

## Practica

Se va a pedir el código SQL

Fecha de entrega 16-06-22 a las 18hs

Funciones de agrupamiento (count - max - min - sum - avg)

1-A

Problema:

Un comercio que tiene un stand en una feria registra en una tabla llamada "visitantes" algunos datos de las personas que visitan o compran en su stand para luego enviarle publicidad de sus productos.

1- Elimine la tabla "visitantes", si existe.

2- Créela con la siguiente estructura:

```
create table visitantes(  
  nombre varchar(30),  
  edad tinyint unsigned,  
  sexo char(1),  
  domicilio varchar(30),  
  ciudad varchar(20),  
  telefono varchar(11),  
  montocompra decimal (6,2) unsigned  
);
```

Note que no tiene clave primaria, no la necesita.

3- Ingrese algunos registros:

```
insert into visitantes (nombre,edad, sexo,domicilio,ciudad,telefono,montocompra)  
values ('Susana Molina', 28,'f','Colon 123','Cordoba',null,45.50);  
  
insert into visitantes (nombre,edad, sexo,domicilio,ciudad,telefono,montocompra)
```

```

values ('Marcela Mercado',36,'f','Avellaneda 345','Cordoba','4545454',0);
insert into visitantes (nombre,edad, sexo,domicilio,ciudad,telefono,montocompra)
values ('Alberto Garcia',35,'m','Gral. Paz 123','Alta Gracia','03547123456',25);
insert into visitantes (nombre,edad, sexo,domicilio,ciudad,telefono,montocompra)
values ('Teresa Garcia',33,'f','Gral. Paz 123','Alta Gracia','03547123456',0);
insert into visitantes (nombre,edad, sexo,domicilio,ciudad,telefono,montocompra)
values ('Roberto Perez',45,'m','Urquiza 335','Cordoba','4123456',33.20);
insert into visitantes (nombre,edad, sexo,domicilio,ciudad,telefono,montocompra)
values ('Marina Torres',22,'f','Colon 222','Villa Dolores','03544112233',25);
insert into visitantes (nombre,edad, sexo,domicilio,ciudad,telefono,montocompra)
values ('Julieta Gomez',24,'f','San Martin 333','Alta Gracia','03547121212',53.50);
insert into visitantes (nombre,edad, sexo,domicilio,ciudad,telefono,montocompra)
values ('Roxana Lopez',20,'f','Triunvirato 345','Alta Gracia',null,0);
insert into visitantes (nombre,edad, sexo,domicilio,ciudad,telefono,montocompra)
values ('Liliana Garcia',50,'f','Paso 999','Cordoba','4588778',48);
insert into visitantes (nombre,edad, sexo,domicilio,ciudad,telefono,montocompra)
values ('Juan Torres',43,'m','Sarmiento 876','Cordoba','4988778',15.30);

```

4- Solicite la cantidad de visitantes al stand (10 registros):

```
select count(*) from visitantes;
```

5- Muestre la suma de la compra de todos los visitantes de "Alta Gracia" (78.5):

```
select sum(montocompra) from visitantes
where ciudad='Alta Gracia';
```

6- Muestre el valor máximo de las compras efectuadas (53.50):

```
select max(montocompra) from visitantes;
```

7- Muestre la edad menor de los visitantes (20):

```
select min(edad) from visitantes;
```

8- Muestre el promedio de edades de los visitantes (33.66):

```
select avg(edad) from visitantes;
```

9- Muestre el promedio del monto de compra (24.55):

```
select avg(montocompra) from visitantes;
```

## 2-B Ejercicio

A) Una academia de informática dicta distintos cursos y almacena en una tabla

llamada "inscripciones" la siguiente información: nombre del curso, documento del alumno, fecha en

que se inscribe el alumno, monto del pago (algunos dejan una seña, otros pagan el curso completo).

1- Elimine la tabla si existe.

