

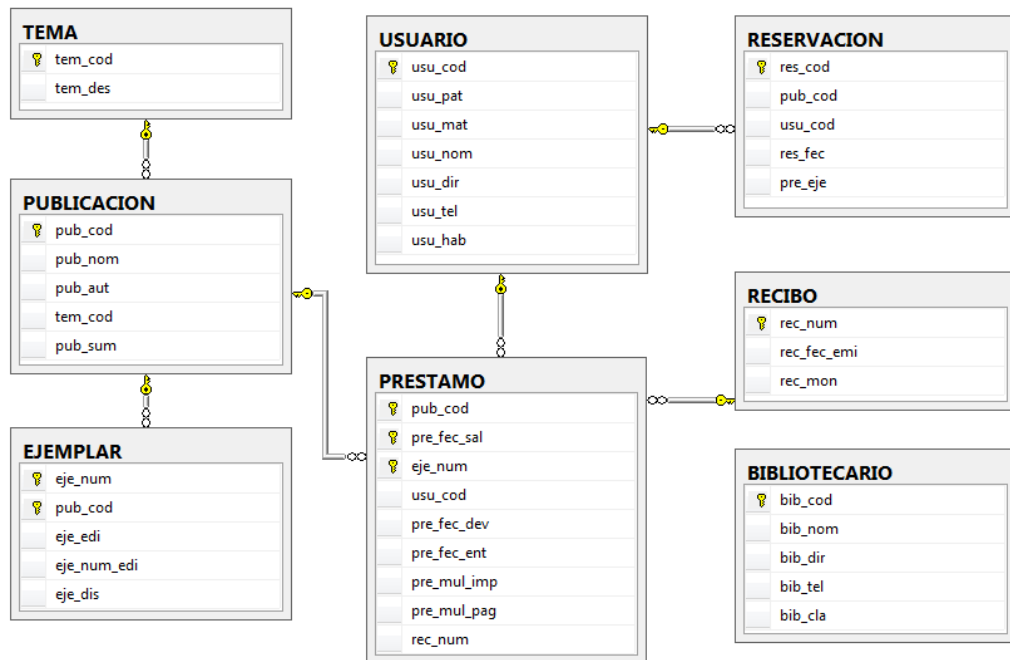
Capítulo IV

Operaciones de Mantenimiento de Datos

En este capítulo conoceremos de modo más detallado cómo operar con los datos en las tablas SQL Server: ingresar nuevos datos (sentencia INSERT), actualizar los datos existentes (sentencia UPDATE), y eliminar registros (sentencia DELETE).

1. PREPARACIÓN DEL ESCENARIO

Para desarrollar este capítulo crearemos previamente una base de datos **BibliotecaGyS** cuyo modelo se muestra a continuación, y en la que ejecutaremos las operaciones de mantenimiento de datos:



A continuación una descripción de lo que almacenará cada tabla:

PUBLICACION: relación de todas las publicaciones registrada en la biblioteca. Las publicaciones están clasificadas por temas.

TEMA: relación de todos los temas disponibles en la biblioteca.

EJEMPLAR: para cada publicación puede haber más de un ejemplar de acuerdo con la demanda de la misma.

BIBLIOTECARIO: relación del personal que atiende a los usuarios de la biblioteca.

USUARIO: para hacer uso de los servicios de la biblioteca se requiere estar registrado como usuario.

PRESTAMO: registra los movimientos de las publicaciones.

RESERVACION: si para una publicación no hubiera un ejemplar disponible, el usuario puede hacer la reserva correspondiente.

Ejercicio 4.1: Creación de la base de datos BibliotecaGyS

1. Inicie su sesión en **SQL Server Management Studio**.
2. En el menú **Archivo** ejecute **Abrir, Archivo**.
3. Ubique en su CD del libro la carpeta **ScriptsBD** y abra el archivo **BibliotecaGyS-Tablas.sql**.
4. Haga clic en el botón **Ejecutar** de la barra de herramientas o pulse la tecla [F5].

2. INSERCIÓN DE FILAS – INSTRUCCIÓN INSERT

Sintaxis

```
INSERT [INTO] nombre_tabla[ ( lista_de_columnas ) ]  
VALUES( lista_de_valores )
```

- **lista_de_columnas** es la relación de columnas en las que se almacenarán los valores especificados en *lista_de_valores*.
- **lista_de_valores** es la relación de los valores a almacenar en la fila a insertar.

