

Administración SQL

Básica

Primera parte

Restricciones

Las restricciones (constraints) en SQL nos permiten definir obligaciones que debe cumplir la información para poder almacenarse en la Base de Datos.

Pueden ser de varios tipos:

NOT NULL - Indica que la columna no puede contener el valor NULL.

UNIQUE - Asegura que la columna o conjunto de columnas sólo puede tener un valor único, cada valor sólo puede aparecer una vez en la tabla

PRIMARY KEY - Asegura que la columna o conjunto de columnas sólo puede tener un valor único, cada valor sólo puede aparecer una vez en la tabla, además de no poder ser NULL y servir de identificador para cada fila de información de la tabla.

FOREIGN KEY - Asegura que determinada columna provenga de los valores ya almacenados en otro campo (normalmente de otra tabla)

CHECK - Asegura que los valores de la tabla cumplan una regla condicional o calculada.

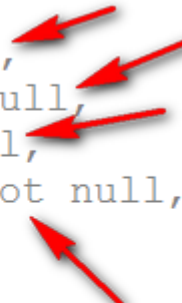
DEFAULT - Especifica el valor que tendrá la columna cuando no lo especifiquemos al grabar información.

Constraint en la creación de una tabla;

```
Create table nombre (  
...,  
campo tipo especificar_constraint,  
...  
)
```

Ejemplo:

```
create table ficha(  
  dni varchar(8) primary key,  
  apellido varchar(50) not null,  
  nombre varchar(50) not null,  
  fechanacimiento datetime not null,  
  fechaingreso datetime,  
  tipocliente char(1),  
  Direccion varchar(100)  
)  
go
```



En las características del campo añadimos las de la restricción, como el hacerlo primary key o el atributo not null, para evitar que se pueda dejar sin valor.


Otra forma de especificar una constraint en la creación de una tabla es especificada como si fuera un campo. Tendremos que haber definido previamente el campo o campos que se ven afectados por ella.

Normalmente se realiza para constraints que afectan a varios campos o a las que queremos darle nombre para poder hacer referencia posteriormente a ellas.

No es aplicable al not null.

```
Create table nombre (  
    ...,  
    campo tipo,  
    ...,  
    constraint nombre-constraint especificar_constraint  
)
```

```
create table ficha(  
dni varchar(8),  
apellido varchar(50) not null,  
nombre varchar(50) not null,  
fechanacimiento datetime not null,  
fechaingreso datetime,  
tipocliente char(1),  
Direccion varchar(100),  
constraint pk_ficha primary key(dni)  
)  
go
```



También podemos añadir una constraint tras la creación de la tabla.

```
alter table nombre_tabla  
add constraint nombre-constraint  
    especificar_constraint;
```

o eliminarla

```
alter table nombre_tabla  
drop constraint nombre-constraint;
```



```
create table ficha(  
dni varchar(8),  
apellido varchar(50) not null,  
nombre varchar(50) not null,  
fechanacimiento datetime not null,  
fechaingreso datetime,  
tipocliente char(1),  
Direccion varchar(100)  
)  
go  
  
alter table ficha  
ALTER COLUMN dni varchar(8) NOT NULL;  
go  
  
alter table ficha  
add constraint pk_ficha  
primary key(dni);  
go
```

Exige que el campo sea Not null
para poder ponerlo como primary
key.

```
]exec sp_helpconstraint ficha;|  
go
```

| Resultados | | Mensajes | | | | | |
|-------------|-------------------------|-----------------|---------------|---------------|----------------|------------------------|-----------------|
| Object Name | | | | | | | |
| 1 | ficha | | | | | | |
| | constraint_type | constraint_name | delete_action | update_action | status_enabled | status_for_replication | constraint_keys |
| 1 | PRIMARY KEY (clustered) | pk_ficha | (n/a) | (n/a) | (n/a) | (n/a) | dni |

```
]alter table ficha  
-drop constraint pk_ficha;  
go
```


Eliminamos la restricción a partir de su nombre

Clave primaria compuesta

Una clave primaria puede estar definida por varios campos.
Esto lo haremos en la creación de la tabla.