2- cree la tabla:

```
create table inscripciones(  
    nombre varchar(30),  
    documento char(8),  
    fechainscripto date,  
    pago decimal(5,2) unsigned not null  
);
```

3- Ingrese algunos registros:

```
insert into inscripciones values('PHP básico', '22333444','2006-08-10',50);  
insert into inscripciones values('PHP básico', '23333444','2006-08-10',50);  
insert into inscripciones values('PHP básico', '24333444','2006-08-11',30);  
insert into inscripciones values('PHP experto', '25333444','2006-08-11',0);  
insert into inscripciones values('PHP experto', '26333444','2006-08-12',200);  
insert into inscripciones values('JavaScript básico', '22333444','2006-08-10',100);  
insert into inscripciones values('Operador de PC', '27333444','2006-08-12',10);
```

```
insert into inscripciones values('Operador de PC', '28333444','2006-08-13',50);
insert into inscripciones values('Operador de PC', '29333444','2006-08-14',40);
insert into inscripciones values('Operador de PC', '30333444','2006-08-14',0);
insert into inscripciones values('Diseño web', '29333444','2006-08-14',200);
insert into inscripciones values('Diseño web', '30333444','2006-08-14',0);
```

4- calcule la cantidad de inscriptos para el curso de "Operador de PC":

```
select count(*)
from inscripciones
where nombre='Operador de PC';
```

5- Calcule la suma recaudada por los pagos de los cursos el día "2006-08-10":

```
select sum(pago)
from inscripciones
where fechainscripto='2006-08-10';
```

6- Calcule el promedio de los pagos de los inscriptos:

```
select avg(pago)
from inscripciones;
```

7- Muestre el máximo y el mínimo valor de pago, sin considerar quienes no pagan:

```
select max(pago),min(pago)
from inscripciones
where pago>0;
```

8- Vea en cuántos cursos se inscribió el alumno con documento "22333444" y cuánto abonó en total:

```
select count(*) as 'cantidad',
sum(pago) as 'Abono'
from inscripciones
where documento='22333444';
```

B) Trabaje con la tabla "peliculas" de un video club.

1- Elimine la tabla, si existe.

2- Créela con la siguiente estructura:

- codigo (entero sin signo, autoincrementable),
- titulo (cadena de 30), not null,
- actor (cadena de 20),
- duracion (entero sin signo no mayor a 200 aprox.),
- clave primaria (codigo).

3- Ingrese los siguientes registros:

```
insert into peliculas (titulo,actor,duracion)
values('Mision imposible','Tom Cruise',120);
insert into peliculas (titulo,actor,duracion)
values('Harry Potter y la piedra filosofal','Daniel R.',180);
insert into peliculas (titulo,actor,duracion)
values('Harry Potter y la camara secreta','Daniel R.',190);
insert into peliculas (titulo,actor,duracion)
values('Mision imposible 2','Tom Cruise',120);
insert into peliculas (titulo,actor,duracion)
values('Mujer bonita','Richard Gere',120);
insert into peliculas (titulo,actor,duracion)
values('Tootsie','D. Hoffman',90);
insert into peliculas (titulo,actor,duracion)
values('Un oso rojo',null,100);
insert into peliculas (titulo,actor,duracion)
values('Elsa y Fred','China Zorrilla',110);
insert into peliculas (titulo,actor,duracion)
```

```
values('Mrs. Johns','Richard Gere',180);
```

4- Muestre el valor de duración más grande:

```
select max(duracion)from peliculas;
```

5- Muestre el promedio de duración de las películas:

```
select avg(duracion) from peliculas;
```

6- Cuente la cantidad de películas que comiencen con la cadena "Harry Potter":

```
select count(*)  
from peliculas  
where titulo like 'Harry Potter%';
```

7- Un socio alquiló todas las películas en las cuales trabaja "Richard Gere", quiere saber el total de minutos que duran todas sus películas:

```
select sum(duracion)  
from peliculas  
where actor='Richard Gere';
```

C) Una concesionaria de autos vende autos usados y almacena la información en una tabla llamada "autos".

1- Elimine la tabla "autos" si existe.

2- Cree la tabla con la siguiente estructura:

```
create table autos(  
    patente char(6),  
    marca varchar(20),  
    modelo char(4),
```

```
precio decimal(8,2) unsigned,  
primary key(patente)  
);
```

3- Ingrese los siguientes registros:

```
insert into autos  
values('ACD123','Fiat 128','1970',15000);  
insert into autos  
values('ACG234','Renault 11','1990',40000);  
insert into autos  
values('BCD333','Peugeot 505','1990',80000);  
insert into autos  
values('GCD123','Renault Clio','1990',70000);  
insert into autos  
values('BCC333','Renault Megane','1998',95000);  
insert into autos  
values('BVF543','Fiat 128','1975',20000);
```

4- Muestre el valor del auto más caro y más barato:

```
select max(precio), min(precio)  
from autos;
```

5- Muestre el valor de auto más caro de 1990:

```
select max(precio)  
from autos  
where modelo='1990';
```

6- Muestre el promedio de los precios de los autos "Fiat 128":

```
select avg(precio)  
from autos  
where marca='Fiat 128';
```

7- Calcule el valor en dinero de todos los autos marca "Renault" con modelos menores a "1995":

```
select sum(precio)
from autos
where marca like '%Renault%' and
modelo<1995;
```

D) Un comercio guarda la información de sus ventas en una tabla llamada "facturas" en la que registra el número de factura, la descripción de los items comprados, el precio por unidad de los items y la cantidad.

