Las claves primarias pueden ser simples, formadas por un solo campo o compuestas, más de un campo.

Recordemos que una clave primaria identifica 1 solo registro en una tabla.

Para un valor del campo clave existe solamente 1 registro. Los valores no se repiten ni pueden ser nulos.

Existe una playa de estacionamiento que almacena cada día los datos de los vehículos que ingresan en la tabla llamada "vehiculos" con los siguientes campos:

- patente char(6) not null,

- tipo char (1), 'a'= auto, 'm'=moto,

- horallegada datetime,

- horasalida datetime,

Necesitamos definir una clave primaria para una tabla con los datos descriptos arriba. No podemos usar solamente la patente porque un mismo auto puede ingresar más de una vez en el día a la playa; tampoco podemos usar la hora de entrada porque varios autos pueden ingresar a una misma hora.  
Tampoco sirven los otros campos.

Como ningún campo, por si sólo cumple con la condición para ser clave, es decir, debe identificar un solo registro, el valor no puede repetirse, debemos usar 2 campos.

Definimos una clave compuesta cuando ningún campo por si solo cumple con la condición para ser clave.

En este ejemplo, un auto puede ingresar varias veces en un día a la playa, pero siempre será a distinta hora.

Usamos 2 campos como clave, la patente junto con la hora de llegada, así identificamos unívocamente cada registro.

Para establecer más de un campo como clave primaria usamos la siguiente sintaxis:

create table vehiculos(

patente char(6) not null,

tipo char(1),--'a'=auto, 'm'=moto

horallegada datetime,

horasalida datetime,

primary key(patente,horallegada)

);

Nombramos los campos que formarán parte de la clave separados por comas.

Al ingresar los registros, SQL Server controla que los valores para los campos establecidos como clave primaria no estén repetidos en la tabla; si estuviesen repetidos, muestra un mensaje y la inserción no se realiza. Lo mismo sucede si realizamos una actualización.

Entonces, si un solo campo no identifica unívocamente un registro podemos definir una clave primaria compuesta, es decir formada por más de un campo.

**Primer 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- Elimine las siguientes reglas:

if object\_id ('RG\_patente\_patron') is not null

drop rule RG\_patente\_patron;

if object\_id ('RG\_horallegada') is not null

drop rule RG\_horallegada;

if object\_id ('RG\_vehiculos\_tipo') is not null

drop rule RG\_vehiculos\_tipo;

if object\_id ('RG\_vehiculos\_tipo2') is not null

drop rule RG\_vehiculos\_tipo2;

if object\_id ('RG\_menor\_fechaactual') is not null

drop rule RG\_menor\_fechaactual;

3- Cree la tabla:

create table vehiculos(

patente char(6) not null,

tipo char(1),--'a'=auto, 'm'=moto

horallegada datetime not null,

horasalida datetime

);

4- Ingrese algunos registros:

insert into vehiculos values ('AAA111','a','1990-02-01 08:10',null);

insert into vehiculos values ('BCD222','m','1990-02-01 08:10','1990-02-01 10:10');

insert into vehiculos values ('BCD222','m','1990-02-01 12:00',null);

insert into vehiculos values ('CC1234','a','1990-02-01 12:00',null);

5- Cree una regla para restringir los valores que se pueden ingresar en un campo "patente" (3 letras

seguidas de 3 dígitos):

create rule RG\_patente\_patron

as @patente like '[A-Z][A-Z][A-Z][0-9][0-9][0-9]'

6- Ejecute el procedimiento almacenado del sistema "sp\_help" para ver que la regla creada

anteriormente existe:

sp\_help;

7- Ejecute el procedimiento almacenado del sistema "sp\_helpconstraint" para ver que la regla creada

anteriormente no está asociada aún a ningún campo de la tabla "vehiculos".

8- Asocie la regla al campo "patente":

Note que hay una patente que no cumple la regla, SQL Server NO controla los datos existentes, pero

si controla las inserciones y actualizaciones:

select \*from empleados;

9- Intente ingresar un registro con valor para el campo "patente" que no cumpla con la regla.

aparece un mensaje de error indicando que hay conflicto con la regla y la inserción no se realiza.

10- Cree otra regla que controle los valores para el campo "tipo" para que solamente puedan

ingresarse los caracteres "a" y "m".

11- Asocie la regla al campo "tipo".

12- Intente actualizar un registro cambiando el valor de "tipo" a un valor que no cumpla con la

regla anterior.

No lo permite.

13- Cree otra regla llamada "RG\_vehiculos\_tipo2" que controle los valores para el campo "tipo" para

que solamente puedan ingresarse los caracteres "a", "c" y "m".

14- Si la asociamos a un campo que ya tiene asociada otra regla, la nueva regla reeemplaza la

asociación anterior. Asocie la regla creada en el punto anterior al campo "tipo".

