

# Valores por defecto

Asigna un valor específico a una columna cuando el valor para ello no ha sido explícitamente proporcionado para tal columna en una sentencia insert o de adición de un nuevo registro de la tabla.

## Ejercicio

Cree la siguiente base de datos en SQL Server.

```
create database programas3
go
use programas3
go
```

Con el siguiente código, usted va a crear tres valores por defecto para el sexo, estado y año actual.

```
CREATE DEFAULT DSexo          AS 'F'
GO
CREATE DEFAULT DAnnoActual AS YEAR(GETDATE());
GO
```

Tomando como base los ejemplos anteriores cree un valor por defecto que se llame DEstado y que tome el valor de A.

# Tipos de datos

Los tipos de datos especifican que tipo de valores son permitidos en cada una de las columnas que conforman la estructura de la fila. Por ejemplo, si desea almacenar precios de productos en una columna debería especificar que el tipo de datos sea **money**, si desea almacenar nombres debe escoger un tipo de dato que permita almacenar información de tipo carácter.

Además, SQL Server nos ofrece un conjunto de tipos de datos predefinidos, pero también existe la posibilidad de definir tipos de datos de usuario.

Al asignar un tipo de datos a un objeto se definen cuatro atributos del objeto:

- La clase de datos que contiene el objeto, por ejemplo, carácter, entero o binario.
- La longitud del valor almacenado o su tamaño.
- La precisión del número (sólo tipos de datos numéricos). La precisión es el número de dígitos que puede contener el número. Por ejemplo, un objeto smallint puede contener hasta 5 dígitos, con lo que tiene una precisión de 5.
- La escala del número (sólo tipos de datos numéricos). La escala es el máximo número de dígitos a la derecha del separador decimal. Por ejemplo, un objeto int no puede aceptar un separador decimal y tiene una escala de 0. Un objeto money puede tener hasta 4 dígitos a la derecha del separador decimal y tiene una escala de 4. Si un objeto se define como money, puede contener hasta 19 dígitos y 4 de ellos pueden estar a la derecha del decimal. El objeto usa 8 bytes para almacenar los datos. Por tanto, el tipo de datos money tiene una precisión de 19, una escala de 4 y una longitud de 8.

La siguiente es una tabla que describe los tipos de datos provistos por SQL Server:

Categoría	Descripción	Tipo de datos	Descripción
Binario	Almacenan cadenas de bits. La data consiste de números hexadecimales. Por ejemplo el decimal 245 es F5 en hexadecimal.	binary	La data debe tener una longitud fija (hasta 8 KB).
		varbinary	Los datos pueden variar en el número de dígitos hexadecimales (hasta 8 KB)
		image	La data puede tener una longitud variable y exceder los 8Kb
Carácter	Consisten de una combinación de letras, símbolos y números. Por ejemplo las combinaciones "John928" y "(0*&(%B99nhjk)".	char	Los datos deben tener una longitud fija (Hasta 8 KB).
		varchar	La data puede variar en el número de caracteres (Hasta 8 KB.)
		text	Los datos pueden ser caracteres ASCII que excedan los 8 KB
Fecha y Hora	Consisten en combinaciones válidas de estos datos. No puede separar en tipos distintos el almacenamiento de sólo fechas o sólo horas.	Datetime	Fechas en el rango 01 Ene 1753 hasta el 31 Dic 9999 (Se requiere 8 bytes por valor).
		smalldatetime	Fechas en el rango 01 Ene 1900 hasta 06 Jun 2079 (Se requiere 4 bytes por valor).
Decimal	Consisten en información que almacena información significativa después del punto decimal.	decimal	Los datos pueden tener hasta 38 dígitos, todos los cuales podrían estar a la derecha del punto decimal. Este tipo de dato guarda un valor exacto del número y no una aproximación
		numeric	Para SQL Server, el tipo de dato numeric es equivalente al tipo de datos decimal.
Punto flotante	Números aproximados (Punto flotante)	float	Datos en el rango de 1.79E + 308 hasta 1.79E + 308.
		real	Datos en el rango de 3.40E + 38 hasta 3.40E + 38.
Enteros	Consiste en información numérica positiva o negativa como por ejemplo -5, 0 y 25	bigint	Datos en el rango de 2 <sup>63</sup> (-9223372036854775808) hasta 2 <sup>63</sup> -1 (9223372036854775807). Se requieren de 8 bytes para almacenar estos valores.
		int	Datos en el rango de -2.147.483.648 hasta 2.147.483.647. Se requieren de 4 bytes para almacenar estos valores.
		smallint	Datos en el rango de -32.768 hasta 32.767. Se requieren 2 bytes por cada valor de este tipo.
		tinyint	Datos entre 0 y 255, se requiere de 1 byte
Monetarios	Cantidades monetarias positivas o negativas.	money	Datos monetarios entre -922.337.203.685.477,5808 y +922.337.203.685.477,5807 (Se requieren 8 bytes por valor)
		smallmoney	Datos monetarios entre -214.748,3648 y 214.748,3647 (Se requieren de 4 bytes por valor)