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

create table ficha(
dni varchar(8) not null,|
apellido varchar(50)not null,
nombre varchar(50) not null,
fechanacimiento datetime not null,
fechaingreso datetime,
tipocliente char(1),
Direccion varchar(100),
constraint pk_ficha primary key(apellido, nombre)
)
go
```




También mediante alter table.

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

```
create table ficha(  
dni varchar(8) not null,  
apellido varchar(50) not null,  
nombre varchar(50) not null,  
fechanacimiento datetime not null,  
fechaingreso datetime,  
tipocliente char(1),  
Direccion varchar(100)  
)  
go
```

```
alter table ficha|  
add constraint pk_ficha  
primary key(apellido, nombre);  
go
```



Integridad de los datos

La exigencia de integridad de los datos garantiza la calidad de los datos de la base de datos. Por ejemplo, si se especifica para un empleado el valor de identificador de **123**, la base de datos no debe permitir que ningún otro empleado tenga el mismo valor de identificador. Si tiene una columna **employee_rating** para la que se prevean valores entre **1** y **5**, la base de datos no debe aceptar valores fuera de ese intervalo. Si en la tabla hay una columna **dept_id** en la que se almacena el número de departamento del empleado, la base de datos sólo debe permitir valores que correspondan a los números de departamento de la empresa. Dos pasos importantes en el diseño de las tablas son la identificación de valores válidos para una columna y la determinación de cómo forzar la integridad de los datos en la columna.

La integridad de datos pertenece a una de las siguientes categorías:

- **Integridad de entidad**
- **Integridad de dominio**
- **Integridad referencial**
- **Integridad definida por el usuario**

Integridad de entidad

La integridad de entidad define una fila como entidad única para una tabla determinada. La integridad de entidad exige la integridad de las columnas de los identificadores o la clave principal de una tabla, mediante índices y restricciones UNIQUE, o restricciones PRIMARY KEY.

Integridad de dominio

La integridad de dominio viene dada por la validez de las entradas para una columna determinada. Puede exigir la integridad de dominio para restringir el tipo mediante tipos de datos, el formato mediante reglas y restricciones CHECK, o el intervalo de valores posibles mediante restricciones FOREIGN KEY, restricciones CHECK, definiciones DEFAULT, definiciones NOT NULL y reglas.

Integridad referencial

La integridad referencial protege las relaciones definidas entre las tablas cuando se crean o se eliminan filas. En SQL Server la integridad referencial se basa en las relaciones entre claves externas y claves principales o entre claves externas y claves exclusivas, mediante restricciones FOREIGN KEY y CHECK. La integridad referencial garantiza que los valores de clave sean coherentes en las distintas tablas. Para conseguir esa coherencia, es preciso que no haya referencias a valores inexistentes y que, si cambia el valor de una clave, todas las referencias a ella se cambien en consecuencia en toda la base de datos.

Cuando se exige la integridad referencial, SQL Server impide a los usuarios:

- Agregar o cambiar filas en una tabla relacionada si no hay ninguna fila asociada en la tabla principal.
- Cambiar valores en una tabla principal que crea filas huérfanas en una tabla relacionada.
- Eliminar filas de una tabla principal cuando hay filas relacionadas coincidentes.

Por ejemplo, en las tablas **Sales.SalesOrderDetail** y **Production.Product** de la base de datos

AdventureWorks2008R2, la integridad referencial se basa en la relación entre la clave externa (**ProductID**) de la tabla **Sales.SalesOrderDetail** y la clave principal (**ProductID**) de la tabla **Production.Product**. Esta relación garantiza que un pedido de ventas no pueda nunca hacer referencia a un producto que no existe en la tabla **Production.Product**

Integridad definida por el usuario

La integridad definida por el usuario permite definir reglas de empresa específicas que no pertenecen a ninguna otra categoría de integridad. Todas las categorías de integridad admiten la integridad definida por el usuario. Esto incluye todas las restricciones de nivel de columna y nivel de tabla en CREATE TABLE, procedimientos almacenados y desencadenadores.

Restricción check

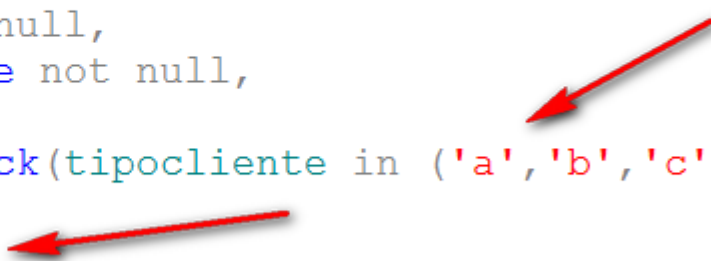
La restricción check permite validar los valores de un campo. Controla que no se puedan insertar valores no permitidos.

Tanto al final de la definición de un campo como añadiendo una constraint, el formato sería:

check (condición entre campos)

```
= if object_id('ficha') is not null
- drop table ficha;

