

## 15 - Truncate table

Aprendimos que para borrar todos los registros de una tabla se usa "delete" sin condición "where". También podemos eliminar todos los registros de una tabla con "truncate table". Por ejemplo, queremos vaciar la tabla "libros", usamos:

```
truncate table libros;
```

La sentencia "truncate table" vacía la tabla (elimina todos los registros) y conserva la estructura de la tabla.

La diferencia con "drop table" es que esta sentencia borra la tabla, "truncate table" la vacía.

La diferencia con "delete" es la velocidad, es más rápido "truncate table" que "delete" (se nota cuando la cantidad de registros es muy grande) ya que éste borra los registros uno a uno.

Otra diferencia es la siguiente: cuando la tabla tiene un campo "identity", si borramos todos los registros con "delete" y luego ingresamos un registro, al cargarse el valor en el campo de identidad, continúa con la secuencia teniendo en cuenta el valor mayor que se había guardado; si usamos "truncate table" para borrar todos los registros, al ingresar otra vez un registro, la secuencia del campo de identidad vuelve a iniciarse en 1.

Por ejemplo, tenemos la tabla "libros" con el campo "codigo" definido "identity", y el valor más alto de ese campo es "2", si borramos todos los registros con "delete" y luego ingresamos un registro, éste guardará el valor de código "3"; si en cambio, vaciamos la tabla con "truncate table", al ingresar un nuevo registro el valor del código se iniciará en 1 nuevamente.

### 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,
    titulo varchar(30),
    autor varchar(20),
    editorial varchar(15),
    precio float
);

go

insert into libros (titulo,autor,editorial,precio)
values ('El aleph','Borges','Emece',25.60);
insert into libros (titulo,autor,editorial,precio)
values ('Uno','Richard Bach','Planeta',18);

select * from libros;
```

```
-- Truncamos la tabla:
truncate table libros;

-- Ingresamos nuevamente algunos registros:
insert into libros (titulo,autor,editorial,precio)
  values ('El aleph','Borges','Emece',25.60);
insert into libros (titulo,autor,editorial,precio)
  values ('Uno','Richard Bach','Planeta',18);

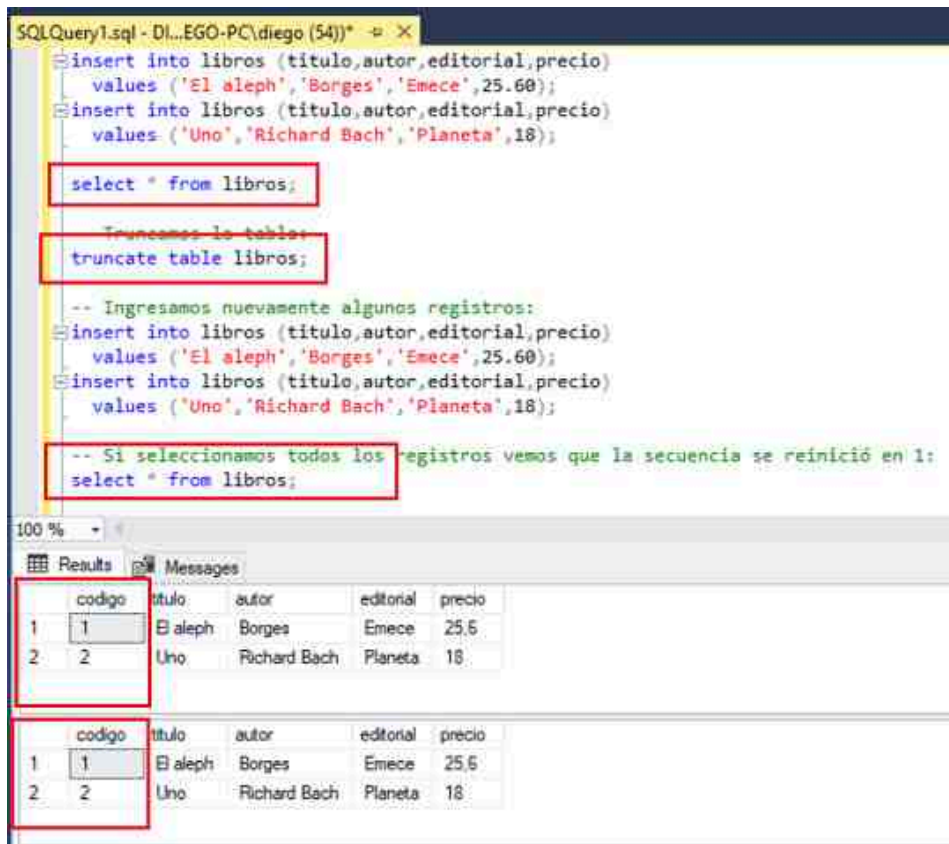
-- Si seleccionamos todos los registros vemos que la secuencia se reinició en 1:
select * from libros;

-- Eliminemos todos los registros con "delete":
delete from libros;

-- Ingresamos nuevamente algunos registros:
insert into libros (titulo,autor,editorial,precio)
  values ('El aleph','Borges','Emece',25.60);
insert into libros (titulo,autor,editorial,precio)
  values ('Uno','Richard Bach','Planeta',18);

-- Seleccionamos todos los registros y vemos que la secuencia continuó:
select * from libros;
```

Tenemos como resultado:



The screenshot shows a SQL Query Editor window with the following SQL commands:

```
insert into libros (titulo,autor,editorial,precio)
  values ('El aleph','Borges','Emece',25.60);
insert into libros (titulo,autor,editorial,precio)
  values ('Uno','Richard Bach','Planeta',18);

select * from libros;

-- Truncamos la tabla:
truncate table libros;

-- Ingresamos nuevamente algunos registros:
insert into libros (titulo,autor,editorial,precio)
  values ('El aleph','Borges','Emece',25.60);
insert into libros (titulo,autor,editorial,precio)
  values ('Uno','Richard Bach','Planeta',18);

-- Si seleccionamos todos los registros vemos que la secuencia se reinició en 1:
select * from libros;
```

The Results window shows the output of the SQL commands. The first query returns two rows, and the second query returns two rows, indicating that the sequence was reset to 1.

	codigo	titulo	autor	editorial	precio
1	1	El aleph	Borges	Emece	25,6
2	2	Uno	Richard Bach	Planeta	18

	codigo	titulo	autor	editorial	precio
1	1	El aleph	Borges	Emece	25,6
2	2	Uno	Richard Bach	Planeta	18