15- Actualice el registro que no pudo actualizar en el punto 12:

update vehiculos set tipo='c' where patente='AAA111';

16- Cree una regla que permita fechas menores o iguales a la actual.

17- Asocie la regla anterior a los campos "horallegada" y "horasalida":

exec sp\_bindrule RG\_menor\_fechaactual, 'vehiculos.horallegada';

exec sp\_bindrule RG\_menor\_fechaactual, 'vehiculos.horasalida';

18- Ingrese un registro en el cual la hora de entrada sea posterior a la hora de salida:

insert into vehiculos values ('NOP555','a','1990-02-01 10:10','1990-02-01 08:30');

19- Intente establecer una restricción "check" que asegure que la fecha y hora de llegada a la playa

no sea posterior a la fecha y hora de salida:

alter table vehiculos

add constraint CK\_vehiculos\_llegada\_salida

check(horallegada<=horasalida);

No lo permite porque hay un registro que no cumple la restricción.

20- Elimine dicho registro:

delete from vehiculos where patente='NOP555';

21- Establezca la restricción "check" que no pudo establecer en el punto 19:

alter table vehiculos

add constraint CK\_vehiculos\_llegada\_salida

check(horallegada<=horasalida);

22- Cree una restricción "default" que almacene el valor "b" en el campo "tipo:

alter table vehiculos

add constraint DF\_vehiculos\_tipo

default 'b'

for tipo;

Note que esta restricción va contra la regla asociada al campo "tipo" que solamente permite los

valores "a", "c" y "m". SQL Server no informa el conflicto hasta que no intenta ingresar el valor

por defecto.

23- Intente ingresar un registro con el valor por defecto para el campo "tipo":

insert into vehiculos values ('STU456',default,'1990-02-01 10:10','1990-02-01 15:30');

No lo permite porque va contra la regla asociada al campo "tipo".

24- Vea las reglas asociadas a "empleados" y las restricciones aplicadas a la misma tabla ejecutando

"sp\_helpconstraint".

Muestra 1 restricción "check", 1 restricción "default" y 4 reglas asociadas.

**SOLUCION**

if object\_id('vehiculos') is not null

drop table vehiculos;

if object\_id ('RG\_patente\_patron') is not null

drop rule RG\_patente\_patron;

if object\_id ('RG\_horallegada') is not null

drop rule RG\_horallegada;

if object\_id ('RG\_vehiculos\_tipo') is not null

drop rule RG\_vehiculos\_tipo;

if object\_id ('RG\_vehiculos\_tipo2') is not null

drop rule RG\_vehiculos\_tipo2;

if object\_id ('RG\_menor\_fechaactual') is not null

drop rule RG\_menor\_fechaactual;

create table vehiculos(

patente char(6) not null,

tipo char(1),--'a'=auto, 'm'=moto

horallegada datetime not null,

horasalida datetime

);

insert into vehiculos values ('AAA111','a','1990-02-01 08:10',null);

insert into vehiculos values ('BCD222','m','1990-02-01 08:10','1990-02-01 10:10');

insert into vehiculos values ('BCD222','m','1990-02-01 12:00',null);

insert into vehiculos values ('CC1234','a','1990-02-01 12:00',null);

create rule RG\_patente\_patron

as @patente like '[A-Z][A-Z][A-Z][0-9][0-9][0-9]';

sp\_help;

sp\_helpconstraint vehiculos;

sp\_bindrule RG\_patente\_patron,'vehiculos.patente';

select \*from empleados;

insert into vehiculos values ('FGHIJK','a','1990-02-01 18:00',null);

create rule RG\_vehiculos\_tipo

as @tipo in ('a','m');

sp\_bindrule RG\_vehiculos\_tipo, 'vehiculos.tipo';

update vehiculos set tipo='c' where patente='AAA111';

create rule RG\_vehiculos\_tipo2

as @tipo in ('a','c','m');

sp\_bindrule RG\_vehiculos\_tipo2, 'vehiculos.tipo';

update vehiculos set tipo='c' where patente='AAA111';

create rule RG\_menor\_fechaactual

as @fecha <= getdate();

exec sp\_bindrule RG\_menor\_fechaactual, 'vehiculos.horallegada';

exec sp\_bindrule RG\_menor\_fechaactual, 'vehiculos.horasalida';

insert into vehiculos values ('NOP555','a','1990-02-01 10:10','1990-02-01 08:30');

alter table vehiculos

add constraint CK\_vehiculos\_llegada\_salida

check(horallegada<=horasalida);

delete from vehiculos where patente='NOP555';

alter table vehiculos

add constraint CK\_vehiculos\_llegada\_salida

check(horallegada<=horasalida);

alter table vehiculos

add constraint DF\_vehiculos\_tipo

default 'b'

for tipo;

insert into vehiculos values ('STU456',default,'1990-02-01 10:10','1990-02-01 15:30');

sp\_helpconstraint vehiculos;