

76 - Insertar registros con valores de otra tabla (insert - select)

Problema:

Un comercio guarda la información de sus ventas en una tabla llamada "facturas".

1- Elimine la tabla si existe.

2- Cree la tabla con la siguiente estructura:

```
create table facturas(  
  numero int(10) zerofill auto_increment,  
  numeroitem smallint unsigned not null,  
  descripcion varchar(30),  
  precioporunidad decimal(5,2) unsigned,  
  cantidad tinyint unsigned,  
  primary key (numero,numeroitem)  
);
```

3- Ingrese algunos registros:

```
insert into facturas (numero,numeroitem,descripcion,precioporunidad,cantidad)  
values(100,1,'escuadra 20 cm.',2.50,20);  
insert into facturas (numero,numeroitem,descripcion,precioporunidad,cantidad)  
values(100,2,'escuadra 50 cm.',5,30);  
insert into facturas (numero,numeroitem,descripcion,precioporunidad,cantidad)  
values(100,3,'goma lapiz-tinta',0.35,100);  
insert into facturas (numero,numeroitem,descripcion,precioporunidad,cantidad)  
values(102,1,'lapices coloresx6',4.40,50);  
insert into facturas (numero,numeroitem,descripcion,precioporunidad,cantidad)  
values(102,2,'lapices coloresx12',8,60);  
insert into facturas (numero,numeroitem,descripcion,precioporunidad,cantidad)  
values(255,1,'lapices coloresx24',12.35,100);  
insert into facturas (numero,numeroitem,descripcion,precioporunidad,cantidad)  
values(567,1,'compas plastico',12,50);  
insert into facturas (numero,numeroitem,descripcion,precioporunidad,cantidad)  
values(567,2,'compas metal',18.90,80);
```

4- Elimine la tabla "montofacturas" si existe.

5- Se quiere guardar en una tabla denominada "montofacturas" el número de factura y el monto total en dinero de todos los items de cada factura (precioporunidad x cantidad). Cree la tabla con la siguiente estructura:

```
create table montofacturas(  
  numero int(10) zerofill,  
  total decimal(9,2) unsigned,  
  primary key(numero)  
);
```

6- Realice una consulta de la tabla "facturas", agrupando por "numero", en la cual aparezca el número de factura y el monto total de todos sus items:

```
select numero,  
  sum(precioporunidad*cantidad) as total  
from facturas  
group by numero;
```

7- Ingrese el resultado de la consulta anterior en la tabla "montofacturas":

```
insert into montofacturas (numero,total)  
select numero,  
  sum(precioporunidad*cantidad) as total  
from facturas
```

group by numero;

8- Muestre todos los registros de la tabla "montofacturas".

Ejecutar los comandos SQL

Borrar comando.

Finalizar

Otros problemas:

A) Un club se dicta clases de distintos deportes a sus socios. Guarda la información de las inscripciones en una tabla denominada "inscriptos".

1- Elimine la tabla, si existe.

2- Cree la tabla:

```
create table inscriptos(
  documento char(8) not null,
  deporte varchar(20) not null,
  año year not null,
  cuota char(1), /*'s' o 'n', si esta paga o no*/
  primary key(documento,deporte,año)
);
```

3- Ingrese los siguientes registros:

```
insert into inscriptos values('22333444','tenis','2015','s');
insert into inscriptos values('22333444','tenis','2016','s');
insert into inscriptos values('22333444','natacion','2015','s');
insert into inscriptos values('24333444','tenis','2015','s');
insert into inscriptos values('24333444','natacion','2016','s');
insert into inscriptos values('25333444','tenis','2015','s');
insert into inscriptos values('25333444','tenis','2016','s');
insert into inscriptos values('25333444','basquet','2016','s');
```

4- El club desea saber cuántas personas se han inscripto en cada deporte cada año:

```
select deporte,año,count(*)
from inscriptos
group by deporte,año;
```

5- El club quiere almacenar esa información en una tabla. Elimine la tabla "inscriptospordeporteporaño", si existe.