Los tipos de datos definidos por el usuario están basados en los tipos de datos disponibles a través de SQL Server. Los tipos de datos definidos por el usuario se pueden emplear para asegurar que un dato tenga las mismas características sobre múltiples tablas.

## Ejercicio

En la base de datos creada anteriormente, ejecute las siguientes instrucciones. Son ejemplos diversos de posibles tipos de datos que puede definir un usuario.

```
--Tipo de datos cedula
CREATE RULE RCedula AS (@Persona like '[0-9]-[0-9][0-9][0-9][0-9]-[0-9][0-9][0-9][0-9]')
GO
EXEC sp_addtype          'TCedula','char (11)', 'not null'
GO
EXEC sp_bindrule         'RCedula','TCedula'
GO

--Tipo de datos telefono
CREATE RULE RTelefono AS
    (@telefono like '[0-9][0-9][0-9][0-9]-[0-9][0-9][0-9][0-9]')
    or
    (@telefono like '([0-9][0-9][0-9]) [0-9][0-9][0-9][0-9]-[0-9][0-9][0-9][0-9]')
GO
EXEC sp_addtype          'TTelefono', 'varchar (17)', 'not null'
GO
EXEC sp_bindrule         'RTelefono','TTelefono'
GO

--Tipo de datos correo
CREATE RULE RCorreo AS (@correo like ('[a-Z]%@[a-Z]%.[a-Z]%'))
GO
EXEC sp_addtype          'TCorreo','varchar (50)', 'not null'
GO
EXEC sp_bindrule         'RCorreo','TCorreo'
GO

--Tipo de datos estado
CREATE RULE REstado AS (@Estado in ('A','I')) --Activo o inactivo
GO
EXEC sp_addtype          'TEstado', 'char (1)', 'not null'
GO
EXEC sp_bindrule         'REstado','TEstado'
GO
EXEC sp_bindefault       'DEstado','TEstado'
GO
```

Ya usted puede crear los siguientes tipos de datos:

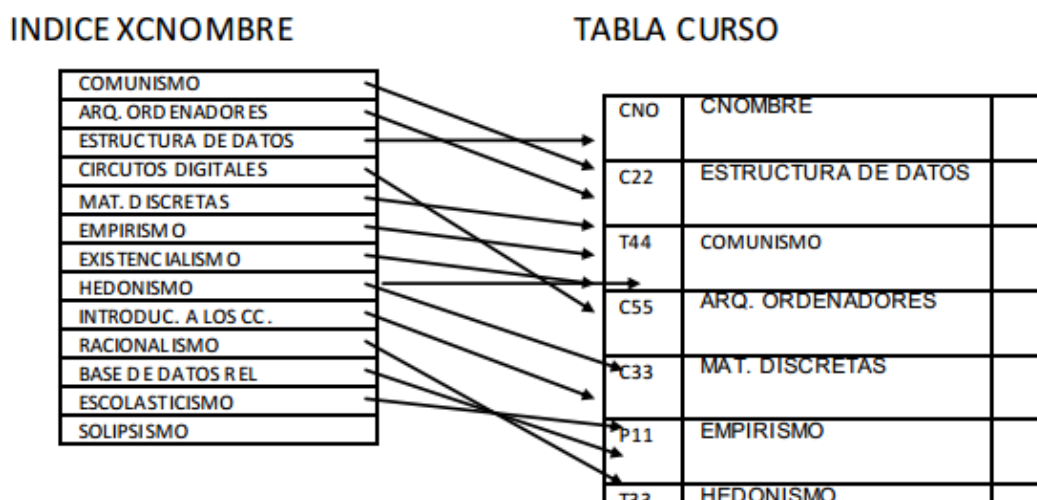
- Nombre = RCodigoCurso
- Tipo = TCodigoCurso
- Formato = IS-1614
- Nombre = RCarne
- Tipo = TCarne
- Formato = 200217875
- Nombre = Rsexo
- Tipo = TSexo
- Valor por defecto = DSexo
- Formato = F o M

Ahora, ejecute el script llamado BASE DE DATOS.sql, que contiene la estructura de las tablas con las que vamos a continuar trabajando.