Ejercicio 4.2: Inserción de una fila con lista de valores completa

1. En su ventana de consulta ejecute las siguientes instrucciones:

```
USE BibliotecaGyS  
go  
  
INSERT Usuario(usu_nom, usu_pat, usu_mat,  
usu_cod, usu_dir, usu_tel, usu_hab)  
VALUES('ABELARDO', 'BARRIONUEVO',  
'CASTILLO', '961002',  
'LOS PINOS 234 - LOS OLIVOS', NULL, 1)  
go
```

En este caso, cuando se entrega valores para cada una de las columnas de una tabla, el orden de los valores en la lista de valores debe coincidir con el orden de las columnas en la lista de columnas.

5. También se puede insertar una fila con lista de valores completa usando la siguiente sintaxis:

```
INSERT usuario  
VALUES('961003', 'ANTAYHUA', 'OSORIO',  
'ANA ROSA',  
'AV. ARICA 1312 - BREÑA', '4321234', 1)  
go
```

En este caso, cuando en la instrucción no se establece la lista de columnas, pero se está entregando la lista de valores completa, los valores de la lista se ordenan según el orden natural de las columnas en la tabla.

6. También es posible insertar una fila con lista de valores completa que contengan valores NULL, siempre y cuando la columna correspondiente los permita.

```
INSERT usuario
VALUES( '961004', 'QUISPE', 'CUYUBAMBA',
'JUAN JOSE',
'AV. URUGUAY 546 - CERCADO - LIMA', NULL, 1)
go
```

En este caso, no se conoce el teléfono del usuario, pero como la columna permite nulos registramos NULL como valor de dicha columna.

Ejercicio 4.3: Inserción de una fila con lista de valores incompleta

Cuando la lista de valores está incompleta es obligatorio establecer en qué columna va cada valor de la lista, por lo que la lista de columnas es obligatoria en la instrucción INSERT.

```
INSERT usuario(usu_cod, usu_pat, usu_mat,
usu_nom, usu_dir, usu_hab)
VALUES('961005', 'ALTAMIRANO', 'CASTRILLON',
'MARIO', 'COSME BUENO 256 - BELLAVISTA', 1)
go
```

El orden de las columnas en la lista de columnas, y el orden de los valores en la lista de valores se deben corresponder.

2.1. Uso de DEFAULT VALUES

Si cada una de las columnas de la tabla tiene definido un valor por defecto, o permite valores nulos, o tiene la propiedad IDENTITY, podemos insertar filas sin tener que especificar los valores a insertar tal como se muestra en la sintaxis siguiente:

```
INSERT [INTO] nombre_tabla DEFAULT VALUES
```

Esta orden inserta una fila de la siguiente manera:

- Si la columna permite nulos, inserta NULL en dicha columna.
- Si la columna tiene definido un valor por defecto, inserta el valor por defecto.
- Si la columna tiene la propiedad IDENTITY, inserta el valor de identidad correspondiente.

2.2. La propiedad IDENTITY

Permite definir una columna numérica entera en la que el valor de la columna es autogenerado a medida que se van insertando filas. Es útil cuando se desea emplear claves primarias numéricas.

Ejercicio 4.4: Uso de la propiedad IDENTITY

1. Cree una tabla que tenga la propiedad IDENTITY.

```
USE BibliotecaGyS
go

CREATE TABLE pruebaIdentidad(
    codigo int IDENTITY(1,1) PRIMARY KEY,
    nombre varchar(15) not null,
    telefono varchar(15) null )
go
```

2. Inserte filas en la tabla ejecutando las siguientes instrucciones:

```
INSERT INTO pruebaIdentidad(nombre, telefono)
VALUES ('Gustavo', '99981234')

INSERT INTO pruebaIdentidad(nombre, telefono)
VALUES ('Ricardo', '2211234')

INSERT INTO pruebaIdentidad(nombre, telefono)
VALUES ('Gloria', '3493212')
go

SELECT * FROM pruebaIdentidad
go
```

En IDENTITY(1,1), el primer argumento establece el primer valor de identidad a generar, y el segundo argumento es el incremento a utilizar para generar los siguientes valores.

Ejercicio 4.5: Inserción de la fecha del sistema

La función **getdate()** entrega la fecha y hora del sistema. La puede utilizar para insertar la fecha y hora del sistema en una columna de tipo fecha-hora.

```
USE BibliotecaGyS
go

INSERT INTO recibo(rec_num, rec_fec_emi, rec_mon)
VALUES(1001, getdate(), 10)
go

SELECT * FROM recibo
go
```

Ejercicio 4.6: Inserción de un valor de fecha específico

Para enviar un dato fecha al servidor, la fecha se envía como una cadena con formato de fecha. Por lo general, cuando la conexión es a un servidor SQL remoto, no sabemos cuál es el formato de fecha predeterminado que está utilizando el servidor, por lo que en ocasiones podemos tener problemas con la manipulación de las fechas.

