

## 70 - join con más de dos tablas.

### Problema:

Un video club que alquila películas en video guarda información de sus películas en alquiler, sus socios y los alquileres en 3 tablas llamadas "peliculas", "socios" y "alquileres" respectivamente.

1- Elimine las tablas si existen.

2- Créelas con las siguientes estructuras:

```
create table peliculas (  
  codigo smallint unsigned auto_increment,  
  titulo varchar(40) not null,  
  actores varchar(40),  
  duracion tinyint unsigned,  
  primary key (codigo)  
);
```

```
create table socios(  
  codigo smallint unsigned auto_increment,  
  documento char(8),  
  nombre varchar(30),  
  domicilio varchar(30),  
  primary key (codigo)  
);
```

```
create table alquileres(  
  codigopelicula smallint unsigned not null,  
  codigosocio smallint unsigned not null,  
  fechaprestamo date not null,  
  fechadevolucion date,  
  primary key (codigopelicula,fechaprestamo)  
);
```

3- Ingrese los siguientes registros para las 3 tablas.

```
insert into peliculas (titulo,actores,duracion)  
  values('Elsa y Fred','China Zorrilla',90);  
insert into peliculas (titulo,actores,duracion)  
  values('Mision imposible','Tom Cruise',120);  
insert into peliculas (titulo,actores,duracion)  
  values('Mision imposible 2','Tom Cruise',180);  
insert into peliculas (titulo,actores,duracion)  
  values('Harry Potter y la piedra filosofal','Daniel H.',120);  
insert into peliculas (titulo,actores,duracion)  
  values('Harry Potter y la camara secreta','Daniel H.',150);
```

```
insert into socios (documento,nombre)  
  values('22333444','Juan Lopez');  
insert into socios (documento,nombre)  
  values('23333444','Diana Perez');  
insert into socios (documento,nombre)  
  values('24333444','Luis Fuentes');
```

```
insert into alquileres (codigopelicula,codigosocio,fechaprestamo)  
  values(1,1,'2016-07-02');  
insert into alquileres (codigopelicula,codigosocio,fechaprestamo)  
  values(2,1,'2016-07-02');  
insert into alquileres (codigopelicula,codigosocio,fechaprestamo)  
  values(3,1,'2016-07-12');
```

```

insert into alquileres (codigopelicula,codigosocio,fechaprestamo)
values(1,2,'2016-08-02');
insert into alquileres (codigopelicula,codigosocio,fechaprestamo)
values(3,2,'2016-08-12');
insert into alquileres (codigopelicula,codigosocio,fechaprestamo)
values(4,2,'2016-08-02');
insert into alquileres (codigopelicula,codigosocio,fechaprestamo)
values(1,3,'2016-09-02');
insert into alquileres (codigopelicula,codigosocio,fechaprestamo)
values(2,3,'2016-08-02');
insert into alquileres (codigopelicula,codigosocio,fechaprestamo)
values(3,3,'2016-08-15');
insert into alquileres (codigopelicula,codigosocio,fechaprestamo)
values(4,3,'2016-08-22');
insert into alquileres (codigopelicula,codigosocio,fechaprestamo)
values(4,1,'2016-08-25');
insert into alquileres (codigopelicula,codigosocio,fechaprestamo)
values(1,3,'2016-08-25');

```

4- Muestre toda la información de los "alquileres" (nombre de la película, nombre del socio, fecha de préstamo y de devolución):

```

select titulo,nombre,fechaprestamo,fechadevolucion from alquileres as a
join peliculas as p
on a.codigopelicula=p.codigo
join socios as s
on s.codigo=a.codigosocio;

```

5- Muestre la cantidad de veces que se alquiló cada película:

```

select p.titulo,count(*) from peliculas as p
join alquileres as a
on p.codigo=a.codigopelicula
group by p.titulo;

```

6- Muestre la cantidad de películas que alquiló cada socio:

```

select s.nombre,count(a.codigopelicula) from socios as s
join alquileres as a
on s.codigo=a.codigosocio
group by s.nombre;

```

7- Muestre la cantidad de películas DISTINTAS que alquiló cada socio:

```

select s.nombre,count(distinct a.codigopelicula) from socios as s
join alquileres as a
on s.codigo=a.codigosocio
group by s.nombre;

```

8- Muestre la cantidad de películas alquiladas por mes por cada socio ordenado por mes:

```

select s.nombre,
monthname(a.fechaprestamo) as mes,
count(a.codigopelicula)
from socios as s
join alquileres as a
on s.codigo=a.codigosocio
group by s.nombre, mes
order by mes;

```

Ejecutar los comandos SQL

Borrar comando.