## Índices

Un índice es una estructura interna que el sistema puede usar para encontrar uno o más registros en una tabla de forma rápida. En efecto, un índice de base de datos es, conceptualmente, similar a un índice encontrado al final de cualquier libro de texto. De la misma forma que el lector de un libro acudiría a un índice para determinar en qué páginas se encuentra un determinado tema, un sistema de base de datos leerá un índice para determinar las posiciones de registros seleccionados por una consulta SQL. En otras palabras, la presencia de un índice puede ayudar al sistema a procesar algunas consultas de un modo más eficiente.

Un índice de base de datos se crea para una columna o grupo de columnas. La figura siguiente muestra un índice (XCNOMBRE) basado en la columna CNOMBRE de la tabla CURSO. Observemos que el índice, a diferencia de la tabla CURSO, representa valores CNOMBRE en orden. Además, el índice es pequeño en relación con el tamaño de la tabla. Por lo tanto, es, probablemente, más fácil que el sistema busque el índice para localizar un registro con un valor CNOMBRE dado, a que explore toda la tabla en busca de ese valor. Por ejemplo, el índice XCNOMBRE podría ser muy útil al sistema cuando ejecute la siguiente sentencia SELECT.



Ventajas de los índices:

- Acceso directo a un registro especificado
- Ordenación

Desventajas de los índices:

- Espacio de disco usado por el índice
- Costos de actualización

Tenemos tres tipos de índices.

- El primero corresponde a las claves primarias, que como vimos, también se pueden crear en la parte de definición de columnas.
- El segundo tipo de índice permite definir índices sobre una columna, sobre varias, o sobre partes de columnas. Para definir estos índices se usan indistintamente las opciones KEY o INDEX.
- El tercero permite definir índices con claves únicas, también sobre una columna, sobre varias o sobre partes de columnas. Para definir índices con claves únicas se usa la opción UNIQUE.

La diferencia entre un índice único y uno normal es que en los únicos no se permite la inserción de filas con claves repetidas. La excepción es el valor NULL, que sí se puede repetir.

### Ejercicio

Puede ejecutar en la base de datos creada anteriormente, el siguiente código que contiene un ejemplo de un índice.

```
--drop INDEX IndApellidosPersonas on Personas
CREATE INDEX IndApellidosPersonas
on Personas(apellido1)
GO
```

Agregue un índice a la tabla estudiantes que utilice el atributo carne.

## Instrucciones insert, update y delete

---

### Insert

La forma más directa de insertar una fila nueva en una tabla es mediante una sentencia INSERT. En la forma más simple de esta sentencia debemos indicar la tabla a la que queremos añadir filas, y los valores de cada columna. Las columnas de tipo cadena o fechas deben estar entre comillas sencillas o dobles, para las columnas numéricas esto no es imprescindible, aunque también pueden estar entrecomilladas.

## Ejercicio

```
--inserciones PERSONAS
INSERT PERSONAS(cedula,nombre,apellido1,apellido2,direccion,sexo)
VALUES ('2-0562-0727','Leonardo','Viquez', 'Acuña', 'Ciudad Quesada','M')
```

Otra opción consiste en indicar una lista de columnas para las que se van a suministrar valores. A las columnas que no se nombren en esa lista se les asigna el valor por defecto.

```
INSERT PERSONAS(cedula,nombre,apellido1,apellido2,direccion)
VALUES ('2-0562-0729','Laura','Viquez', 'Acuña', 'Ciudad Quesada'),
       ('2-0562-0726','Martha','Viquez', 'Acuña', 'Ciudad Quesada'),
       ('2-0562-0722','Gaudy','Esquivel', 'Vega', 'San Ramón'),
       ('2-0562-0730','Lorena','Valerio', 'Solís', 'San Ramón'),
       ('2-0562-0735','Mario','Briceño', 'Campos', 'San José')
```

Utilice la siguiente instrucción: `select * from nombreTabla`.

