

27 - Funciones para el uso de fechas y horas

Primer problema:

Una empresa almacena los datos de sus empleados en una tabla denominada "empleados".

1- Elimine la tabla si existe:

```
if object_id ('empleados') is not null  
drop table empleados;
```

2- Cree la tabla:

```
create table empleados(  
    nombre varchar(30) not null,  
    apellido varchar(20) not null,  
    documento char(8),  
    fechanacimiento datetime,  
    fechaingreso datetime,  
    sueldo decimal(6,2),  
    primary key(documento)  
);
```

3- Ingrese algunos registros:

```
insert into empleados  
values('Ana','Acosta','22222222','1970/10/10','1995/05/05',228.50);  
insert into empleados  
values('Carlos','Caseres','25555555','1978/02/06','1998/05/05',309);  
insert into empleados  
values('Francisco','Garcia','26666666','1978/10/15','1998/10/02',250.68);  
insert into empleados  
values('Gabriela','Garcia','30000000','1985/10/25','2000/12/22',300.25);  
insert into empleados  
values('Luis','Lopez','31111111','1987/02/10','2000/08/21',350.98);
```

4- Muestre nombre y apellido concatenados, con el apellido en letras mayúsculas, el documento precedido por "DNI Nº " y el sueldo precedido por "\$ ".

5- Muestre el documento y el sueldo redondeado hacia arriba y precedido por "\$ ".

6- Muestre los nombres y apellidos de los empleados que cumplen años en el mes "october" (3 registros)

7- Muestre los nombres y apellidos de los empleados que ingresaron en un determinado año (2 registros).

28 - Ordenar registros (order by)

Primer problema:

En una página web se guardan los siguientes datos de las visitas: número de visita, nombre, mail, país, fecha.

1- Elimine la tabla "visitas", si existe:

```
if object_id('visitas') is not null  
drop table visitas;
```

2- Créela con la siguiente estructura:

```
create table visitas (  
    numero int identity,  
    nombre varchar(30) default 'Anonimo',  
    mail varchar(50),  
    pais varchar (20),  
    fecha datetime,  
    primary key(numero)  
);
```

3- Ingrese algunos registros:

```
insert into visitas (nombre,mail,pais,fecha)  
values ('Ana Maria Lopez','AnaMaria@hotmail.com','Argentina','2006-10-10 10:10');  
insert into visitas (nombre,mail,pais,fecha)  
values ('Gustavo Gonzalez','GustavoGGonzalez@hotmail.com','Chile','2006-10-10  
21:30');  
insert into visitas (nombre,mail,pais,fecha)  
values ('Juancito','JuanJosePerez@hotmail.com','Argentina','2006-10-11 15:45');  
insert into visitas (nombre,mail,pais,fecha)  
values ('Fabiola Martinez','MartinezFabiola@hotmail.com','Mexico','2006-10-12  
08:15');  
insert into visitas (nombre,mail,pais,fecha)  
values ('Fabiola Martinez','MartinezFabiola@hotmail.com','Mexico','2006-09-12  
20:45');  
insert into visitas (nombre,mail,pais,fecha)  
values ('Juancito','JuanJosePerez@hotmail.com','Argentina','2006-09-12 16:20');  
insert into visitas (nombre,mail,pais,fecha)  
values ('Juancito','JuanJosePerez@hotmail.com','Argentina','2006-09-15 16:25');
```

4- Ordene los registros por fecha, en orden descendente.

5- Muestre el nombre del usuario, país y el nombre del mes, ordenado por país (ascendente) y nombre del mes (descendente)

6- Muestre el país, el mes, el día y la hora y ordene las visitas por nombre del mes, del día y la hora.

7- Muestre los mail, país, ordenado por país, de todos los que visitaron la página en octubre (4 registros)

29 - Operadores lógicos (and - or - not)

Primer problema:

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

1- Elimine la tabla, si existe:

```
if object_id('medicamentos') is not null
drop table medicamentos;
```

2- Cree la tabla con la siguiente estructura:

```
create table medicamentos(
codigo int identity,
nombre varchar(20),
laboratorio varchar(20),
precio decimal(5,2),
cantidad tinyint,
primary key(codigo)
);
```

3- Ingrese algunos registros:

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

4- Recupere los códigos y nombres de los medicamentos cuyo laboratorio sea 'Roche' y cuyo precio sea menor a 5 (1 registro cumple con ambas condiciones)

5- Recupere los medicamentos cuyo laboratorio sea 'Roche' o cuyo precio sea menor a 5 (4 registros):

```
select * from medicamentos
where laboratorio='Roche' or
precio<5;
```

Note que el resultado es diferente al del punto 4, hemos cambiado el operador de la sentencia anterior.

6- Muestre todos los medicamentos cuyo laboratorio NO sea "Bayer" y cuya cantidad sea=100 (1 registro)

7- Muestre todos los medicamentos cuyo laboratorio sea "Bayer" y cuya cantidad NO sea=100 (2 registros):

```
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 los puntos 6 y 7 son diferentes.

8- Elimine todos los registros cuyo laboratorio sea igual a "Bayer" y su precio sea mayor a 10 (1 registro eliminado)

9- Cambie la cantidad por 200, a todos los medicamentos de "Roche" cuyo precio sea mayor a 5 (1 registro afectado)

10- Borre los medicamentos cuyo laboratorio sea "Bayer" o cuyo precio sea menor a 3 (3 registros borrados)

Segundo problema:

Trabajamos con la tabla "peliculas" de un video club que alquila películas en video.

1- Elimine la tabla, si existe;
if object_id('peliculas') is not null
drop table peliculas;

2- Créela con la siguiente estructura:

```
create table peliculas(  
codigo int identity,  
titulo varchar(40) not null,  
actor varchar(20),  
duracion tinyint,  
primary key (codigo)  
);
```

3- Ingrese algunos registros:

```
insert into peliculas  
values('Mision imposible','Tom Cruise',120);  
insert into peliculas  
values('Harry Potter y la piedra filosofal','Daniel R.',180);  
insert into peliculas  
values('Harry Potter y la camara secreta','Daniel R.',190);  
insert into peliculas  
values('Mision imposible 2','Tom Cruise',120);  
insert into peliculas  
values('Mujer bonita','Richard Gere',120);  
insert into peliculas  
values('Tootsie','D. Hoffman',90);  
insert into peliculas  
values('Un oso rojo','Julio Chavez',100);
```

```
insert into peliculas  
values('Elsa y Fred','China Zorrilla',110);
```

4- Recupere los registros cuyo actor sea "Tom Cruise" or "Richard Gere"
(3 registros)

5- Recupere los registros cuyo actor sea "Tom Cruise" y duración menor
a 100 (ninguno cumple ambas
condiciones)

6- Cambie la duración a 200, de las películas cuyo actor sea "Daniel R."
y cuya duración sea 180 (1
registro afectado)

7- Borre todas las películas donde el actor NO sea "Tom Cruise" y cuya
duración sea mayor o igual a
100 (2 registros eliminados)

30 - Otros operadores relacionales (is null)

Primer problema:

Trabajamos con la tabla "peliculas" de un video club que alquila películas en video.

1- Elimine la tabla, si existe;
if object_id('peliculas') is not null
drop table peliculas;

2- Créela con la siguiente estructura:

```
create table peliculas(  
codigo int identity,  
titulo varchar(40) not null,  
actor varchar(20),  
duracion tinyint,  
primary key (codigo)  
);
```

3- Ingrese algunos registros:

```
insert into peliculas  
values('Mision imposible','Tom Cruise',120);  
insert into peliculas  
values('Harry Potter y la piedra filosofal','Daniel R.',null);  
insert into peliculas  
values('Harry Potter y la camara secreta','Daniel R.',190);  
insert into peliculas  
values('Mision imposible 2','Tom Cruise',120);  
insert into peliculas  
values('Mujer bonita',null,120);  
insert into peliculas  
values('Tootsie','D. Hoffman',90);
```

