

## 14 - Otras características del atributo Identity

El atributo "identity" permite indicar el valor de inicio de la secuencia y el incremento, para ello usamos la siguiente sintaxis:

```
create table libros(  
  codigo int identity(100,2),  
  titulo varchar(20),  
  autor varchar(30),  
  precio float  
);
```

Los valores comenzarán en "100" y se incrementarán de 2 en 2; es decir, el primer registro ingresado tendrá el valor "100", los siguientes "102", "104", "106", etc.

La función "ident\_seed()" retorna el valor de inicio del campo "identity" de la tabla que nombramos:

```
select ident_seed('libros');
```

La función "ident\_incr()" retorna el valor de incremento del campo "identity" de la tabla nombrada:

```
select ident_incr('libros');
```

Hemos visto que en un campo declarado "identity" no puede ingresarse explícitamente un valor. Para permitir ingresar un valor en un campo de identidad se debe activar la opción "identity\_insert":

```
set identity_insert libros on;
```

Es decir, podemos ingresar valor en un campo "identity" seteando la opción "identity\_insert" en "on".

Cuando "identity\_insert" está en ON, las instrucciones "insert" deben explicitar un valor:

```
insert into libros (codigo,titulo)  
values (5,'Alicia en el pais de las maravillas');
```

Si no se coloca un valor para el campo de identidad, la sentencia no se ejecuta y aparece un mensaje de error:

```
insert into libros (titulo,autor, editorial)  
values ('Matematica estas ahi','Paenza','Paidos');
```

El atributo "identity" no implica unicidad, es decir, permite repetición de valores; por ello hay que tener cuidado al explicitar un valor porque se puede ingresar un valor repetido.

Para desactivar la opción "identity\_insert" tipeamos:

```
set identity_insert libros off;
```

### Servidor de SQL Server instalado en forma local.

Ingresemos el siguiente lote de comandos en el SQL Server Management Studio:

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

```
create table libros(  
  codigo int identity(100,2),  
  titulo varchar(20),  
  autor varchar(30),  
  precio float  
);
```

go

```
insert into libros (titulo,autor,precio)
  values('El aleph','Borges',23);
insert into libros (titulo,autor,precio)
  values('Uno','Richard Bach',18);
insert into libros (titulo,autor,precio)
  values('Aprenda PHP','Mario Molina',45.60);

select * from libros;

-- Para saber cuál es el valor de inicio del campo "identity" de la tabla "libros":
select ident_seed('libros');

-- Si intentamos ingresar un valor para el campo "codigo" (genera error):
insert into libros (codigo,titulo,autor,precio)
  values(106,'Martin Fierro','Jose Hernandez',25);

-- Para permitir ingresar un valor en un campo de identidad activamos la opción "identity_insert":
set identity_insert libros on;

-- Recordemos que si "identity_insert" está en ON, la instrucción "insert" DEBE explicitar un valor:
insert into libros (codigo,titulo,autor)
  values (100,'Matematica estas ahi','Paenza');

-- Note que ingresamos un valor de código que ya existe; esto está permitido porque
-- el atributo "identity" no implica unicidad.
insert into libros (codigo,titulo,autor)
  values (1,'Ilusiones','Richard Bach');

-- Si no se coloca un valor para el campo de identidad,
-- la sentencia no se ejecuta y aparece un mensaje de error:
insert into libros (titulo,autor)
  values ('Uno','Richard Bach');

-- Para desactivar la opción "identity_insert"
set identity_insert libros off;

-- Intentemos ingresar un valor para el campo "codigo" (genera error):
insert into libros (codigo,titulo,autor)
  values (300,'Uno','Richard Bach');
```

Tenemos como resultado:

```
SQLQuery1.sql - DI...EGO-PC\diego (62)) * X
insert into libros (titulo,autor,precio)
values('Aprenda PHP','Mario Molina',45.60);

select * from libros;

-- Para saber cuál es el valor de inicio del campo "identity" de la tabla "libros":
select ident_seed('libros');

-- Si intentamos ingresar un valor para el campo "codigo" (genera error):
insert into libros (codigo,titulo,autor,precio)
values(106,'Martin Fierro','Jose Hernandez',25);

-- Para permitir ingresar un valor en un campo de identidad activamos la opción "identity_insert":
set identity_insert libros on;

-- Recordemos que si "identity_insert" está en ON, la instrucción "insert" DEBE explicitar un valor:
insert into libros (codigo,titulo,autor)
values (100,'Matematica estas ahí','Paenza');
```

100 %

Results Messages

(1 row affected)  
(1 row affected)  
(1 row affected)  
(3 rows affected)  
(1 row affected)  
Msg 544, Level 16, State 1, Line 24  
No se puede insertar un valor explícito en la columna de identidad de la tabla "libros" cuando IDENTITY\_INSERT es OFF.  
(1 row affected)  
(1 row affected)  
Msg 544, Level 16, State 1, Line 21  
100 %

Query completed with errors. DIEGO-PC (14.0 RTM) DIEGO-PC\diego (62) b

Podemos comentar todos los comandos SQL que generan error y volver a ejecutar:

```
SQLQuery1.sql - DI...EGO-PC\diego (62)) * X
precio float
);

insert into libros (titulo,autor,precio)
values('El aleph','Borges',23);
insert into libros (titulo,autor,precio)
values('Uno','Richard Bach',18);
insert into libros (titulo,autor,precio)
values('Aprenda PHP','Mario Molina',45.60);

select * from libros;

-- Para saber cuál es el valor de inicio del campo "identity" de la tabla "libros":
select ident_seed('libros');

-- Si intentamos ingresar un valor para el campo "codigo" (genera error):
-- insert into libros (codigo,titulo,autor,precio)
-- values(106,'Martin Fierro','Jose Hernandez',25);
```

100 %

Results Messages

	codigo	titulo	autor	precio
1	100	El aleph	Borges	23
2	102	Uno	Richard Bach	18
3	104	Aprenda PHP	Mario Molina	45,6

	(No column name)
1	100

Query executed successfully.