Si intentamos insertar dos filas con el mismo valor de la clave única se produce un error y la sentencia no se ejecuta.

```
INSERT PERSONAS(cedula,nombre,apellido1,apellido2,direccion) VALUES ('2-0562-0729','Marcela','Chaves', 'Álvarez',
'Ciudad Quesada')
```

Inserte los siguientes datos:

- En la tabla telefonos:
  - El teléfono 2401-3130 pertenece a Leonardo Viquez
  - El teléfono 2401-3137 pertenece a Gaudy Esquivel
  - El teléfono 2401-3133 pertenece a Lorena Valerio
  - El teléfono 8888-9876 pertenece a Lorena Valerio
- En la tabla correos:
  - El correo [lviquez@gmail.com](mailto:lviquez@gmail.com) pertenece a Leonardo Viquez
  - El correo [gesquivez@hotmail.com](mailto:gesquivez@hotmail.com) pertenece a Gaudy Esquivel
  - El correo [lvalerio@hotmail.com](mailto:lvalerio@hotmail.com) pertenece a Lorena Valerio
  - El correo [lorenavalerio@gmail.com](mailto:lorenavalerio@gmail.com) pertenece a Lorena Valerio

## Update

Podemos modificar valores de las filas de una tabla usando la sentencia UPDATE. En su forma más simple, los cambios se aplican a todas las filas, y a las columnas que especifiquemos.

```
UPDATE personas SET direccion= 'San Carlos'
```

Podemos, del mismo modo, actualizar el valor de más de una columna, separándolas en la sección SET mediante comas:

```
UPDATE personas SET direccion= 'San Carlos', sexo= 'F'
```

Pero no tenemos por qué actualizar todas las filas de la tabla. Podemos limitar el número de filas afectadas de varias formas. La primera es mediante la cláusula WHERE. Usando esta cláusula podemos establecer una condición. Sólo las filas que cumplan esa condición serán actualizadas:

```
UPDATE personas SET direccion= 'San José', sexo= 'M' where cedula ='2-0562-0735'
```

### Ejercicio

Cambie el sexo por M cuando la cedula sea 2-0562-0727.

### Delete

Para eliminar filas se usa la sentencia DELETE. La sintaxis es muy parecida a la de UPDATE. Del siguiente modo se eliminan todas las filas de la tabla.

