

4 - Tipos de datos básicos

Ya explicamos que al crear una tabla debemos resolver qué campos (columnas) tendrá y que tipo de datos almacenará cada uno de ellos, es decir, su estructura.

El tipo de dato especifica el tipo de información que puede guardar un campo: caracteres, números, etc.

Estos son algunos tipos de datos básicos de SQL Server (posteriormente veremos otros):

- **varchar**: se usa para almacenar cadenas de caracteres. Una cadena es una secuencia de caracteres. Se coloca entre comillas (simples); ejemplo: 'Hola', 'Juan Perez'. El tipo "varchar" define una cadena de longitud variable en la cual determinamos el máximo de caracteres entre paréntesis. Puede guardar hasta 8000 caracteres. Por ejemplo, para almacenar cadenas de hasta 30 caracteres, definimos un campo de tipo varchar(30), es decir, entre paréntesis, junto al nombre del campo colocamos la longitud. Si asignamos una cadena de caracteres de mayor longitud que la definida, la cadena no se carga, aparece un mensaje indicando tal situación y la sentencia no se ejecuta. Por ejemplo, si definimos un campo de tipo varchar(10) e intentamos asignarle la cadena 'Buenas tardes', aparece un mensaje de error y la sentencia no se ejecuta.
- **integer**: se usa para guardar valores numéricos enteros, de -2000000000 a 2000000000 aprox. Definimos campos de este tipo cuando queremos representar, por ejemplo, cantidades.
- **float**: se usa para almacenar valores numéricos con decimales. Se utiliza como separador el punto (.). Definimos campos de este tipo para precios, por ejemplo.

Antes de crear una tabla debemos pensar en sus campos y optar por el tipo de dato adecuado para cada uno de ellos.

Por ejemplo, si en un campo almacenaremos números enteros, el tipo "float" sería una mala elección; si vamos a guardar precios, el tipo "float" es más adecuado, no así "integer" que no tiene decimales. Otro ejemplo, si en un campo vamos a guardar un número telefónico o un número de documento, usamos "varchar", no "integer" porque si bien son dígitos, con ellos no realizamos operaciones matemáticas.

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(
    titulo varchar(80),
    autor varchar(40),
    editorial varchar(30),
    precio float,
    cantidad integer
);

exec sp_columns libros;

insert into libros (titulo,autor,editorial,precio, cantidad)
values ('El aleph','Borges','Emece',25.50, 100);
insert into libros (titulo,autor,editorial,precio,cantidad)
values ('Matematica estas ahi','Paenza','Siglo XXI',18.8,200);

select * from libros;
```

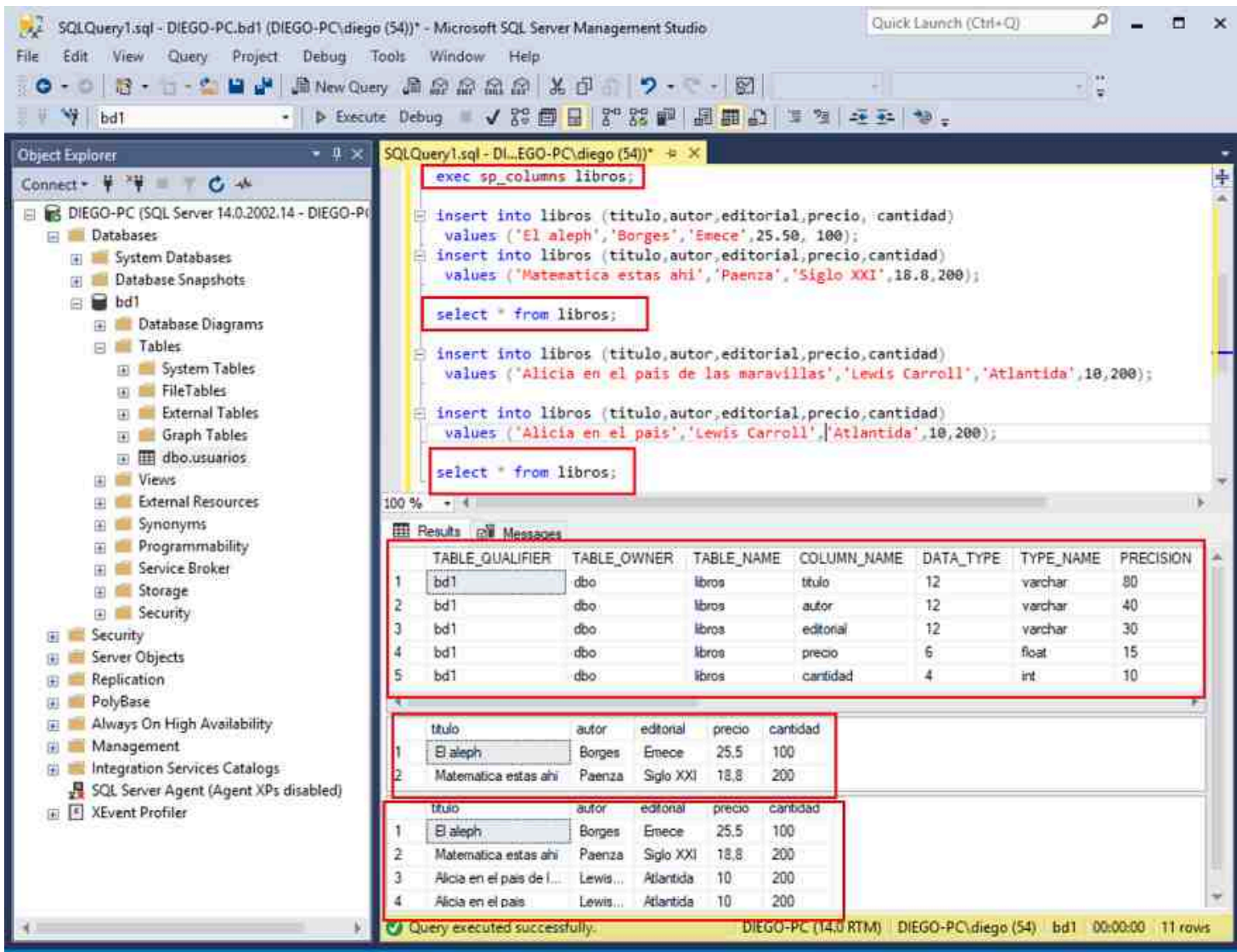
```
insert into libros (titulo,autor,editorial,precio,cantidad)
values ('Alicia en el pais de las maravillas','Lewis Carroll','Atlantida',10,200);

insert into libros (titulo,autor,editorial,precio,cantidad)
values ('Alicia en el pais','Lewis Carroll','Atlantida',10,200);

select * from libros;
```

Es importante el valor que definimos entre paréntesis luego de la palabra clave varchar. Si definimos el campo titulo de tamaño 80, luego significa que no podemos almacenar un nombre de libro de más de 80 caracteres.

La ejecución de este lote de comandos SQL genera una salida similar a:



Recordar que muchos de los datos que obtenemos con los procedimientos almacenados como 'sp_columns' los podemos consultar en forma más sencilla en la ventana del "Object Explorer":