1- Elimine la tabla si existe.

2- Cree la tabla:

```
create table facturas(
numero int(10) zerofill,
descripcion varchar(30),
precioporunidad decimal(5,2) unsigned,
cantidad tinyint unsigned
);
```

3- Ingrese algunos registros:

```
insert into facturas values(504,'escuadra 20 cm.',2.5,100);
insert into facturas values(504,'escuadra 50 cm.',5,80);
insert into facturas values(2002,'compas plastico',8,120);
insert into facturas values(2002,'compas metal',15.4,100);
insert into facturas values(2002,'escuadra 20 cm.',2.5,100);
insert into facturas values(4567,'escuadra 50 cm.',5,200);
```



4- Cuente la cantidad de items de la factura número "2002":

```
select count(*)  
from facturas  
where numero='2002';
```

5- Sume la cantidad de productos de la factura número "2002":

```
select sum(cantidad)  
from facturas  
where numero='2002';
```

6- Muestre el total en dinero de la factura "504":

```
select sum(cantidad*precioporunidad)  
from facturas  
where numero='504';
```

3-A

Operadores Lógicos (and - or - not)

Problema:

Trabaje con la tabla llamada "medicamentos" de una farmacia.

1- Elimine la tabla, si existe.

2- Cree la tabla con la siguiente estructura:

```
create table medicamentos(  
codigo int unsigned auto_increment,  
nombre varchar(20),  
laboratorio varchar(20),  
precio decimal(5,2),
```

```
cantidad int unsigned,  
primary key(codigo)  
);
```

3- Visualice la estructura de la tabla "medicamentos".

4- Ingrese los siguientes registros (insert into):

```
insert into medicamentos (nombre, laboratorio,precio,cantidad)  
values('Sertal','Roche',5.2,100);  
insert into medicamentos (nombre, laboratorio,precio,cantidad)  
values('Buscapina','Roche',4.10,200);  
insert into medicamentos (nombre, laboratorio,precio,cantidad)  
values('Amoxidal 500','Bayer',15.60,100);  
insert into medicamentos (nombre, laboratorio,precio,cantidad)  
values('Paracetamol 500','Bago',1.90,200);  
insert into medicamentos (nombre, laboratorio,precio,cantidad)  
values('Bayaspirina','Bayer',2.10,150);  
insert into medicamentos (nombre, laboratorio,precio,cantidad)  
values('Amoxidal jarabe','Bayer',5.10,250);
```

5- Recupere los códigos y nombres de los medicamentos cuyo laboratorio sea 'Roche' y cuyo precio

sea menor a 5:

```
select codigo,nombre  
from medicamentos  
where laboratorio='Roche' and  
precio<5;
```

Quedó seleccionado 1 registro, es el único que cumple con ambas condiciones.

6- Recupere los medicamentos cuyo laboratorio sea 'Roche' o cuyo precio sea menor a 5:

```
select * from medicamentos
```

```
where laboratorio='Roche' or  
precio<5;
```

Note que la salida es diferente, hemos cambiado el operador de la sentencia anterior. Ahora se recuperaron todos los registros cuyo laboratorio es igual a 'Roche' y todos los registros cuyo precio es menor a 5. Aquellos registros que no cumplieron con la condición 1 (no son de "Roche") ni con la condicion2 (no cuestan menos de 5) no aparecen.

7- Muestre todos los registros que no sean de "Bayer" usando el operador "not".

8- Muestre todos los medicamentos cuyo laboratorio NO sea "Bayer" y cuya cantidad sea=100:

```
select * from medicamentos  
where not laboratorio='Bayer' and  
cantidad=100;
```

9- Muestre todos los medicamentos cuyo laboratorio sea "Bayer" y cuya cantidad NO sea=100:

```
select * from medicamentos  
where laboratorio='Bayer' and  
not cantidad=100;
```

Analice estas 2 últimas sentencias. El operador "not" afecta a la condición a la cual antecede, no

a las siguientes. Los resultados de estas 2 sentencias son diferentes.