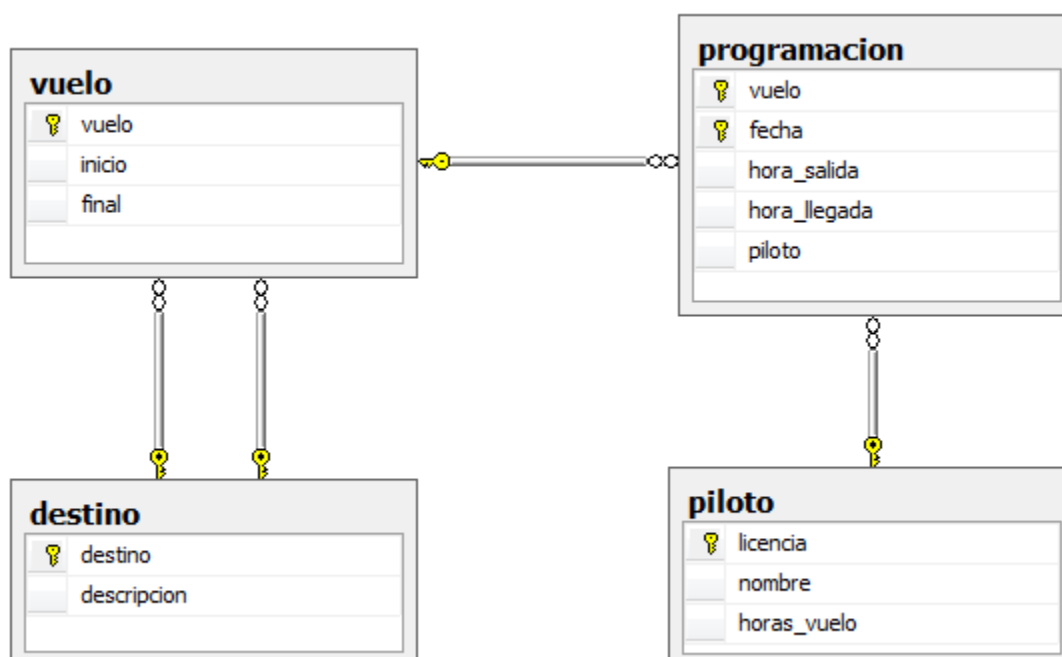
```
delete from correos
```

Pero es más frecuente que sólo queramos eliminar ciertas filas que cumplan determinadas condiciones. La forma más normal de hacer esto es usar la cláusula WHERE.

```
delete from correos where cedula ='2-0562-0735'
```

## Ejercicio

Usted cuenta con la siguiente base de datos en SQL Server:



Para crear dicha base de datos, abra los scripts aerolinea(1).sql y aerolinea(2).sql y elabore los siguientes ejercicios:

1. Cree un tipo de datos, con las siguientes características. Además, asígnelo a la tabla destino.
  - Nombre = RDestino
  - Tipo = TDestino
  - Formato = 100D
2. Cree un índice para la tabla programación al atributo fecha.
3. Cree un tipo de datos con las siguientes características.
  - Nombre = REstado
  - Tipo = TEstado
  - Formato
    - P=Programado
    - V=Volando
    - A=Aterrizado
    - R=Reprogramado
    - C=Cancelado
4. Cree un valor por defecto para el tipo de datos anterior que ponga el estado P.
5. Agregue un atributo estado a la tabla vuelo que utilice es tipo de datos TEstado. Tome en cuenta que este atributo puede ser nulo.
6. Agregue un atributo tarifa a la tabla vuelo que es de tipo money y puede ser nulo.
7. Ejecute los dos scripts mencionados anteriormente.
8. Escriba las sentencias SQL para que cada tabla de la Base de Datos tenga almacenados los siguientes datos como se muestra en las tablas.

### **Tabla Piloto**

<b>Licencia</b>	<b>Nombre</b>	<b>HorasVuelo</b>
1111	Juan Pérez	4000
2222	Carlos Mora	5000
3333	Marta Mena	8000
4444	Jorge Alfaro	2900
5555	Pedro Rojas	2000

### **Tabla Vuelo**

<b>Vuelo</b>	<b>Inicio</b>	<b>Final</b>
520	San José	New York
618	Panamá	New York
705	San José	Londres
820	Caracas	Montreal

### **Tabla Programación**

<b>Vuelo</b>	<b>Fecha</b>	<b>HoraSalida</b>	<b>HoraLlegada</b>	<b>Piloto</b>
520	25/01/99	5:50	6:30	2222
520	30/01/99	12:36	16:23	3333
618	2/02/99	10:20	15:55	2222
705	1/03/99	16:15	5:29	1111
705	5/03/99	22:00	8:00	1111



## **Tabla Destino**

<b>Destino</b>	<b>Descripción</b>
1D	San José
2D	Panamá
3D	Miami
4D	New York
5D	Londres
6D	Monterreal

9. Inserte el destino París con el identificador 7D.
10. Cambiar el destino final del vuelo 705 de Londres a París.
11. Agregar a todos los pilotos 100 horas de vuelo.
12. Eliminar de la base de datos todos los pilotos que tienen menos de 3000 horas de vuelo.
13. Modifique el siguiente caso: La programación del vuelo 618 del día 2/02/99 lo va a realizar el piloto 4444.
14. Programar el vuelo 820 para el 1° de noviembre de 2013 con hora de salida a las 08:00 am y hora de llegada 11:35 am. Además, el piloto que realizará el vuelo es Marta Mena.
15. El aeropuerto de New York recibió un ataque terrorista por lo todos los vuelos que tienen como destino final New York serán reprogramados a Miami. Por lo que usted debe actualizar en la base de datos tanto el estado de los vuelos como el destino final.
16. Agregue una nueva tabla llamada avión con los siguientes atributos: identificador, tipo, capacidad y descripción.
17. Agregue al menos 5 aviones a la tabla creada anteriormente.
18. Modifique la tabla vuelo para que contenga el avión que realizará el viaje. Para esto primero debe agregar una nueva columna a la tabla vuelo, luego insertar los datos correspondientes a cada valor y finalmente crear la llave foránea a la tabla avión.