6- Cree la tabla:

```
create table inscriptospordeporteporaño(
  deporte varchar(20),
  año year,
  cantidad smallint unsigned
```

);

7- Intente ingresar los registros colocando un campo menos en la lista de campos:

```
insert into inscriptospordeporteporaño (deporte,cantidad)
```

```
select deporte,año,count(*)
```

```
from inscriptos
```

```
group by deporte,año;
```

La sentencia no se ejecuta.

8- Intente ingresar los registros colocando los 3 campos de la nueva tabla y quitándole una columna a la selección:

```
insert into inscriptospordeporteporaño (deporte,año,cantidad)
```

```
select deporte,año
```

```
from inscriptos
```

```
group by deporte,año;
```

La sentencia no se ejecuta.

9- Ingrese los registros a partir de la consulta del punto 4, correctamente:

```
insert into inscriptospordeporteporaño (deporte,año,cantidad)
```

```
select deporte,año,count(*)
```

```
from inscriptos
```

```
group by deporte,año;
```

10- Muestre todos los registros de la nueva tabla.

11- El club desea saber cuántas veces se ha inscripto una persona en algún deporte:

```
select documento,count(*)
```

```
from inscriptos
```

```
group by documento;
```

12- Elimine la tabla "inscripcionesporpersona" si existe.

13- Cree la tabla con la siguiente estructura:

```
create table inscripcionesporpersona(
```

```
documento char(8),
```

```
cantidad tinyint unsigned,
```

```
primary key(documento)
```

```
);
```

14- Guarde la información del punto 11 en la tabla recientemente creada:

```
insert into inscripcionesporpersona (documento,cantidad)
```

```
select documento,count(*)
```

```
from inscriptos
```

```
group by documento;
```

15- Vea los registros de la nueva tabla.

B) Un profesor guarda en una tabla llamada "notas" las notas de los alumnos de un curso.

1- Elimine la tabla "notas" si existe.

2- Cree la tabla con la siguiente estructura:

```
create table notas(
```

```
documento char(8) not null,
```

```
nombre varchar(30),
```

```
nota decimal(4,2) unsigned
```

```
);
```

Algunos alumnos tienen más notas que otros porque presentaron trabajos especiales.

3- Ingrese los siguientes registros:

```

insert into notas values('30333444','Juan Perez',8);
insert into notas values('30433444','Ana Maria Garcia',7.5);
insert into notas values('30533444','Marcelo Morales',3.5);
insert into notas values('30633444','Fabiola Juarez',4.5);
insert into notas values('30333444','Juan Perez',7.8);
insert into notas values('30433444','Ana Maria Garcia',6);
insert into notas values('30533444','Marcelo Morales',4.2);
insert into notas values('30633444','Fabiola Juarez',7);
insert into notas values('30333444','Juan Perez',9);
insert into notas values('30433444','Ana Maria Garcia',5.8);
insert into notas values('30633444','Fabiola Juarez',2);
insert into notas values('30333444','Juan Perez',10);
insert into notas values('30633444','Fabiola Juarez',6.7);

```

4- El profesor necesita una tabla llamada "promedios" que contenga el documento y nombre del alumno y el promedio de sus notas. Elimine la tabla "promedios" si existe.

5- Cree la tabla con los campos "documento" y "promedio":

```

create table promedios (
  documento char(8) not null,
  nombre varchar(30),
  promedio decimal(4,2),
  primary key(documento)
);

```

6- Realice la consulta necesaria a la tabla "notas":

```

select documento,nombre,avg(nota) as promedio
from notas
group by documento;

```

7- Ingrese el resultado de la consulta anterior en la tabla "promedios":

```

insert into promedios (documento,nombre,promedio)
select documento,nombre,avg(nota)
from notas
group by documento;

```

8- Vea los registros de "promedios".

9- Agrupe por documento y seleccione el documento y nombre de los alumnos aprobados (promedio mayor o igual a 4):

```

select documento,nombre
from notas
group by documento
having avg(nota)>=4;

```

10- Elimine la tabla "aprobados" si existe.