10- Elimine todos los registros cuyo laboratorio sea igual a "Bayer" y su precio sea mayor a 10:

```
delete from medicamentos  
where laboratorio='Bayer' and  
precio>10;
```

Sólo 1 registro debe ser eliminado, el que cumple con ambas condiciones.

11- Cambie la cantidad por 200, a todos los medicamentos de "Roche" cuyo precio sea mayor a 5:

```
update medicamentos set cantidad=200
```

```
where laboratorio='Roche' and  
precio>5;
```

Un solo registro fue actualizado porque sólo uno cumplió con las condiciones especificadas.

12- Borre los medicamentos cuyo laboratorio sea "Bayer" o cuyo precio sea menor a 3. Antes veamos

cuáles cumplen con la condición, los registros 5 y 6 son de "Bayer, cumplen con la primera condición, los registros 4 y 5 cumplen con la segunda condición, es decir, se borrarán 3 registros:

el 4 porque cumple con la segunda condición, el 5 porque cumple con ambas y el 6 porque cumple con

la primera. Tipeamos:

```
delete from medicamentos
```

```
where laboratorio='Bayer' or precio<3;
```

4-A --

Problema:

Una pequeña biblioteca de barrio registra los préstamos de sus libros en una tabla llamada "prestamos". En ella almacena la siguiente información:

- título del libro,
- documento de identidad del socio a quien se le presta el libro,
- fecha de préstamo,
- fecha de devolución del libro,
- devuelto: si el libro ha sido o no devuelto.

1- Elimine la tabla "prestamos" si existe.

2- Necesitamos una clave que identifique cada registro en la tabla "prestamos". El mismo libro no puede prestarse en la misma fecha.

3- Cree la tabla:

```
create table prestamos(
```

```
titulo varchar(40) not null,  
documento char(8) not null,  
fechaprestamo date not null,  
fechadevolucion date,  
devuelto char(1) default 'N',  
primary key(titulo,fechaprestamo)  
);
```

4- Ingrese los siguientes registros para la tabla "prestamos":

```
insert into prestamos (titulo,documento,fechaprestamo)  
values('Manual de 1 grado','22333444','2006-07-10');  
insert into prestamos (titulo,documento,fechaprestamo)  
values('Manual de 1 grado','22333444','2006-07-20');  
insert into prestamos (titulo,documento,fechaprestamo)  
values('Manual de 1 grado','23333444','2006-07-15');  
insert into prestamos (titulo,documento,fechaprestamo)  
values('El aleph','22333444','2006-07-10');  
insert into prestamos (titulo,documento,fechaprestamo)  
values('El aleph','30333444','2006-08-10');
```

Note que un mismo libro fue prestado a un mismo socio pero en una fecha distinta.

5- Intente ingresar un valor de clave primaria repetida:

```
insert into prestamos (titulo,documento,fechaprestamo)  
values('Manual de 1 grado','25333444','2006-07-10');
```

5- A ---

A) Un consultorio médico en el cual trabajan 3 médicos registra las consultas de los pacientes en

una tabla llamada "consultas".

1- Elimine la tabla si existe.

2- La tabla contiene los siguientes datos:

- fechayhora: datetime not null, fecha y hora de la consulta,
- medico: varchar(30), not null, nombre del médico (Perez,Lopez,Duarte),
- documento: char(8) not null, documento del paciente,
- paciente: varchar(30), nombre del paciente,
- obrasocial: varchar(30), nombre de la obra social ('IPAM','PAMI').

);

3- Un médico sólo puede atender a un paciente en una fecha y hora determinada. En una fecha y hora

determinada, varios médicos atienden a distintos pacientes. Cree la tabla definiendo una clave primaria compuesta:

```
create table consultas(  
    fechayhora datetime not null,  
    medico varchar(30) not null,  
    documento char(8) not null,  
    paciente varchar(30),  
    obrasocial varchar(30),  
    primary key(fechayhora,medico)  
);
```

4- Ingrese varias consultas para un mismo médico en distintas horas el mismo día.

5- Ingrese varias consultas para diferentes médicos en la misma fecha y hora.

6- Intente ingresar una consulta para un mismo médico en la misma hora el mismo día.

B) Un club dicta clases de distintos deportes. En una tabla llamada "inscriptos" almacena la información necesaria.

1- Elimine la tabla "inscriptos" si existe.