Finalizar

### Otros problemas:

Un club de dicta clases de distintos deportes a sus socios. Guarda la información de sus socios en una tabla llamada "socios", los datos de los deportes en "deportes" y las inscripciones en "inscriptos".

1- Elimine las 3 tablas, si existen.

2- Cree las tablas:

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

```
create table deportes(  
  codigo tinyint unsigned auto_increment,  
  nombre varchar(30),  
  primary key(codigo)  
);
```

```
create table inscriptos(  
  documento char(8) not null,  
  codigodeporte tinyint unsigned,  
  año year not null,  
  cuota char(1), /*'s' o 'n', si esta paga o no*/  
  primary key(documento,codigodeporte,año)  
);
```

3- Ingrese los siguientes registros:

```
insert into socios values('22333444','Juan Perez');  
insert into socios values('23333444','Ana Garcia');  
insert into socios values('24333444','Hector Fuentes');  
insert into socios values('25333444','Marta Molina');
```

```
insert into deportes (nombre) values('tenis');  
insert into deportes (nombre) values('natacion');  
insert into deportes (nombre) values('basquet');  
insert into deportes (nombre) values('voley');
```

```
insert into inscriptos values('22333444',1,'2015','s');  
insert into inscriptos values('22333444',1,'2016','s');  
insert into inscriptos values('22333444',2,'2015','s');
```

```
insert into inscriptos values('24333444',1,'2015','s');
insert into inscriptos values('24333444',2,'2016','s');
insert into inscriptos values('25333444',1,'2015','s');
insert into inscriptos values('25333444',1,'2016','s');
insert into inscriptos values('25333444',3,'2016','s');
```

4- Muestre el nombre del socio, el deporte en el cual se ha inscripto y el año de inscripción usando "join":

```
select s.nombre,d.nombre,i.año
from inscriptos as i
join socios as s
on s.documento=i.documento
join deportes as d
on d.codigo=i.codigodeporte;
```

5- Muestre los nombres de todos los socios y el nombre de los deportes en los cuales se han inscripto, incluso, si no se ha inscripto en ninguno:

```
select s.nombre,d.nombre
from socios as s
left join inscriptos as i
on s.documento=i.documento
left join deportes as d
on d.codigo=i.codigodeporte;
```

6- Muestre todos los deportes y los nombres de los socios inscriptos, incluso para aquellos que no tienen socios inscriptos:

```
select d.nombre,s.nombre
from deportes as d
left join inscriptos as i
on d.codigo=i.codigodeporte
left join socios as s
on s.documento=i.documento;
```

7- Muestre la cantidad de socios inscriptos en cada deporte:

```
select d.nombre,count(i.codigodeporte)
from deportes as d
left join inscriptos as i
on d.codigo=i.codigodeporte
left join socios as s
on s.documento=i.documento
group by d.nombre;
```

8- Muestre los distintos socios que se inscribieron en el año "2016":

```
select distinct s.nombre
from inscriptos as i
join socios as s
on s.documento=i.documento
where año='2016';
```

B) Un instituto de enseñanza guarda en una tabla llamada "carreras" los datos de las carreras que dicta, en "materias" las materias de cada carrera y en "inscriptos" las inscripciones.

1- Elimine las 3 tablas, si existen:

```
drop table if exists carreras, materias, inscriptos;
```

2- Cree las tablas con las siguientes estructuras:

```
create table carreras(
codigo tinyint unsigned auto_increment,
nombre varchar(30),
primary key(codigo)
```

```
);
```

```
create table materias(  
codigo tinyint unsigned auto_increment,  
codigocarrera tinyint unsigned,  
nombre varchar(30),  
profesor varchar(30),  
primary key(codigo,codigocarrera)  
);
```

```
create table inscriptos(  
documento char(8) not null,  
codigocarrera tinyint unsigned,  
codigomateria tinyint unsigned,  
año year,  
cuota char(1),/* si esta paga o no*/  
primary key (documento,codigocarrera,codigomateria,año)  
);
```