11- Cree una tabla llamada "aprobados" con la siguiente estructura:

```

create table aprobados(
  numero tinyint unsigned auto_increment,
  documento char(8),
  nombre varchar(30),
  primary key (numero)
);

```

12- Ingrese los registros resultado de la consulta del punto 9 en la tabla creada anteriormente:

```

insert into aprobados (documento,nombre)
select documento,nombre
from notas
group by documento
having avg(nota)>=4;

```

Note que en la lista de campos a cargar no se incluye el campo "numero".

13- Vea los registros recientemente cargados. Note que en el campo "numero" se cargó automáticamente.

14- Muestre el documento y nombre del alumno y en una columna informe si el alumno está aprobado (promedio de notas mayor o igual a 4) o desaprobado (promedio menor a 4), empleando un "if" en una consulta a la tabla "notas" :

```
select documento,nombre,  
if (avg(nota)>4,'aprobado','desaprobado') as condicion  
from notas  
group by documento;
```

15- Elimine la tabla "condicion" si existe.

16- Cree la tabla "condicion" con la siguiente estructura:

```
create table condicion(  
documento char(8) not null,  
nombre varchar(30),  
condicion varchar(15)  
);
```

17- Ingrese registros en la tabla creada anteriormente a partir de la consulta del punto 14:

```
insert into condicion (documento,nombre,condicion)  
select documento,nombre,  
if (avg(nota)>4,'aprobado','desaprobado')  
from notas  
group by documento;
```

18- Vea todos los registros de la tabla "condicion".

C) Un banco tiene registrados las cuentas corrientes de sus clientes en una tabla llamada "cuentas".

1- Elimine la tabla, si existe.

2- Cree la tabla:

```
create table cuentas(  
numero int(8) zerofill auto_increment,  
documento char(8) not null,  
nombre varchar(30),  
saldo decimal(9,2),  
primary key(numero)  
);
```

3- Ingrese los siguientes registros:

```
insert into cuentas values(1234,'22333444','Juan Perez',2000.60);  
insert into cuentas values(2566,'23333444','Maria Pereyra',5050);  
insert into cuentas values(5987,'24333444','Marcos Torres',200);  
insert into cuentas values(14434,'25333444','Ana Juarez',8000.60);  
insert into cuentas values(28566,'23333444','Maria Pereyra',8050);  
insert into cuentas values(35987,'24333444','Marcos Torres',2000);
```

4- Agrupe por documento y muestre el documento del cliente, en una columna llamada "cantidad" calcule la cantidad de cuentas de cada cliente y en otra columna llamada "total" calcule la suma de todas las cuentas de cada cliente:

```
select documento,  
count(*) as cantidad,  
sum(saldo) as total  
from cuentas  
group by documento;
```

5- Elimine la tabla "cuentasporcliente", si existe.

6- Cree la tabla "cuentasporcliente" con la siguiente estructura:

```
create table cuentasporcliente(  
  numero int unsigned auto_increment,  
  documento char(8),  
  cantidad tinyint unsigned,  
  total decimal(9,2) unsigned,  
  primary key(numero)  
);
```

7- Inserte el resultado de la consulta del punto 4 en la tabla creada anteriormente:

```
insert into cuentasporcliente (documento,cantidad,total)  
select documento,count(*),sum(saldo)  
from cuentas  
group by documento;
```

8- Vea los registros de "cuentasporcliente".

D) Una agencia matrimonial almacena la información de sus clientes en una tabla llamada "clientes".