= create table ficha(
dni varchar(8) primary key,
apellido varchar(50) not null,
nombre varchar(50) not null,
fechanacimiento datetime not null,
fechaingreso datetime,
tipocliente char(1) check(tipocliente in ('a', 'b', 'c')),
Direccion varchar(100),
constraint ck_fechas
    check(fechaingreso >= fechanacimiento)
)
go
```

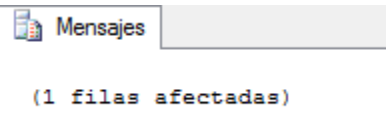


Tenemos dos restricciones check.

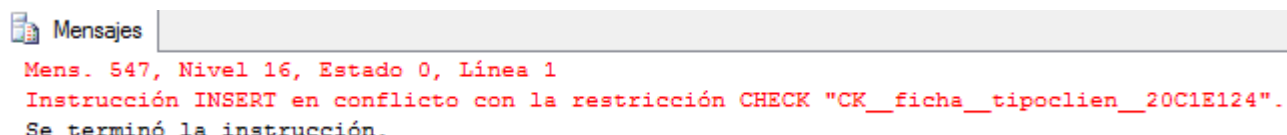
Una valida que los valores posibles del campo tipocliente sean sólo a, b o c.

La otra valida una relación entre las dos fechas, obligando a que fechaingreso sean menor o igual que fechanacimiento

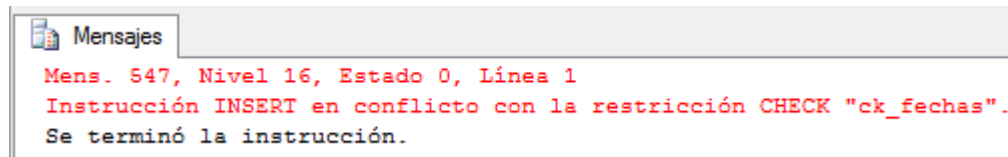
```
insert into ficha
values ('4444444W', 'Juan', 'Pérez', '10/01/1969', '01/01/2004',
       'a', 'C/primera nº 6');
```



```
insert into ficha
values ('4444445W', 'Juan', 'Hernández', '10/01/1969', '01/01/2004',
       'r', 'C/primera nº 6');
```



```
insert into ficha
values ('4444446Z', 'María', 'González', '10/01/1969', '01/01/1960',
       'a', 'C/primera nº 6');
```



Restricción única

Similar a la primary key.

Para exigir la ausencia de repeticiones en otro caso diferente a la primary key.

Su formato también es similar

Al final de un campo:

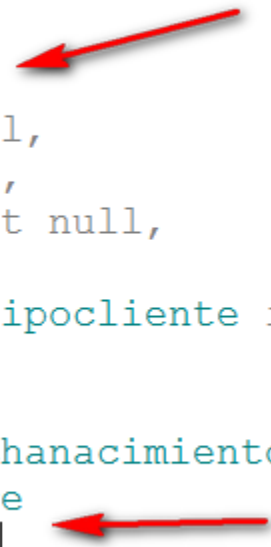
campo tipo unique

O como constraint

**constraint nombre_restriccion
unique(lista de campos)**

```
= if object_id('ficha') is not null
-   drop table ficha;

= create table ficha(
dni varchar(8) primary key,
codigosocio integer unique,
apellido varchar(50) not null,
nombre varchar(50) not null,
fechanacimiento datetime not null,
fechaingreso datetime,
tipocliente char(1) check(tipocliente in ('a','b','c')),
Direccion varchar(100),
constraint ck_fechas
    check(fechaingreso>=fechanacimiento),
constraint uq_apellidonombre
    unique(apellido,nombre)|
)
go
```




```
insert into ficha
values ('4444444W',123,'Juan','Pérez','10/01/1969','01/01/2004',
       'a','C/primera nº 6');
```

Mensajes

(1 filas afectadas)

```
insert into ficha
values ('4444445W',123,'Juan','Hernández','10/01/1969','01/01/2004',
       'b','C/primera nº 6');
```

Mensajes

Mens. 2627, Nivel 14, Estado 1, Línea 2

Infracción de la restricción UNIQUE KEY 'UQ__ficha__7955D0C231EC6D26'. No se puede insertar una clave duplicada
Se terminó la instrucción.

```
insert into ficha
values ('4444446Z',125,'Juan','Pérez','10/01/1969','01/01/1980',
       'a','C/primera nº 6');
```

Mensajes

Mens. 2627, Nivel 14, Estado 1, Línea 1

Infracción de la restricción UNIQUE KEY 'uq_apellidonombre'. No se puede insertar una clave duplicada
Se terminó la instrucción.

Deshabilitar restricciones

Como ya vimos en las restricciones foreign key, podemos deshabilitar las restricciones.

Para desactivarla tendremos que detectar el nombre primero y después aplicar...

```
alter table NOMBRETABLA  
nocheck constraint NOMBRERESTRICCION;
```

Para activarla de nuevo...

```
alter table NOMBRETABLA  
check constraint NOMBRERESTRICCION;
```