```
insert into peliculas (titulo)
values('Un oso rojo');
```

4- Recupere las películas cuyo actor sea nulo (2 registros)

5- Cambie la duración a 0, de las películas que tengan duración igual a "null" (2 registros)

6- Borre todas las películas donde el actor sea "null" y cuya duración sea 0 (1 registro)

31 - Otros operadores relacionales (between)

Primer problema:

En una página web se guardan los siguientes datos de las visitas: número de visita, nombre, mail, pais, fechayhora de la visita.

1- Elimine la tabla "visitas", si existe:
if object_id('visitas') is not null
drop table visitas;

2- Créela con la siguiente estructura:
create table visitas (
numero int identity,
nombre varchar(30) default 'Anonimo',
mail varchar(50),
pais varchar (20),
fechayhora datetime,
primary key(numero)
);

3- Ingrese algunos registros:

```
insert into visitas (nombre,mail,pais,fechayhora)
values ('Ana Maria Lopez','AnaMaria@hotmail.com','Argentina','2006-10-10 10:10');
insert into visitas (nombre,mail,pais,fechayhora)
values ('Gustavo Gonzalez','GustavoGGonzalez@gotmail.com','Chile','2006-10-10
21:30');
insert into visitas (nombre,mail,pais,fechayhora)
values ('Juancito','JuanJosePerez@hotmail.com','Argentina','2006-10-11 15:45');
insert into visitas (nombre,mail,pais,fechayhora)
values ('Fabiola Martinez','MartinezFabiola@hotmail.com','Mexico','2006-10-12
08:15');
insert into visitas (nombre,mail,pais,fechayhora)
values ('Fabiola Martinez','MartinezFabiola@hotmail.com','Mexico','2006-09-12
20:45');
insert into visitas (nombre,mail,pais,fechayhora)
values ('Juancito','JuanJosePerez@gmail.com','Argentina','2006-09-12 16:20');
insert into visitas (nombre,mail,pais,fechayhora)
values ('Juancito','JuanJosePerez@hotmail.com','Argentina','2006-09-15 16:25');
insert into visitas (nombre,mail,pais)
values ('Federico1','federicogarcia@xaxamail.com','Argentina');
```

4- Seleccione los usuarios que visitaron la página entre el '2006-09-12' y '2006-10-11' (5 registros)

Note que incluye los de fecha mayor o igual al valor mínimo y menores al valor máximo, y que los valores null no se incluyen.

5- Recupere las visitas cuyo número se encuentra entre 2 y 5 (4 registros)

Note que incluye los valores límites.

Segundo problema:

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

1- Elimine la tabla "autos" si existe:

```
if object_id('autos') is not null
drop table autos;
```

2- Cree la tabla con la siguiente estructura:

```
create table autos(
patente char(6),
marca varchar(20),
modelo char(4),
precio decimal(8,2),
primary key(patente)
);
```

3- Ingrese algunos registros:

```
insert into autos
values('ACD123','Fiat 128','1970',15000);
insert into autos
values('ACG234','Renault 11','1980',40000);
insert into autos
values('BCD333','Peugeot 505','1990',80000);
insert into autos
values('GCD123','Renault Clio','1995',70000);
insert into autos
values('BCC333','Renault Megane','1998',95000);
insert into autos
values('BVF543','Fiat 128','1975',20000);
```

4- Seleccione todos los autos cuyo modelo se encuentre entre '1970' y '1990' usando el operador

"between" y ordénelos por dicho campo(4 registros)

5- Seleccione todos los autos cuyo precio esté entre 50000 y 100000.

32 - Otros operadores relacionales (in)

Primer problema:

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

1- Elimine la tabla, si existe:

```
if object_id('medicamentos') is not null
drop table medicamentos;
```

2- Cree la tabla con la siguiente estructura:

```
create table medicamentos(
codigo int identity,
nombre varchar(20),
laboratorio varchar(20),
precio decimal(6,2),
cantidad tinyint,
fechavencimiento datetime not null,
primary key(codigo)
);
```

3- Ingrese algunos registros:

```
insert into medicamentos
values('Sertal','Roche',5.2,1,'2015-02-01');
insert into medicamentos
values('Buscapina','Roche',4.10,3,'2016-03-01');
insert into medicamentos
values('Amoxidal 500','Bayer',15.60,100,'2017-05-01');
insert into medicamentos
values('Paracetamol 500','Bago',1.90,20,'2018-02-01');
insert into medicamentos
values('Bayaspirina','Bayer',2.10,150,'2019-12-01');
insert into medicamentos
values('Amoxidal jarabe','Bayer',5.10,250,'2020-10-01');
```

4- Recupere los nombres y precios de los medicamentos cuyo laboratorio sea "Bayer" o "Bago"

empleando el operador "in" (4 registros)

5- Seleccione los remedios cuya cantidad se encuentre entre 1 y 5 empleando el operador "between" y

luego el operador "in" (2 registros):

```
select * from medicamentos
where cantidad between 1 and 5;
select * from medicamentos
where cantidad in (1,2,3,4,5);
```

Note que es más conveniente emplear, en este caso, el operador ""between".

33 - Búsqueda de patrones (like - not like)

Primer problema:

Una empresa almacena los datos de sus empleados en una tabla "empleados".

1- Elimine la tabla, si existe:

```
if object_id('empleados') is not null  
drop table empleados;
```

2- Cree la tabla:

```
create table empleados(  
    nombre varchar(30),  
    documento char(8),  
    domicilio varchar(30),  
    fechaingreso datetime,  
    seccion varchar(20),  
    sueldo decimal(6,2),  
    primary key(documento)  
);
```

3- Ingrese algunos registros:

```
insert into empleados  
values('Juan Perez','22333444','Colon 123','1990-10-08','Gerencia',900.50);  
insert into empleados  
values('Ana Acosta','23444555','Caseros 987','1995-12-18','Secretaria',590.30);  
insert into empleados  
values('Lucas Duarte','25666777','Sucre 235','2005-05-15','Sistemas',790);  
insert into empleados  
values('Pamela Gonzalez','26777888','Sarmiento 873','1999-02-12','Secretaria',550);  
insert into empleados  
values('Marcos Juarez','30000111','Rivadavia 801','2002-09-22','Contaduria',630.70);  
insert into empleados  
values('Yolanda Perez','35111222','Colon 180','1990-10-08','Administracion',400);  
insert into empleados  
values('Rodolfo Perez','35555888','Coronel Olmedo 588','1990-05-  
28','Sistemas',800);
```

4- Muestre todos los empleados con apellido "Perez" empleando el operador "like" (3 registros)

5- Muestre todos los empleados cuyo domicilio comience con "Co" y tengan un "8" (2 registros)

6- Seleccione todos los empleados cuyo documento finalice en 0,2,4,6 u 8 (4 registros)

7- Seleccione todos los empleados cuyo documento NO comience con 1 ni 3 y cuyo nombre finalice en "ez" (2 registros)

8- Recupere todos los nombres que tengan una "y" o una "j" en su nombre o apellido (3 registros)

- 9- Muestre los nombres y sección de los empleados que pertenecen a secciones que comiencen con "S" o "G" y tengan 8 caracteres (3 registros)
- 10- Muestre los nombres y sección de los empleados que pertenecen a secciones que NO comiencen con "S" o "G" (2 registros)
- 11- Muestre todos los nombres y sueldos de los empleados cuyos sueldos incluyen centavos (3 registros)
- 12- Muestre los empleados que hayan ingresado en "1990" (3 registros)

34 - Contar registros (count)

Primer problema:

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

1- Elimine la tabla, si existe:

```
if object_id('medicamentos') is not null  
drop table medicamentos;
```

2- Cree la tabla con la siguiente estructura:

```
create table medicamentos(  
codigo int identity,  
nombre varchar(20),  
laboratorio varchar(20),  
precio decimal(6,2),  
cantidad tinyint,  
fechavencimiento datetime not null,  
numerolote int default null,  
primary key(codigo)  
);
```

3- Ingrese algunos registros:

```
insert into medicamentos  
values('Sertal','Roche',5.2,1,'2015-02-01',null);  
insert into medicamentos  
values('Buscapina','Roche',4.10,3,'2016-03-01',null);  
insert into medicamentos  
values('Amoxidal 500','Bayer',15.60,100,'2017-05-01',null);  
insert into medicamentos  
values('Paracetamol 500','Bago',1.90,20,'2018-02-01',null);  
insert into medicamentos  
values('Bayaspirina',null,2.10,null,'2019-12-01',null);  
insert into medicamentos  
values('Amoxidal jarabe','Bayer',null,250,'2019-12-15',null);
```

4- Muestre la cantidad de registros empleando la función "count(*)" (6 registros)

5- Cuente la cantidad de medicamentos que tienen laboratorio conocido (5 registros)

6- Cuente la cantidad de medicamentos que tienen precio distinto a "null" y que tienen cantidad distinto a "null", disponer alias para las columnas.

7- Cuente la cantidad de remedios con precio conocido, cuyo laboratorio comience con "B" (2 registros)

8- Cuente la cantidad de medicamentos con número de lote distinto de "null" (0 registros)

35 - Contar registros (count_big)

Primer problema:

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

1- Elimine la tabla, si existe:

```
if object_id('medicamentos') is not null
drop table medicamentos;
```

2- Cree la tabla con la siguiente estructura:

```
create table medicamentos(
codigo int identity,
nombre varchar(20),
laboratorio varchar(20),
precio decimal(6,2),
cantidad tinyint,
fechavencimiento datetime not null,
numerolote int default null,
primary key(codigo)
);
```

3- Ingrese algunos registros:

```
insert into medicamentos
values('Sertal','Roche',5.2,1,'2015-02-01',null);
insert into medicamentos
values('Buscapina','Roche',4.10,3,'2016-03-01',null);
insert into medicamentos
values('Amoxidal 500','Bayer',15.60,100,'2017-05-01',null);
insert into medicamentos
values('Paracetamol 500','Bago',1.90,20,'2018-02-01',null);
insert into medicamentos
values('Bayaspirina',null,2.10,null,'2019-12-01',null);
insert into medicamentos
values('Amoxidal jarabe','Bayer',null,250,'2019-12-15',null);
```

4- Muestre la cantidad de registros empleando la función "count_big(*)" (6 registros)

5- Cuente la cantidad de laboratorios distintos (3 registros)

6- Cuente la cantidad de medicamentos que tienen precio y cantidad distinto de "null" (5 y 5)

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

Primer problema:

Una empresa almacena los datos de sus empleados en una tabla "empleados".

1- Elimine la tabla, si existe:

```
if object_id('empleados') is not null  
drop table empleados;
```

2- Cree la tabla:

```
create table empleados(  
    nombre varchar(30),  
    documento char(8),  
    domicilio varchar(30),  
    seccion varchar(20),  
    sueldo decimal(6,2),  
    cantidadhijos tinyint,  
    primary key(documento)  
);
```

3- Ingrese algunos registros:

```
insert into empleados  
values('Juan Perez','22333444','Colon 123','Gerencia',5000,2);  
insert into empleados  
values('Ana Acosta','23444555','Caseros 987','Secretaria',2000,0);  
insert into empleados  
values('Lucas Duarte','25666777','Sucre 235','Sistemas',4000,1);  
insert into empleados  
values('Pamela Gonzalez','26777888','Sarmiento 873','Secretaria',2200,3);  
insert into empleados  
values('Marcos Juarez','30000111','Rivadavia 801','Contaduria',3000,0);  
insert into empleados  
values('Yolanda Perez','35111222','Colon 180','Administracion',3200,1);  
insert into empleados  
values('Rodolfo Perez','35555888','Coronel Olmedo 588','Sistemas',4000,3);  
insert into empleados  
values('Martina Rodriguez','30141414','Sarmiento 1234','Administracion',3800,4);  
insert into empleados  
values('Andres Costa','28444555',default,'Secretaria',null,null);
```

4- Muestre la cantidad de empleados usando "count" (9 empleados)

5- Muestre la cantidad de empleados con sueldo no nulo de la sección "Secretaria" (2 empleados)

6- Muestre el sueldo más alto y el más bajo colocando un alias (5000 y 2000)

7- Muestre el valor mayor de "cantidadhijos" de los empleados "Perez" (3 hijos)

8- Muestre el promedio de sueldos de todo los empleados (3400. Note que hay un sueldo nulo y no es tenido en cuenta)

9- Muestre el promedio de sueldos de los empleados de la sección "Secretaría" (2100)

10- Muestre el promedio de hijos de todos los empleados de "Sistemas" (2)

37 - Agrupar registros (group by)

Primer 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:

```
if object_id('visitantes') is not null  
drop table visitantes;
```

2- Cree la tabla con la siguiente estructura:

```
create table visitantes(  
    nombre varchar(30),  
    edad tinyint,  
    sexo char(1) default 'f',  
    domicilio varchar(30),  
    ciudad varchar(20) default 'Cordoba',  
    telefono varchar(11),  
    mail varchar(30) default 'no tiene',  
    montocompra decimal (6,2)  
);
```

3- Ingrese algunos registros:

```
insert into visitantes  
values ('Susana Molina',35,default,'Colon 123',default,null,null,59.80);  
insert into visitantes  
values ('Marcos Torres',29,'m',default,'Carlos  
Paz',default,'marcostorres@hotmail.com',150.50);  
insert into visitantes  
values ('Mariana Juarez',45,default,default,'Carlos Paz',null,default,23.90);  
insert into visitantes (nombre, edad,sexo,telefono, mail)  
values ('Fabian Perez',36,'m','4556677','fabianperez@xaxamail.com');  
insert into visitantes (nombre, ciudad, montocompra)  
values ('Alejandra Gonzalez','La Falda',280.50);  
insert into visitantes (nombre, edad,sexo, ciudad, mail,montocompra)  
values ('Gaston Perez',29,'m','Carlos Paz','gastonperez1@gmail.com',95.40);  
insert into visitantes  
values ('Liliana Torres',40,default,'Sarmiento 876',default,default,default,85);  
insert into visitantes  
values ('Gabriela Duarte',21,null,null,'Rio  
Tercero',default,'gabrielaaltorres@hotmail.com',321.50);
```

4- Queremos saber la cantidad de visitantes de cada ciudad utilizando la cláusula "group by" (4 filas devueltas)

- 5- Queremos la cantidad visitantes con teléfono no nulo, de cada ciudad (4 filas devueltas)
- 6- Necesitamos el total del monto de las compras agrupadas por sexo (3 filas)
- 7- Se necesita saber el máximo y mínimo valor de compra agrupados por sexo y ciudad (6 filas)
- 8- Calcule el promedio del valor de compra agrupados por ciudad (4 filas)
- 9- Cuente y agrupe por ciudad sin tener en cuenta los visitantes que no tienen mail (3 filas):
- 10- Realice la misma consulta anterior, pero use la palabra clave "all" para mostrar todos los valores de ciudad, incluyendo las que devuelven cero o "null" en la columna de agregado (4 filas)

Segundo problema:

Una empresa almacena los datos de sus empleados en una tabla "empleados".

1- Elimine la tabla, si existe:

```
if object_id('empleados') is not null
drop table empleados;
```

2- Cree la tabla:

```
create table empleados(
    nombre varchar(30),
    documento char(8),
    domicilio varchar(30),
    seccion varchar(20),
    sueldo decimal(6,2),
    cantidadhijos tinyint,
    fechaingreso datetime,
    primary key(documento)
);
```

3- Ingrese algunos registros:

```
insert into empleados
values('Juan Perez','22333444','Colon 123','Gerencia',5000,2,'1980-05-10');
insert into empleados
values('Ana Acosta','23444555','Caseros 987','Secretaria',2000,0,'1980-10-12');
insert into empleados
values('Lucas Duarte','25666777','Sucre 235','Sistemas',4000,1,'1985-05-25');
```

```

insert into empleados
  values('Pamela Gonzalez','26777888','Sarmiento
873','Secretaria',2200,3,'1990-06-25');
insert into empleados
  values('Marcos Juarez','30000111','Rivadavia
801','Contaduria',3000,0,'1996-05-01');
insert into empleados
  values('Yolanda Perez','35111222','Colon
180','Administracion',3200,1,'1996-05-01');
insert into empleados
  values('Rodolfo Perez','35555888','Coronel Olmedo
588','Sistemas',4000,3,'1996-05-01');
insert into empleados
  values('Martina Rodriguez','30141414','Sarmiento
1234','Administracion',3800,4,'2000-09-01');
insert into empleados
  values('Andres Costa','28444555',default,'Secretaria',null,null,null);

```

4- Cuento la cantidad de empleados agrupados por sección (5 filas)

5- Calcule el promedio de hijos por sección (5 filas):

6- Cuento la cantidad de empleados agrupados por año de ingreso (6 filas)

7- Calcule el promedio de sueldo por sección de los empleados con hijos (4 filas)

8- Realice la misma consulta anterior pero esta vez incluya las secciones que devuelven cero o "null" en la columna de agregado (5 filas)

38 - Seleccionar grupos (having)

Primer problema:

Una empresa tiene registrados sus clientes en una tabla llamada "clientes".

1- Elimine la tabla "clientes", si existe:

```

if object_id('clientes') is not null
  drop table clientes;

```

2- Créela con la siguiente estructura:

```

create table clientes (
  codigo int identity,
  nombre varchar(30) not null,
  domicilio varchar(30),

```

```
ciudad varchar(20),
provincia varchar (20),
telefono varchar(11),
primary key(codigo)
);
```

3- Ingrese algunos registros:

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

4- Obtenga el total de los registros agrupados por ciudad y provincia (6 filas)

5- Obtenga el total de los registros agrupados por ciudad y provincia sin considerar los que tienen menos de 2 clientes (3 filas)

6- Obtenga el total de los clientes que viven en calle "San Martin" (where), agrupados por provincia (group by), teniendo en cuenta todos los valores (all), de aquellas ciudades que tengan menos de 2 clientes (having) y omitiendo la fila correspondiente a la ciudad de "Cordoba" (having) (4 filas devueltas).

Segundo 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:

```
if object_id('visitantes') is not null
drop table visitantes;
```


2- Créela con la siguiente estructura:

```
create table visitantes(  
  nombre varchar(30),  
  edad tinyint,  
  sexo char(1),  
  domicilio varchar(30),  
  ciudad varchar(20),  
  telefono varchar(11),  
  montocompra decimal(6,2) not null  
);
```

3- Ingrese algunos registros:

```
insert into visitantes  
  values ('Susana Molina',28,'f',null,'Cordoba',null,45.50);  
insert into visitantes  
  values ('Marcela Mercado',36,'f','Avellaneda 345','Cordoba','4545454',22.40);  
insert into visitantes  
  values ('Alberto Garcia',35,'m','Gral. Paz 123','Alta Gracia','03547123456',25);  
insert into visitantes  
  values ('Teresa Garcia',33,'f',default,'Alta Gracia','03547123456',120);  
insert into visitantes  
  values ('Roberto Perez',45,'m','Urquiza 335','Cordoba','4123456',33.20);  
insert into visitantes  
  values ('Marina Torres',22,'f','Colon 222','Villa Dolores','03544112233',95);  
insert into visitantes  
  values ('Julieta Gomez',24,'f','San Martin 333','Alta Gracia',null,53.50);  
insert into visitantes  
  values ('Roxana Lopez',20,'f','null','Alta Gracia',null,240);  
insert into visitantes  
  values ('Liliana Garcia',50,'f','Paso 999','Cordoba','4588778',48);  
insert into visitantes  
  values ('Juan Torres',43,'m','Sarmiento 876','Cordoba',null,15.30);
```

4- Obtenga el total de las compras agrupados por ciudad y sexo de aquellas filas que devuelvan un valor superior a 50 (3 filas)

5- Obtenga el total de las compras agrupados por ciudad y sexo (group by), teniendo en cuenta todos los valores (all), considerando sólo los montos de compra superiores a 50 (where), los visitantes con teléfono (where), sin considerar la ciudad de "Cordoba" (having), ordenados por ciudad (order by) (3 filas)

6- Muestre el monto mayor de compra agrupado por ciudad, siempre que dicho valor supere los 50 pesos (having), considerando sólo los visitantes de sexo femenino y domicilio conocido (where) (2 filas)

7- Agrupe por ciudad y sexo, muestre para cada grupo el total de visitantes, la suma de sus compras

y el promedio de compras, ordenado por la suma total y considerando las filas con promedio superior a 30 (3 filas)

39 - Modificador del group by (with rollup)

Primer problema:

Una empresa tiene registrados sus clientes en una tabla llamada "clientes".

1- Elimine la tabla "clientes", si existe:

```
if object_id('clientes') is not null
drop table clientes;
```

2- Créela con la siguiente estructura:

```
create table clientes (
codigo int identity,
nombre varchar(30) not null,
domicilio varchar(30),
ciudad varchar(20),
estado varchar (20),
pais varchar(20),
primary key(codigo)
);
```

3- Ingrese algunos registros:

```
insert into clientes
values ('Lopez Marcos','Colon 111', 'Cordoba','Cordoba','Argentina');
insert into clientes
values ('Perez Ana','San Martin 222', 'Carlos Paz','Cordoba','Argentina');
insert into clientes
values ('Garcia Juan','Rivadavia 333', 'Carlos Paz','Cordoba','Argentina');
insert into clientes
values ('Perez Luis','Sarmiento 444', 'Rosario','Santa Fe','Argentina');
insert into clientes
values ('Gomez Ines','San Martin 987', 'Santa Fe','Santa Fe','Argentina');
insert into clientes
values ('Gomez Ines','San Martin 666', 'Santa Fe','Santa Fe','Argentina');
insert into clientes
values ('Lopez Carlos','Irigoyen 888', 'Cordoba','Cordoba','Argentina');
insert into clientes
values ('Ramos Betina','San Martin 999', 'Cordoba','Cordoba','Argentina');
insert into clientes
values ('Fernando Salas','Mariano Osorio 1234', 'Santiago','Region metropolitana','Chile');
insert into clientes
values ('German Rojas','Allende 345', 'Valparaiso','Region V','Chile');
insert into clientes
values ('Ricardo Jara','Pablo Neruda 146', 'Santiago','Region metropolitana','Chile');
insert into clientes
values ('Joaquin Robles','Diego Rivera 147', 'Guadalajara','Jalisco','Mexico');
```

4- Necesitamos la cantidad de clientes por país y la cantidad total de clientes en una sola consulta
(4 filas)

Note que la consulta retorna los registros agrupados por país y una fila extra en la que la columna

"pais" contiene "null" y la columna con la cantidad muestra la cantidad total.

5- Necesitamos la cantidad de clientes agrupados por país y estado, incluyendo resultados parciales
(9 filas)

Note que la salida muestra los totales por país y estado y produce 4 filas extras: 3 muestran los

totales para cada país, con la columna "estado" conteniendo "null" y 1 muestra el total de todos los

clientes, con las columnas "pais" y "estado" conteniendo "null".

6- Necesitamos la cantidad de clientes agrupados por país, estado y ciudad, empleando "rollup" (16 filas)

El resultado muestra los totales por país, estado y ciudad y genera 9 filas extras: 5 muestran los

totales para cada estado, con la columna correspondiente a "ciudad" conteniendo "null", 3 muestran

los totales para cada país, con las columnas "ciudad" y "estado" conteniendo "null" y 1 muestra el

total de todos los clientes, con las columnas "pais", "estado" y "ciudad" conteniendo "null".

Segundo problema:

Un instituto de enseñanza guarda las notas de sus alumnos en una tabla llamada "notas".

1- Elimine la tabla si existe:

```
if object_id('notas') is not null  
drop table notas;
```

2- Cree la tabla con la siguiente estructura:

```
create table notas(  
    documento char(8) not null,  
    materia varchar(30),  
    nota decimal(4,2)  
);
```

3-Ingresa algunos registros:

```
insert into notas values ('22333444','Programacion',8);  
insert into notas values ('22333444','Programacion',9);  
insert into notas values ('22333444','Ingles',8);  
insert into notas values ('22333444','Ingles',7);  
insert into notas values ('22333444','Ingles',6);  
insert into notas values ('22333444','Sistemas de datos',10);  
insert into notas values ('22333444','Sistemas de datos',9);
```

```
insert into notas values ('23444555','Programacion',5);  
insert into notas values ('23444555','Programacion',4);
```

```
insert into notas values ('23444555','Programacion',3);
insert into notas values ('23444555','Ingles',9);
insert into notas values ('23444555','Ingles',7);
insert into notas values ('23444555','Sistemas de datos',9);
```