3- Ingrese algunos registros:

```
insert into carreras values(1,'Analista de sistemas');  
insert into carreras values(2,'Diseñador web');
```

```
insert into materias values(1,1,'Programacion I','Alfredo Lopez');  
insert into materias values(2,1,'Sistemas de datos I','Bernardo Garcia');  
insert into materias values(3,1,'Ingles tecnico','Edit Torres');  
insert into materias values(1,2,'Programacion basica','Alfredo Lopez');  
insert into materias values(2,2,'Ingles I','Edit Torres');  
insert into materias values(3,2,'Protocolos','Hector Juarez');
```

```
insert into inscriptos values('22333444',1,1,'2015','s');  
insert into inscriptos values('22333444',1,2,'2015','s');  
insert into inscriptos values('22333444',1,3,'2016','n');  
insert into inscriptos values('23222222',1,1,'2015','s');  
insert into inscriptos values('23222222',1,2,'2016','s');  
insert into inscriptos values('24555666',1,1,'2015','s');  
insert into inscriptos values('24555666',2,1,'2015','s');  
insert into inscriptos values('25000999',1,1,'2015','s');  
insert into inscriptos values('25000999',1,2,'2015','s');  
insert into inscriptos values('25000999',2,1,'2016','n');  
insert into inscriptos values('25000999',2,2,'2016','s');
```

4- Muestre el nombre de las materias, a qué carrera pertenecen y el nombre del profesor que las dicta ordenadas por carrera:

```
select c.nombre, m.nombre,m.profesor  
from materias as m  
join carreras as c  
on c.codigo=m.codigocarrera  
order by c.nombre;
```

5- Muestre el nombre de los profesores y la cantidad de materias que dicta cada uno:

```
select m.profesor,count(*) as cantidad  
from materias as m  
group by m.profesor;
```

6- Muestre todos los datos de la tabla "inscriptos" (sin códigos) incluyendo los nombres de las materias y carreras ordenado por nombre de carrera y nombre de materia:

```
select i.documento,c.nombre,m.nombre,año,cuota  
from inscriptos as i  
join carreras as c  
on c.codigo=i.codigocarrera
```

```
join materias as m
on m.codigo=i.codigomateria and
m.codigocarrera=c.codigo
order by c.nombre,c.nombre;
```

Note que unimos "inscriptos" con "carreras" por el código de la carrera, "inscriptos" con "materias" por el código de la materia y "carreras" con "materias" por el código de la carrera; si olvidamos el último enlace, se combinarán todos los códigos de carreras con todos los códigos de materias.

7- Muestre la cantidad de alumnos que tiene cada profesor (hay profesores que dictan varias materias en distintas carreras):

```
select m.profesor,count(*)
from inscriptos as i
join carreras as c
on c.codigo=i.codigocarrera
join materias as m
on m.codigo=i.codigomateria and
m.codigocarrera=c.codigo
group by m.profesor;
```

8- Muestre la cantidad de alumnos inscriptos en cada materia de cada carrera:

```
select c.nombre,m.nombre,count(i.codigomateria)
from carreras as c
join materias as m
on c.codigo=m.codigocarrera
left join inscriptos as i
on m.codigo=i.codigomateria and
c.codigo=i.codigocarrera
group by c.nombre,m.nombre;
```

Note que usamos "left join" para mostrar todas las materias, incluso para aquellas en las cuales no hay inscriptos.

9- Muestre el documento de los alumnos y la cantidad de materias por carrera en las que se ha inscripto cada uno de ellos:

```
select i.documento,c.nombre,
count(i.codigomateria) as materias
from carreras as c
join materias as m
on c.codigo=m.codigocarrera
join inscriptos as i
on m.codigo=i.codigomateria and
c.codigo=i.codigocarrera
group by i.documento,c.nombre;
```

10- Muestre la cantidad de alumnos distintos inscriptos en la institución:

```
select count(distinct documento) from inscriptos;
```

11- Muestre la cantidad de alumnos que no pagaron la cuota, por carrera y materia:

```
select c.nombre,m.nombre,count(*)
from inscriptos as i
join carreras as c
on c.codigo=i.codigocarrera
join materias as m
on m.codigo=i.codigomateria and
m.codigocarrera=c.codigo
where i.cuota='n'
group by c.nombre,m.nombre;
```

