas si existen.

2- Cree las tablas:

```
create table socios(
documento char(8) not null,
nombre varchar(30),
domicilio varchar(30),
primary key(documento)
);
```

```
create table inscriptos(
documento char(8) not null,
deporte varchar(15) not null,
año year,
matricula char(1), /*si esta paga ='s' sino 'n'*/
primary key(documento,deporte,año)
);
```

3- Ingrese algunos registros para ambas tablas:

```
insert into socios values('22333444','Juan Perez','Colon 234');
insert into socios values('23333444','Maria Lopez','Sarmiento 465');
insert into socios values('24333444','Antonio Juarez','Caseros 980');
```

```
insert into inscriptos values ('22333444','natacion','2005','s');
insert into inscriptos values ('22333444','natacion','2006','n');
insert into inscriptos values ('23333444','natacion','2005','s');
insert into inscriptos values ('23333444','tenis','2006','s');
insert into inscriptos values ('23333444','natacion','2006','s');
insert into inscriptos values ('24333444','tenis','2006','n');
insert into inscriptos values ('24333444','basquet','2006','n');
```

4- Muestre el nombre del socio y todos los campos de la tabla "inscriptos":

```
select s.nombre,i.*
from socios as s
join inscriptos as i
on s.documento=i.documento;
```

5- Omita la referencia a las tablas en la condición "on" para verificar que la sentencia no se ejecuta porque el nombre del campo "documento" es ambiguo (ambas tablas lo tienen):

```
select s.nombre,i.*
from socios as s
join inscriptos as i
on documento=documento;
```

6- Muestre el nombre de los socios y los deportes en los cuales están inscriptos este año:

```
select s.nombre,i.deporte
from socios as s
join inscriptos as i
on s.documento=i.documento
where año=2006;
```

7- Muestre el nombre y todas las inscripciones del socio con número de documento='23333444':

```
select s.nombre,i.*
from socios as s
join inscriptos as i
on s.documento=i.documento
where s.documento='23333444';
```

B) Una pequeña biblioteca de barrio registra los préstamos de sus libros en una tabla llamada "prestamos" y los datos de sus libros en una tabla llamada "libros".

1- Elimine las tablas, si existen.

2- Cree las tablas:

```
create table libros(
codigo int unsigned auto_increment,
titulo varchar(40),
autor varchar (30),
editorial varchar (15),
primary key (codigo)
);

create table prestamos(
codigolibro int unsigned not null,
documento char(8) not null,
fechaprestamo date not null,
fechadevolucion date,
primary key(codigolibro,fechaprestamo)
);
```

3- Ingrese algunos registros para ambas tablas:

```
insert into libros values (15,'Manual de 1º grado','Moreno Luis','Emece');
insert into libros values (28,'Manual de 2º grado','Moreno Luis','Emece');
insert into libros values (30,'Alicia en el pais de las maravillas','Lewis Carroll','Planeta');
insert into libros values (35,'El aleph','Borges','Emece');

insert into prestamos
values(15,'22333444','2006-07-10','2006-07-12');
insert into prestamos
values(15,'22333444','2006-07-20','2006-07-21');
insert into prestamos (codigolibro,documento,fechaprestamo)
values(15,'23333444','2006-07-25');
insert into prestamos (codigolibro,documento,fechaprestamo)
values(30,'23333444','2006-07-28');
insert into prestamos (codigolibro,documento,fechaprestamo)
values(28,'25333444','2006-08-10');
```

4- Muestre todos los datos de los préstamos, incluyendo el nombre del libro (join con "libros"):

```
select l.titulo,p.*
from prestamos as p
join libros as l
on l.codigo=p.codigolibro;
```

5- Muestre la información de los préstamos del libro "Manual de 1º grado":

```
select p.documento,fechaprestamo,fechadevolucion
from prestamos as p
join libros as l
on l.codigo=p.codigolibro
where l.titulo='Manual de 1º grado';
```

6- Muestre los títulos de los libros, la fecha de préstamo y el documento del socio de todos los libros que no han sido devueltos:

```
select l.titulo,p.documento,p.fechaprestamo
from prestamos as p
join libros as l
on l.codigo=p.codigolibro
where p.fechadevolucion is null;
```

C) Una clínica registra las consultas de los pacientes en una tabla llamada "consultas" y en otra tabla denominada "obrassociales" almacena los datos de las obras sociales que atiende.

1- Elimine las tablas si existen.

2- Cree las tablas:

```
create table consultas(
fecha date,
hora time,
documento char(8) not null,
codigoobrasocial tinyint unsigned,
medico varchar(30),
primary key(fecha,hora,medico)
);
```

```
create table obrassociales(
codigo tinyint unsigned auto_increment,
nombre varchar(15),
monto decimal(5,2) unsigned,
primary key(codigo)
);
```

3- Ingrese algunos registros:

```
insert into obrassociales (nombre, monto)
values('PAMI', 2);
insert into obrassociales (nombre, monto)
values('IPAM', 5);
insert into obrassociales (nombre, monto)
values('OSDOP', 3);
```

```
insert into consultas values('2006-08-10', '8:00', '22333444', 1, 'Perez');
insert into consultas values('2006-08-10', '10:00', '22333444', 1, 'Lopez');
insert into consultas values('2006-08-10', '8:30', '23333444', 1, 'Perez');
insert into consultas values('2006-08-10', '9:00', '24333444', 2, 'Perez');
insert into consultas values('2006-08-10', '10:00', '25333444', 3, 'Perez');
insert into consultas values('2006-08-10', '8:30', '25333444', 1, 'Garcia');
insert into consultas values('2006-09-10', '8:30', '25333444', 1, 'Lopez');
```

4- Muestre la fecha, hora, documento del paciente, médico, nombre y monto de la obra social de todas las consultas (join con "obrassociales"):

```
select c.fecha, c.hora, c.documento, c.medico, os.nombre, os.monto
from consultas as c
join obrassociales as os
on os.codigo=c.codigoobrasocial;
```

5- Muestre fecha, hora, documento del paciente y nombre de la obra social para las consultas del doctor "Perez":

```
select c.fecha, c.hora, c.documento, os.nombre, os.monto
from consultas as c
join obrassociales as os
on os.codigo=c.codigoobrasocial
where c.medico='Perez';
```

6- Muestre las obras sociales DISTINTAS que atendió el doctor "Perez" el día "2006-08-10":

```
select distinct os.nombre
from consultas as c
join obrassociales as os
on os.codigo=c.codigoobrasocial
where c.fecha='2006-08-10' and
medico='Perez';
```

[Retornar](#)