1- Elimine la tabla si existe:

2- Cree la tabla:

```
create table clientes(  
  nombre varchar(30),  
  domicilio varchar(30),  
  sexo char(1),  
  edad int  
);
```

3- Ingrese los siguientes registros:

```
insert into clientes (nombre,sexo,edad) values('Juan Perez','m',45);  
insert into clientes (nombre,sexo,edad) values('Ana Lopez','f',50);  
insert into clientes (nombre,sexo,edad) values('Federico Herrero','m',30);  
insert into clientes (nombre,sexo,edad) values('Mariano Juarez','m',35);  
insert into clientes (nombre,sexo,edad) values('Maria Torres','f',36);  
insert into clientes (nombre,sexo,edad) values('Ines Duarte','f',55);  
insert into clientes (nombre,sexo,edad) values('Alejandra Figueroa','f',40);
```

4- La agencia necesita la combinación de todas las personas de sexo femenino con las de sexo masculino. Además, las edades de las posibles parejas no deben tener una diferencia superior a 10 años. Use "cross join":

```
select f.nombre,f.edad,  
  m.nombre,m.edad  
from clientes as f  
cross join clientes as m  
where f.sexo='f' and m.sexo='m' and  
  m.edad - f.edad between -10 and 10;
```

5- La agencia quiere organizar encuentros entre dichas parejas. Elimine la tabla "encuentros" si existe.

6- Cree la tabla "encuentros" con la siguiente estructura:

```
create table encuentros(  
  codigo tinyint unsigned auto_increment,  
  dama varchar(30),  
  edaddama int,  
  caballero varchar(30),  
  edadcaballero tinyint unsigned,
```

```
fecha date,  
primary key (codigo)  
);
```

7- Inserte las parejas resultado de la consulta del punto 4:

```
insert into encuentros (dama,edaddama,caballero,edadcaballero)  
select f.nombre,f.edad,m.nombre,m.edad  
from clientes as f  
cross join clientes as m  
where f.sexo='f' and m.sexo='m' and  
m.edad - f.edad between -10 and 10;
```

8- Vea la información que se almacenó en "encuentros":

```
select * from encuentros;
```

Note que el código se cargó automáticamente y en las fechas de encuentro contienen todas el valor "null" para luego ser llenadas.

E) Varios clubes de barrio se organizaron para realizar campeonatos entre ellos. La tabla llamada "equipos" guarda la informacion de los distintos equipos que jugarán.

1- Elimine la tabla, si existe.

2- Cree la tabla:

```
create table equipos(  
codigo tinyint unsigned auto_increment,  
nombre varchar(30),  
domicilio varchar(30),  
barrio varchar(20),  
primary key(codigo)  
);
```

3- Ingrese los siguientes registros:

```
insert into equipos values(1,'Los tigres','Sarmiento 234','Gral. Paz');  
insert into equipos values(2,'Los leones','Colon 123','Centro');  
insert into equipos values(3,'Campeones','Guemes 346','Pueyrredon');  
insert into equipos values(4,'Cebollitas','Colon 1234','Alberdi');
```

4- Realice un "cross join" combinando los distintos equipos y sedes de encuentro:

```
select e1.nombre,e2.nombre,e1.barrio as 'sede'  
from equipos as e1  
cross join equipos as e2  
where e1.nombre<>e2.nombre;
```

5- Se necesita almacenar en una tabla los datos de la consulta anterior y una fecha de encuentro. Elimine la tabla "encuentros" si existe.

6- Cree la tabla "encuentros" con la siguiente estructura:

```
create table encuentros(  
codigo tinyint unsigned auto_increment,  
equipo1 varchar(30),  
equipo2 varchar(30),  
lugarencuentro varchar(30),  
barrioencuentro varchar(20),  
fecha date,  
primary key(codigo)  
);
```

7- Ingrese los datos de la consulta del punto 4 en la tabla recientemente creada:

```
insert into encuentros(equipo1,equipo2,lugarencuentro,barrioencuentro)  
select e1.nombre,e2.nombre,e1.domicilio,e1.barrio
```

```
from equipos as e1  
cross join equipos as e2  
where e1.nombre<>e2.nombre;
```

8- Vea todos los registros de la tabla creada anteriormente.

Note que el código se cargó automáticamente y las fechas de encuentro contienen "null" para luego ser llenadas.

[Retornar](#)