```
insert into notas values ('24555666','Programacion',1);
insert into notas values ('24555666','Programacion',3.5);
insert into notas values ('24555666','Ingles',4.5);
insert into notas values ('24555666','Sistemas de datos',6);
```

4- Se necesita el promedio por alumno por materia y el promedio de cada alumno en todas las materias

cursadas hasta el momento (13 registros):

```
select documento,materia,
      avg(nota) as promedio
from notas
group by documento,materia with rollup;
```

Note que hay 4 filas extras, 3 con el promedio de cada alumno y 1 con el promedio de todos los alumnos de todas las materias.

5- Compruebe los resultados parciales de la consulta anterior realizando otra consulta mostrando el promedio de todas las carreras de cada alumno (4 filas)

6- Muestre la cantidad de notas de cada alumno, por materia (9 filas)

7- Realice la misma consulta anterior con resultados parciales incluidos (13 filas)

8- Muestre la nota menor y la mayor de cada alumno y la menor y mayor nota de todos (use "rollup")
(4 filas)

40 - Modificador del group by (with cube)

Primer problema:

Un comercio guarda la información de sus ventas en una tabla llamada "ventas" en la que se registra diariamente, el número de venta, el monto total de la compra, el tipo de pago (c=contado, t=tarjeta) y el nombre del vendedor.

1- Elimine la tabla si existe:

```
if object_id('ventas') is not null
drop table ventas;
```

2- Cree la tabla:

```
create table ventas(
  numero int identity,
  montocompra decimal(6,2),
  tipopago char(1),--c=contado, t=tarjeta
  vendedor varchar(30),
```

```
primary key (numero)
);
```

3- Ingrese algunos registros:

```
insert into ventas
values(100.50,'c','Marisa Perez');
insert into ventas
values(200,'c','Marisa Perez');
insert into ventas
values(50,'t','Juan Lopez');
insert into ventas
values(220,'c','Juan Lopez');
insert into ventas
values(150,'t','Marisa Perez');
insert into ventas
values(550.80,'c','Marisa Perez');
insert into ventas
values(300,'t','Juan Lopez');
insert into ventas
values(25,'c','Marisa Perez');
```

4- Agrupe por "tipopago" y "vendedor" y cuente la cantidad empleando "rollup".

Las agrupaciones de resumen son las siguientes:

- vendedor (tipopago seteado a "null"), 2 filas y
- total (todos los campos seteados a "null"), 1 fila

5- Agrupe por "tipopago" y "vendedor" y cuente la cantidad empleando "cube".

Las agrupaciones de resumen son las siguientes:

- vendedor (tipopago seteado a "null"), 2 filas,
- total (todos los campos seteados a "null"), 1 fila y
- tipopago (vendedor seteado a "null"), 2 filas.

Segundo 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:

```
if object_id('visitantes') is not null
drop table visitantes;
```

2- Créela con la siguiente estructura:

```
create table visitantes(
nombre varchar(30),
edad tinyint,
sexo char(1),
domicilio varchar(30),
ciudad varchar(20),
mail varchar(30) default 'no tiene',
montocompra decimal(6,2)
);
```

3- Ingrese algunos registros:

```
insert into visitantes
  values ('Susana Molina',28,'f',null,'Cordoba',null,45.50);
insert into visitantes
  values ('Marcela Mercado',36,'f','Avellaneda 345','Cordoba',default,22.40);
insert into visitantes
  values ('Alberto Garcia',35,'m',default,'Alta Gracia','albertogarcia@hotmail.com',25);
insert into visitantes
  values ('Teresa Garcia',33,'f',default,'Alta Gracia',default,120);
insert into visitantes
  values ('Roberto Perez',45,'m',null,'Cordoba','robertoperez@xaxamail.com',33.20);
insert into visitantes
  values ('Marina Torres',22,'f',null,'Villa Dolores',default,95);
insert into visitantes
  values ('Julieta Gomez',24,'f','San Martin 333','Alta
Gracia','julietagomez@gmail.com',53.50);
insert into visitantes
  values ('Roxana Lopez',20,'f','null','Alta Gracia',default,240);
insert into visitantes
  values ('Liliana Garcia',50,'f','Paso 999','Cordoba',default,48);
insert into visitantes
  values ('Juan Torres',43,'m','Sarmiento 876','Cordoba',null,15.30);
```

4- Agrupe por sexo y ciudad y calcule el total de la compra empleando "rollup" y luego "cube":

```
select sexo,ciudad,
  sum(montocompra) as total
from visitantes
group by sexo,ciudad
with rollup;
```

Se generan 2 filas con "ciudad" conteniendo "null" y 1 fila con ambos campos seteados a "null".

```
select sexo,ciudad,
  sum(montocompra) as total
from visitantes
group by sexo,ciudad
with cube;
```

Se generan 2 filas con "ciudad" conteniendo "null" (total de sexo femenino y masculino); 1 fila con ambos campos seteados a "null" (monto total de todos los visitantes) y 3 filas con "sexo" conteniendo "null" (monto total de cada ciudad).

5- Calcule la edad promedio de los visitantes agrupando por sexo y ciudad, sin considerar los que no

tienen mail, use ambos modificadores de "group by" ("rollup" y "cube"):

```
select sexo,ciudad,
  avg(edad) as 'edad promedio'
from visitantes
where mail is not null and
mail <>'no tiene'
group by sexo,ciudad
with rollup;
```

Se generan 2 filas con el promedio de edad por sexo de cada ciudad y 1 fila con el promedio de edad de todos los visitantes.

```
select sexo,ciudad,  
avg(edad) as 'edad promedio'  
from visitantes  
where mail is not null and  
mail <>'no tiene'  
group by sexo,ciudad  
with cube;
```

Se generan 2 filas con el promedio de edad por sexo de cada ciudad, 1 fila con el promedio de edad

de todos los visitantes y 3 filas con el promedio de edad por ciudad.

41 - Función grouping

Primer problema:

Una empresa tiene registrados sus empleados en una tabla llamada "empleados".

1- Elimine la tabla si existe:

```
if object_id('empleados') is not null  
drop table empleados;
```

2- Créela con la siguiente estructura:

```
create table empleados (  
documento varchar(8) not null,  
nombre varchar(30),  
sexo char(1),  
estadocivil char(1),--c=casado, s=soltero,v=viudo  
seccion varchar(20),  
primary key (documento)  
);
```

3- Ingrese algunos registros:

```
insert into empleados  
values ('22222222','Alberto Lopez','m','c','Sistemas');  
insert into empleados  
values ('23333333','Beatriz Garcia','f','c','Administracion');  
insert into empleados  
values ('24444444','Carlos Fuentes','m','s','Administracion');  
insert into empleados  
values ('25555555','Daniel Garcia','m','s','Sistemas');  
insert into empleados  
values ('26666666','Ester Juarez',null,'c','Sistemas');  
insert into empleados  
values ('27777777','Fabian Torres',null,'s','Sistemas');  
insert into empleados  
values ('28888888','Gabriela Lopez','f',null,'Sistemas');  
insert into empleados  
values ('29999999','Hector Garcia','m',null,'Administracion');
```

4- Cuento la cantidad de empleados agrupados por sexo y estado civil, empleando "rollup".

Es difícil distinguir los valores de detalle y resumen.

5- Realice la misma consulta anterior pero emplee la función "grouping" para los dos campos por los

que se agrupa para distinguir los valores de resumen y de detalle.

Note que las columnas de resumen contienen 1 y las de detalle 0.

6- Realice la misma consulta anterior pero con "cube" en lugar de "rollup", distinguiendo los

valores de resumen y de detalle.

Note que las columnas de resumen contienen 1 y las de detalle 0.

42 - Cláusulas compute y compute by

Primer problema:

La cláusula compute y compute by están discontinuadas en las versiones de SQL Server 2012 y sucesivas.

No utilice esta característica en nuevos proyectos y modifique lo antes posible las aplicaciones que actualmente la utilizan. En su lugar, utilice rollup.

La provincia almacena en una tabla llamada "inmuebles" los siguientes datos de los inmuebles y sus

propietarios para cobrar impuestos:

1- Elimine la tabla si existe:

```
if object_id('inmuebles') is not null
drop table inmuebles;
```

2- Créela con la siguiente estructura:

```
create table inmuebles (
    documento varchar(8) not null,
    nombre varchar(30),
    domicilio varchar(20),
    barrio varchar(20),
    ciudad varchar(20),
    tipo char(1),--b=baldio, e: edificado
    superficie decimal (8,2),
    monto decimal (8,2)
);
```

3- Ingrese algunos registros:

```
insert into inmuebles
values ('11111111','Alberto Acosta','Avellaneda 800','Centro','Cordoba','e',100,1200);
insert into inmuebles
values ('11111111','Alberto Acosta','Sarmiento 245','Gral.
Paz','Cordoba','e',200,2500);
insert into inmuebles
```



```

values ('22222222','Beatriz Barrios','San Martin 202','Centro','Cordoba','e',250,1900);
insert into inmuebles
values ('33333333','Carlos Caseres','Paso 1234','Alberdi','Cordoba','b',200,1000);
insert into inmuebles
values ('33333333','Carlos Caseres','Guemes 876','Alberdi','Cordoba','b',300,1500);
insert into inmuebles
values ('44444444','Diana Dominguez','Calderon
456','Matienzo','Cordoba','b',200,800);
insert into inmuebles
values ('55555555','Estela Fuentes','San Martin 321','Flores','Carlos
Paz','e',500,4500);
insert into inmuebles
values ('55555555','Estela Fuentes','Lopez y Planes
853','Alberdi','Cordoba','e',350,2200);

```

4- Muestre todos los datos y el promedio del monto empleando "compute" (1 resultado parcial)

5- Empleando cláusulas "compute" consulte el promedio de las superficies y el total de los montos (2 columnas extras)

6- Realice la misma consulta anterior pero empleando "compute by" para obtener resultados parciales por documento,barrio y ciudad.

7- Realice la misma consulta anterior pero con resultados parciales por documento y barrio (6 resultados parciales dobles)

8- Realice la misma consulta anterior pero con resultados parciales por documento (4 resultados parciales dobles)

9- Intente realizar la misma consulta anterior pero con resultados parciales por documento y ciudad.
Aparece un mensaje de error indicando que el subgrupo de campos listados luego del "by" no es correcto.

10- Combine cláusulas "compute" con "compute by" para averiguar el total de monto a pagar por propietario y el promedio de monto de todos (4 resultados parciales y 1 general)

43 - Registros duplicados (distinct)

Primer problema:

Una empresa tiene registrados sus clientes en una tabla llamada "clientes".

1- Elimine la tabla "clientes", si existe:

```
if object_id('clientes') is not null
```

```
drop table clientes;
```

2- Créela con la siguiente estructura:

```
create table clientes (  
  codigo int identity,  
  nombre varchar(30) not null,  
  domicilio varchar(30),  
  ciudad varchar(20),  
  provincia varchar (20),  
  primary key(codigo)  
);
```

3- Ingrese algunos registros:

```
insert into clientes  
  values ('Lopez Marcos','Colon 111','Cordoba','Cordoba');  
insert into clientes  
  values ('Perez Ana','San Martin 222','Cruz del Eje','Cordoba');  
insert into clientes  
  values ('Garcia Juan','Rivadavia 333','Villa del Rosario','Cordoba');  
insert into clientes  
  values ('Perez Luis','Sarmiento 444','Rosario','Santa Fe');  
insert into clientes  
  values ('Pereyra Lucas','San Martin 555','Cruz del Eje','Cordoba');  
insert into clientes  
  values ('Gomez Ines','San Martin 666','Santa Fe','Santa Fe');  
insert into clientes  
  values ('Torres Fabiola','Alem 777','Villa del Rosario','Cordoba');  
insert into clientes  
  values ('Lopez Carlos',null,'Cruz del Eje','Cordoba');  
insert into clientes  
  values ('Ramos Betina','San Martin 999','Cordoba','Cordoba');  
insert into clientes  
  values ('Lopez Lucas','San Martin 1010','Posadas','Misiones');
```

4- Obtenga las provincias sin repetir (3 registros)

5- Cuento las distintas provincias.

6- Se necesitan los nombres de las ciudades sin repetir (6 registros)

7- Obtenga la cantidad de ciudades distintas.

8- Combine con "where" para obtener las distintas ciudades de la provincia de Cordoba (3 registros)

9- Contamos las distintas ciudades de cada provincia empleando "group by" (3 registros)

Segundo problema:

La provincia almacena en una tabla llamada "inmuebles" los siguientes datos de los inmuebles y sus

propietarios para cobrar impuestos:

1- Elimine la tabla si existe:

```
if object_id('inmuebles') is not null
drop table inmuebles;
```

2- Créela con la siguiente estructura:

```
create table inmuebles (
  documento varchar(8) not null,
  apellido varchar(30),
  nombre varchar(30),
  domicilio varchar(20),
  barrio varchar(20),
  ciudad varchar(20),
  tipo char(1),--b=baldio, e: edificado
  superficie decimal (8,2)
);
```

3- Ingrese algunos registros:

```
insert into inmuebles
values ('11000000','Perez','Alberto','San Martin 800','Centro','Cordoba','e',100);
insert into inmuebles
values ('11000000','Perez','Alberto','Sarmiento 245','Gral. Paz','Cordoba','e',200);
insert into inmuebles
values ('12222222','Lopez','Maria','San Martin 202','Centro','Cordoba','e',250);
insert into inmuebles
values ('13333333','Garcia','Carlos','Paso 1234','Alberdi','Cordoba','b',200);
insert into inmuebles
values ('13333333','Garcia','Carlos','Guemes 876','Alberdi','Cordoba','b',300);
insert into inmuebles
values ('14444444','Perez','Mariana','Caseros 456','Flores','Cordoba','b',200);
insert into inmuebles
values ('15555555','Lopez','Luis','San Martin 321','Centro','Carlos Paz','e',500);
insert into inmuebles
values ('15555555','Lopez','Luis','Lopez y Planes 853','Flores','Carlos Paz','e',350);
insert into inmuebles
values ('16666666','Perez','Alberto','Sucre 1877','Flores','Cordoba','e',150);
```

4- Muestre los distintos apellidos de los propietarios, sin repetir (3 registros)

5- Muestre los distintos documentos de los propietarios, sin repetir (6 registros)

6- Cuente, sin repetir, la cantidad de propietarios de inmuebles de la ciudad de Cordoba (5)

7- Cuente la cantidad de inmuebles con domicilio en 'San Martin', sin repetir la ciudad (2)

8- Muestre los apellidos y nombres, sin repetir (5 registros)

Note que hay 2 personas con igual nombre y apellido que aparece una sola vez.

9- Muestre la cantidad de inmuebles que tiene cada propietario agrupando por documento, sin repetir barrio (6 registros)

44 - Cláusula top

Primer problema:

Una empresa tiene registrados sus empleados en una tabla llamada "empleados".

1- Elimine la tabla si existe:

```
if object_id('empleados') is not null  
drop table empleados;
```

2- Créela con la siguiente estructura:

```
create table empleados (  
    documento varchar(8) not null,  
    nombre varchar(30),  
    estadocivil char(1),--c=casado, s=soltero,v=viudo  
    seccion varchar(20)  
);
```

3- Ingrese algunos registros:

```
insert into empleados  
values ('22222222','Alberto Lopez','c','Sistemas');  
insert into empleados  
values ('23333333','Beatriz Garcia','c','Administracion');  
insert into empleados  
values ('24444444','Carlos Fuentes','s','Administracion');  
insert into empleados  
values ('25555555','Daniel Garcia','s','Sistemas');  
insert into empleados  
values ('26666666','Ester Juarez','c','Sistemas');  
insert into empleados  
values ('27777777','Fabian Torres','s','Sistemas');  
insert into empleados  
values ('28888888','Gabriela Lopez',null,'Sistemas');  
insert into empleados  
values ('29999999','Hector Garcia',null,'Administracion');
```

4- Muestre los 5 primeros registros (5 registros)

5- Muestre nombre y seccion de los 4 primeros registros ordenados por sección (4 registros)

6- Realice la misma consulta anterior pero incluya todos los registros que tengan el mismo valor en "seccion" que el último (8 registros)

7- Muestre nombre, estado civil y seccion de los primeros 4 empleados ordenados por estado civil y sección (4 registros)

8- Realice la misma consulta anterior pero incluya todos los valores iguales al último registro retornado (5 registros)

45 - Clave primaria compuesta

Primer problema:

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:

```
if object_id('consultas') is not null
drop table consultas;
```

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, etc.).
- ```
);
```

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.

### Segundo problema:

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

1- Elimine la tabla "inscriptos" si existe:

```
if object_id('inscriptos') is not null
drop table inscriptos;
```

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 inscripcion: datetime,
- 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 en distintos años. Cree la tabla con una clave

compuesta:

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

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

```
insert into inscriptos
 values ('12222222','Juan Perez','tenis','2005','s');
insert into inscriptos
 values ('23333333','Marta Garcia','tenis','2005','s');
insert into inscriptos
 values ('34444444','Luis Perez','tenis','2005','n');
```

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

```
insert into inscriptos
 values ('12222222','Juan Perez','futbol','2005','s');
insert into inscriptos
 values ('12222222','Juan Perez','natacion','2005','s');
insert into inscriptos
 values ('12222222','Juan Perez','basquet','2005','n');
```

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

```
insert into inscriptos
 values ('12222222','Juan Perez','tenis','2006','s');
insert into inscriptos
 values ('12222222','Juan Perez','tenis','2007','s');
```

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 actualizar un registro para que la clave primaria se repita.

## 47 - Restricción default

### Primer 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:

```
if object_id('visitantes') is not null
drop table visitantes;
```

2- Cree la tabla con la siguiente estructura:

```
create table visitantes(
numero int identity,
nombre varchar(30),
edad tinyint,
domicilio varchar(30),
ciudad varchar(20),
montocompra decimal (6,2) not null
);
```

3- Defina una restricción "default" para el campo "ciudad" que almacene el valor "Cordoba" en caso de no ingresar valor para dicho campo:

```
alter table visitantes
add constraint DF_visitantes_ciudad
default 'Cordoba'
for ciudad;
```

4- Defina una restricción "default" para el campo "montocompra" que almacene el valor "0" en caso de no ingresar valor para dicho campo:

```
alter table visitantes
add constraint DF_visitantes_montocompra
default 0
for montocompra;
```

5- Ingrese algunos registros sin valor para los campos con restricción "default":

```
insert into visitantes
values ('Susana Molina',35,'Colon 123',default,59.80);
insert into visitantes (nombre,edad,domicilio)
values ('Marcos Torres',29,'Carlos Paz');
insert into visitantes
values ('Mariana Juarez',45,'Carlos Paz',null,23.90);
```

6- Vea cómo se almacenaron los registros:

```
select * from visitantes;
```

7- Vea las restricciones creadas anteriormente. aparecen dos filas, una por cada restricción.

8- Intente agregar otra restricción "default" al campo "ciudad".

Aparece un mensaje de error indicando que el campo ya tiene una restricción "default" y sabemos que no puede establecerse más de una restricción "default" por campo.

9- Intente establecer una restricción "default" al campo "identity".

No se permite.

**Segundo problema:**

Una playa de estacionamiento almacena cada día los datos de los vehículos que ingresan en la tabla llamada "vehiculos".

1- Elimine la tabla, si existe:

```
if object_id('vehiculos') is not null
drop table vehiculos;
```

2- Cree la tabla:

```
create table vehiculos(
patente char(6) not null,
tipo char(1),--'a'=auto, 'm'=moto
horallegada datetime,
horasalida datetime
);
```

3- Establezca una restricción "default" para el campo "tipo" que almacene el valor "a" en caso de no ingresarse valor para dicho campo.

4- Ingrese un registro sin valor para el campo "tipo":

```
insert into vehiculos values('BVB111',default,default,null);
```

5- Recupere los registros:

```
select * from vehiculos;
```

6- Intente establecer otra restricción "default" para el campo "tipo" que almacene el valor "m" en caso de no ingresarse valor para dicho campo.

No lo permite porque un campo solamente admite una restricción "default" y ya tiene una.

7- Establezca una restricción "default" para el campo "horallegada" que almacene la fecha y hora del sistema.

8- Ingrese un registro sin valor para los campos de tipo datetime.

9- Recupere los registros:

```
select * from vehiculos;
```

10- Vea las restricciones.

2 restricciones.

## 48 - Restricción check

### Primer problema:

Una empresa tiene registrados datos de sus empleados en una tabla llamada "empleados".

1- Elimine la tabla si existe:

```
if object_id('empleados') is not null
drop table empleados;
```



2- Créela con la siguiente estructura:

```
create table empleados (
 documento varchar(8),
 nombre varchar(30),
 fechanacimiento datetime,
 cantidadhijos tinyint,
 seccion varchar(20),
 sueldo decimal(6,2)
);
```

3- Agregue una restricción "check" para asegurarse que no se ingresen valores negativos para el sueldo:

```
alter table empleados
 add constraint CK_empleados_sueldo_positivo
 check (sueldo>0);
```

4- Ingrese algunos registros válidos:

```
insert into empleados values ('22222222','Alberto
Lopez','1965/10/05',1,'Sistemas',1000);
insert into empleados values ('33333333','Beatriz
Garcia','1972/08/15',2,'Administracion',3000);
insert into empleados values ('34444444','Carlos
Caseres','1980/10/05',0,'Contaduría',6000);
```

5- Intente agregar otra restricción "check" al campo sueldo para asegurar que ninguno supere el valor 5000:

```
alter table empleados
 add constraint CK_empleados_sueldo_maximo
 check (sueldo<=5000);
```

La sentencia no se ejecuta porque hay un sueldo que no cumple la restricción.

6- Elimine el registro infractor y vuelva a crear la restricción:

```
delete from empleados where sueldo=6000;
```

```
alter table empleados
 add constraint CK_empleados_sueldo_maximo
 check (sueldo<=5000);
```

7- Establezca una restricción para controlar que la fecha de nacimiento que se ingresa no supere la fecha actual:

```
alter table empleados
 add constraint CK_fechanacimiento_actual
 check (fechanacimiento<getdate());
```

8- Establezca una restricción "check" para "seccion" que permita solamente los valores "Sistemas",

"Administracion" y "Contaduría":

```
alter table empleados
 add constraint CK_empleados_seccion_lista
```

```
check (seccion in ('Sistemas','Administracion','Contaduria'));
```

9- Establezca una restricción "check" para "cantidadhijos" que permita solamente valores entre 0 y 15.

10- Vea todas las restricciones de la tabla (5 filas):  
exec sp\_helpconstraint empleados;

11- Intente agregar un registro que vaya contra alguna de las restricciones al campo "sueldo".  
Mensaje de error porque se infringe la restricción "CK\_empleados\_sueldo\_positivo".

12- Intente agregar un registro con fecha de nacimiento futura.  
Mensaje de error.

13- Intente modificar un registro colocando en "cantidadhijos" el valor "21".  
Mensaje de error.

14- Intente modificar el valor de algún registro en el campo "seccion" cambiándolo por uno que no esté incluido en la lista de permitidos.  
Mensaje de error.

15- Intente agregar una restricción al campo sección para aceptar solamente valores que comiencen con la letra "B":

```
alter table empleados
add constraint CK_seccion_letrainicial
check (seccion like '%B');
```

Note que NO se puede establecer esta restricción porque va en contra de la establecida anteriormente para el mismo campo, si lo permitiera, no podríamos ingresar ningún valor para "seccion".

### **Segundo problema:**

Una playa de estacionamiento almacena los datos de los vehículos que ingresan en la tabla llamada "vehiculos".

1- Elimine la tabla, si existe:  
if object\_id('vehiculos') is not null  
drop table vehiculos;

2- Cree la tabla:  
create table vehiculos(  
numero int identity,  
patente char(6),  
tipo char(4),  
fechahoraentrada datetime,  
fechahorasalida datetime  
);

3- Ingresamos algunos registros:

```
insert into vehiculos values('AIC124','auto','2007/01/17 8:05','2007/01/17 12:30');
insert into vehiculos values('CAA258','auto','2007/01/17 8:10',null);
insert into vehiculos values('DSE367','moto','2007/01/17 8:30','2007/01/17 18:00');
```

4- Agregue una restricción "check" que especifique un patrón de 3 letras y 3 dígitos para "patente":