1. Cree una tabla para hacer algunas pruebas con las fechas:

```
USE BibliotecaGyS
go

CREATE TABLE PruebaFechas (
    fecha datetime not null )
go
```

2. Ahora, inserte la fecha 26 de noviembre de 1980 mediante la siguiente instrucción:

```
-- Prueba 1: insertar 26 de noviembre de 1980
```

```
INSERT INTO PruebaFechas VALUES( '26 Nov 1980' )  
go
```

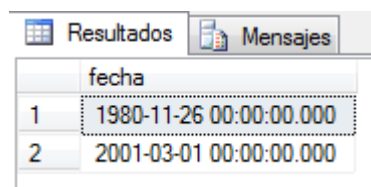
```
SELECT * FROM PruebaFechas  
go
```

Observe que la fecha se registró correctamente.

3. Ahora, vamos a suponer que la conexión al servidor SQL se realiza desde algún lugar de USA, donde se acostumbra a utilizar el formato mes/día/año para expresar las fechas.

```
-- Prueba 2: insertar la fecha 3 de enero  
-- del 2001 usando el formato mes/día/año  
INSERT INTO PruebaFechas VALUES( '01/03/2001' )  
go
```

```
SELECT * FROM PruebaFechas  
go
```



	fecha
1	1980-11-26 00:00:00.000
2	2001-03-01 00:00:00.000

Note que se ha insertado incorrectamente. La fecha registrada es 1 de marzo del 2001. El servidor con el que se ha desarrollado este libro tiene como formato predeterminado de fecha 'dd/mm/aaaa'.

4. Ahora, inserte la fecha 26 de noviembre de 1980 con la siguiente instrucción:

```
-- Prueba 3: insertar la fecha 26 de noviembre  
-- de 1980 usando el formato mes/día/año  
INSERT INTO PruebaFechas VALUES( '11/26/1980' )  
go
```

5. Se genera el siguiente mensaje de error:

Mens. 242, Nivel 16, Estado 3, Línea 1

La conversión del tipo de datos varchar en datetime produjo un valor fuera de intervalo.
Se terminó la instrucción.

El valor de fecha se ha leído como el día 11 del mes 26 del año 1980, lo que produce un valor de fecha fuera de rango (fecha inválida). Recuerde que mi servidor lee las fechas en el formato día/mes/año.

Cuando se envía fechas al servidor se recomienda informar al servidor en qué formato se le está enviando las fechas para que las lea correctamente.

6. Infórmele al servidor acerca del formato de sus fechas:

```
-- Informando al servidor en qué formato
-- se envían las cadenas de fecha
SET DATEFORMAT MDY
go
```

7. Ahora, trate de insertar nuevamente la fecha 26 de noviembre de 1980 ejecutando la instrucción que generó el error.

```
-- Prueba 4: insertar la fecha 26 de noviembre
-- de 1980 usando el formato mes/día/año
INSERT INTO PruebaFechas VALUES( '11/26/1980' )
go

SELECT * FROM PruebaFechas
go
```

La fecha se inserta correctamente.

3. ACTUALIZACIÓN DE DATOS – INSTRUCCIÓN UPDATE)

Sintaxis

```
UPDATE nombre_tabla
SET      columnaX = expresiónX,
        columnaP = expresiónP, ...
[ WHERE condición_de_las_filas_a_actualizar ]
```

- **columnaX, columnaP** son las columnas cuyo contenido se actualizará.
- **expresiónX, expresiónP** establecen los nuevos valores a almacenar en las columnas *columnaX* y *columnaP* respectivamente.
- **condición_de_las_filas_a_actualizar** es una expresión lógica que determina las filas en las que la actualización se debe llevar a cabo.

Ejercicio 4.7: Uso de UPDATE

1. Ejecute las siguientes instrucciones en su ventana de consulta:

```
USE BibliotecaGyS
go

SELECT * FROM usuario
go
```

8. Note que el usuario "BARRIONUEVO" no tiene registrado su teléfono. Procederemos a hacerlo.

```
UPDATE usuario
SET usu_tel = '5714315'
WHERE usu_cod = '961002'
go

SELECT * FROM usuario
WHERE usu_cod = '961002'
go
```

4. ELIMINACIÓN DE FILAS – INSTRUCCIÓN DELETE)

Sintaxis

```
DELETE [FROM] nombre_tabla  
[ WHERE condición_de_las_filas_a_eliminar ]
```

- **condición_de_las_filas_a_eliminar** es una expresión lógica que determina las filas en las que la eliminación se debe llevar a cabo.