2- La tabla contiene los siguientes campos:

- documento del socio alumno: char(8) not null
- nombre del socio: varchar(30),
- nombre del deporte (tenis, futbol, natación, basquet): varchar(15) not null,
- año de inscripción: year,
- matrícula: si la matrícula ha sido o no pagada ('s' o 'n').

3- Necesitamos una clave primaria que identifique cada registro. Un socio puede inscribirse en varios deportes en distintos años. Un socio no puede inscribirse en el mismo deporte el mismo año.

varios socios se inscriben en un mismo deporte. Cree la tabla con una clave compuesta:

```
create table inscriptos(  
    documento char(8) not null,  
    nombre varchar(30),  
    deporte varchar(15) not null,  
    año year,  
    matricula char(1),  
    primary key(documento,deporte,año)  
);
```

4- Inscriba a varios alumnos en el mismo deporte en el mismo año.

5- Inscriba a un mismo alumno en varios deportes en el mismo año.

6- Ingrese un registro con el mismo documento de socio en el mismo deporte en distintos años.

7- Intente inscribir a un socio alumno en un deporte en el cual ya esté inscripto en un año en el cual ya se haya inscripto.

8- Intente eliminar un campo parte de la clave.

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

1- Elimine la tabla si existe.

2- Intente crear la tabla con la siguiente estructura:

```
create table facturas(  
  serie char(1) not null,  
  numero int(10) zerofill auto_increment,  
  descripcion varchar(30),  
  precioporunidad decimal(5,2) unsigned,  
  cantidad tinyint unsigned,  
  primary key (serie,numero)  
);
```

3- Aparece un mensaje de error, la tabla no puede ser creada porque el campo definido como "auto\_increment" es secundario (primero está "serie") y sabemos que un campo "auto\_increment"

debe estar primero en orden al ser definido parte de la clave compuesta.

4- Cree la tabla cambiando el orden de los campos establecidos como clave primaria:

```
create table facturas(  
  serie char(1) not null,  
  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,serie,numeroitem)

);

Tenga en cuenta al ingresar registros que el campo "numero" se autoincrementará sin tener en cuenta

los demás campos.

5- Ingrese 3 registros con igual "serie", "numero" y distintos números de items:

```
insert into facturas (serie,numero,numeroitem,descripcion,precioporunidad,cantidad)
```

```
values('A',100,1,'escuadra 20 cm.',2.50,20);
```

```
insert into facturas (serie,numero,numeroitem,descripcion,precioporunidad,cantidad)
```

```
values('A',100,2,'escuadra 50 cm.',5,30);
```

```
insert into facturas (serie,numero,numeroitem,descripcion,precioporunidad,cantidad)
```

```
values('A',100,3,'goma lapiz-tinta',0.35,100);
```

6- Ingrese los siguientes registros:

```
insert into facturas (serie,numero,numeroitem,descripcion,precioporunidad,cantidad)
```

```
values('A',102,1,'lapices coloresx6',4.40,50);
```

```
insert into facturas (serie,numero,numeroitem,descripcion,precioporunidad,cantidad)
```

```
values('A',102,2,'lapices coloresx12',8,60);
```

```
insert into facturas (serie,numero,numeroitem,descripcion,precioporunidad,cantidad)
```

```
values('B',102,1,'lapices coloresx24',12.35,100);
```

```
insert into facturas (serie,numero,numeroitem,descripcion,precioporunidad,cantidad)
```

```
values('B',102,2,'goma lapiz-tinta',0.35,200);
```

7- Ingrese los siguientes registros y vea qué valor da al "numero" que no se ingresa:

```
insert into facturas (serie,numeroitem,descripcion,precioporunidad,cantidad)
```

```
values('A',1,'compas plastico',12,50);
```

```
insert into facturas (serie,numeroitem,descripcion,precioporunidad,cantidad)
```

```
values('A',1,'compas metal',18.90,80);
```

8- Intente ingresar un registro con valores de clave repetida:

```
insert into facturas (serie,numero,numeroitem,descripcion,precioporunidad,cantidad)
values('A',104,1,'compas metal',18.90,80);
```

9- Muestre los registros concatenando "serie" con "numero", usando un alias para esa columna,

muestre los demás campos y ordene por el alias:

```
select concat_ws('-',serie,numero) as serienumero,
numeroitem,descripcion,precioporunidad,cantidad
from facturas
order by serienumero;
```

10- Agrupe los registros por serie y número de factura y muestre el total (en una columna calculada) de cada factura:

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