```
alter table vehiculos
add constraint CK_vehiculos_patente_patron
check (patente like '[A-Z][A-Z][A-Z][0-9][0-9][0-9]');
```

5- Intente ingresar un registro con un valor inapropiado para "patente":

```
insert into vehiculos values('AB1234','auto',getdate(),null);
```

No lo permite.

6- Agregue una restricción "check" que especifique que el campo "tipo" acepte solamente los valores

"auto" y "moto":

```
alter table vehiculos
add constraint CK_vehiculos_tipo_valores
check (tipo in ('auto','moto'));
```

7- Intente modificar el valor del campo "tipo" ingresando un valor inexistente en la lista de

valores permitidos por la restricción establecida a dicho campo:

```
update vehiculos set tipo='bici' where patente='AIC124';
```

No lo permite.

8- Agregue una restricción "default" para el campo "tipo" que almacene el valor "bici":

```
alter table vehiculos
add constraint DF_vehiculos_tipo
default 'bici'
for tipo;
```

Lo acepta. Pero, note que va en contra de la restricción "check" impuesta en el punto 6.

9- Intente ingresar un registro sin valor para "tipo":

```
insert into vehiculos values('SDF134',default,null,null);
```

No lo permite porque va contra la restricción "check" del campo.

10- Agregue una restricción "check" para asegurarse que la fecha de entrada a la playa no sea

posterior a la fecha y hora actual:

```
alter table vehiculos
add constraint CK_vehiculos_fechahoraentrada_actual
check (fechahoraentrada<=getdate());
```

11- Agregue otra restricción "check" al campo "fechahoraentrada" que establezca que sus valores no

sean posteriores a "fechahorasalida":

```
alter table vehiculos
add constraint CK_vehiculos_fechahoraentradasalida
check (fechahoraentrada<=fechahorasalida);
```

12- Intente ingresar un valor que no cumpla con la primera restricción establecida en el campo

"fechahoraentrada":

```
insert into vehiculos values('ABC123','auto','2007/05/05 10:10',null);
```

La inserción no se realiza.

13- Intente modificar un registro para que la salida sea anterior a la entrada:

```
update vehiculos set fechahorasalida='2007/01/17 7:30'
```

```
where patente='CAA258';
```

Mensaje de error.

14- Vea todas las restricciones para la tabla "vehiculos":

```
exec sp_helpconstraint vehiculos;
```

aparecen 5 filas, 4 correspondientes a restricciones "check" y 1 a "default".

15- Establezca una restricción "default" para el campo "fechahoraentrada" para que almacene la fecha

actual del sistema:

```
alter table vehiculos
```

```
add constraint DF_vehiculos_fechahoraentrada
```

```
default getdate()
```

```
for fechahoraentrada;
```

16- Ingrese un registro sin valor para "fechahoraentrada":

```
insert into vehiculos values('DFR156','moto',default,default);
```

17- Vea todos los registros:

```
select * from vehiculos;
```

18- Vea las restricciones:

```
exec sp_helpconstraint vehiculos;
```

4 restricciones "check" y 2 "default".

## 49 - Deshabilitar restricciones (with check - nocheck)

### Primer problema:

Una empresa tiene registrados datos de sus empleados en una tabla llamada "empleados".

1- Elimine la tabla (si existe):

```
if object_id('empleados') is not null
```

```
drop table empleados;
```

2- Créela con la siguiente estructura e ingrese los registros siguientes:

```
create table empleados (
```

```
documento varchar(8),
```

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

```
seccion varchar(20),
```

```
sueldo decimal(6,2)
```

```
);
```

```
insert into empleados
```

```
values ('22222222','Alberto Acosta','Sistemas',-10);
insert into empleados
values ('33333333','Beatriz Benitez','Recursos',3000);
insert into empleados
values ('34444444','Carlos Caseres','Contaduria',4000);
```

3- Intente agregar una restricción "check" para asegurarse que no se ingresen valores negativos para el sueldo:

```
alter table empleados
add constraint CK_empleados_sueldo_positivo
check (sueldo>=0);
```

No se permite porque hay un valor negativo almacenado.

5- Vuelva a intentarlo agregando la opción "with nocheck":

```
alter table empleados
with nocheck
add constraint CK_empleados_sueldo_positivo
check (sueldo>=0);
```

6- Intente ingresar un valor negativo para sueldo:

```
insert into empleados
values ('35555555','Daniel Duarte','Administracion',-2000);
```

No es posible a causa de la restricción.

7- Deshabilite la restricción e ingrese el registro anterior:

```
alter table empleados
nocheck constraint CK_empleados_sueldo_positivo;
insert into empleados
values ('35555555','Daniel Duarte','Administracion',2000);
```

8- Establezca una restricción "check" para "seccion" que permita solamente los valores "Sistemas", "Administracion" y "Contaduría":

```
alter table empleados
add constraint CK_empleados_seccion_lista
check (seccion in ('Sistemas','Administracion','Contaduria'));

```

No lo permite porque existe un valor fuera de la lista.

9- Establezca la restricción anterior evitando que se controlen los datos existentes.

10- Vea si las restricciones de la tabla están o no habilitadas:

```
exec sp_helpconstraint empleados;
```

Muestra 2 filas, una por cada restricción.

11- Habilite la restricción deshabilitada.

12- Intente modificar la sección del empleado "Carlos Caseres" a "Recursos".

No lo permite.

13- Deshabilite la restricción para poder realizar la actualización del punto precedente.

## 50 - Restricción primary key

### Primer problema:

Una empresa tiene registrados datos de sus empleados en una tabla llamada "empleados".

1- Elimine la tabla si existe:

```
if object_id('empleados') is not null
drop table empleados;
```

2- Créela con la siguiente estructura:

```
create table empleados (
 documento varchar(8) not null,
 nombre varchar(30),
 seccion varchar(20)
);
```

3- Ingrese algunos registros, dos de ellos con el mismo número de documento:

```
insert into empleados
values ('22222222','Alberto Lopez','Sistemas');
insert into empleados
values ('23333333','Beatriz Garcia','Administracion');
insert into empleados
values ('23333333','Carlos Fuentes','Administracion');
```

4- Intente establecer una restricción "primary key" para la tabla para que el documento no se repita

ni admita valores nulos:

```
alter table empleados
add constraint PK_empleados_documento
primary key(documento);
```

No lo permite porque la tabla contiene datos que no cumplen con la restricción, debemos eliminar (o

modificar) el registro que tiene documento duplicado:

```
delete from empleados
where nombre='Carlos Fuentes';
```

5- Establezca la restricción "primary key" del punto 4.

6- Intente actualizar un documento para que se repita.

No lo permite porque va contra la restricción.

7-Intente establecer otra restricción "primary key" con el campo "nombre".

No lo permite, sólo puede haber una restricción "primary key" por tabla.

8- Intente ingresar un registro con valor nulo para el documento.

No lo permite porque la restricción no admite valores nulos.

9- Establezca una restricción "default" para que almacene "00000000" en el documento en caso de omitirlo en un "insert".

10- Ingrese un registro sin valor para el documento.

11- Vea el registro:

```
select * from empleados;
```

12- Intente ingresar otro empleado sin documento explícito.

No lo permite porque se duplicaría la clave.

13- Vea las restricciones de la tabla empleados (2 filas):

```
exec sp_helpconstraint empleados;
```

### **Segundo problema:**

Una empresa de remises tiene registrada la información de sus vehículos en una tabla llamada

"remis".

1- Elimine la tabla si existe:

```
if object_id('remis') is not null
drop table remis;
```

2- Cree la tabla con la siguiente estructura:

```
create table remis(
numero tinyint identity,
patente char(6),
marca varchar(15),
modelo char(4)
);
```

3- Ingrese algunos registros sin repetir patente:

```
insert into remis values('ABC123','Renault 12','1990');
insert into remis values('DEF456','Fiat Duna','1995');
```

4- Intente definir una restricción "primary key" para el campo "patente".

No lo permite porque el campo no fue definido "not null".

5- Establezca una restricción "primary key" para el campo "numero".

Si bien "numero" no fue definido explícitamente "not null", no acepta valores nulos por ser

"identity".

6- Vea la información de las restricciones (2 filas):

```
exec sp_helpconstraint remis;
```