Ejercicio 4.8: Uso de DELETE

1. Ejecute las siguientes instrucciones en su ventana de código:

```
USE BibliotecaGyS  
go
```

```
SELECT * FROM usuario  
go
```

9. Elimine el registro del usuario cuyo código es 961004.

```
DELETE FROM usuario  
WHERE usu_cod = '961004'  
go
```

```
SELECT * FROM usuario  
go
```

5. ELIMINACIÓN DE TODAS LAS FILAS DE UNA TABLA - INSTRUCCIÓN TRUNCATE TABLE)

Si bien la sentencia DELETE utilizada sin la cláusula WHERE permite eliminar todas las filas de una tabla, se recomienda el uso de TRUNCATE TABLE porque esta instrucción se ejecuta más rápido debido a que no registra la eliminación de cada fila en el log de transacciones de la base de datos como lo hace DELETE. La sintáxis es:

```
TRUNCATE TABLE nombre_tabla
```

Sin embargo, esta instrucción no puede ejecutarse sobre una tabla que es referenciada por una clave foránea, independientemente de que la tabla que hace la referencia (la tabla que posee la clave foránea) tenga filas registradas ó no.

Ejercicio 4.9: Uso de TRUNCATE TABLE

1. Ejecute las siguientes instrucciones en su ventana de consulta:

```
USE BibliotecaGyS  
go
```

```
TRUNCATE TABLE usuario  
go
```

2. Se genera el siguiente mensaje de error:

```
Mens. 4712, Nivel 16, Estado 1, Línea 1  
No se puede truncar la tabla 'usuario'. Una  
restricción FOREIGN KEY hace referencia a ella.
```

6. EJERCICIOS PROPUESTOS

1. Ejecute el script **BibliotecaGyS-Tablas.sql** para recrear la base de datos **BibliotecaGyS**.
2. Ejecute las instrucciones necesarias para registrar los siguientes datos en las tablas indicadas. Si alguna instrucción falla, explique porqué, y qué acción tomaría para corregir la falla.

Tabla TEMA

Resultados		Mensajes
	tem_cod	tem_des
1	1	ANALISIS Y DISEÑO DE SISTEMAS
2	2	ADMINISTRADORES DE BASES DE DATOS
3	3	LENGUAJES DE PROGRAMACION

Tabla PUBLICACION

Resultados		Mensajes			
	pub_cod	pub_nom	pub_aut	tem_cod	pub_sum
1	57	FOUNDATIONS OF POWERBUILDER PROGRAMMING	SMITH, BRIAN - SCHAAD, GORDON	4	NULL
2	55	MS VISUAL BASIC 6 - MANUAL DEL PROGRAMADOR	MICROSOFT PRESS	4	NULL
3	54	APRENDIENDO MS SQL SERVER 2000 EN 21 DIAS	WAYMIRE, RICHARD - SAWTELL, RICK	2	NULL
4	49	IMPLEMENTING A DATABASE ON MS SQL SERVER 7 - MOC ...	MICROSOFT	2	NULL
5	48	ORACLE 9i - GUIA DE APRENDIZAJE	ABBEY - COREY - ABRAMSIN	2	NULL
6	46	JAVA - GUIA DE DESARROLLO	JAWORSKI, JAMIE	3	NULL
7	44	DESARROLLO DE APLICACIONES C/S CON ERWIN 3, SQL S...	CORONEL CASTILLO, GUSTAVO	4	NULL
8	40	ANALISIS Y DISEÑO ORIENTADO A OBJETOS	MARTIN, JAMES - ODELL, JAMES	1	NULL
9	29	ORACLE - INTRODUCCION A PL/SQL V.2.0	ORACLE CORPORATION	2	NULL
10	28	MICROSOFT SQL SERVER - ADMINISTRATOR COMPANION	MICROSOFT	2	NULL

Tabla EJEMPLAR

Resultados

Mensajes

	eje_num	pub_cod	eje_edt	eje_num_edt	eje_dis
1	1	28	MICROSFOT CORPORATION	1	N
2	1	29	ORACLE CORPORATION	1	S
3	1	40	PRENTICE HALL HISPANOAMERICANA S.A.	1	N
4	2	40	PRENTICE HALL HISPANOAMERICANA S.A.	1	S
5	3	40	PRENTICE HALL HISPANOAMERICANA S.A.	1	N

Tabla USUARIO

Resultados		Mensajes					
	usu_cod	usu_pat	usu_mat	usu_nom	usu_dir	usu_tel	usu_hab
1	971017	HUERTAS	MARCELO	ANA MARIA	JR. RECUAY 397 - BREÑA	NULL	1
2	971018	ARRIOLA	EVANS	MARCELA	LAS AMAZONAS S/N MZA. Z LOTE 2 - LOS OLIVOS	4523489	1
3	971019	JIMENEZ	MAC ALLISTER	DIANA	JR. GENERAL VIDAL 348 - SAN ISIDRO	NULL	1
4	971020	APOLAYA	CASTRO	IVONNE	BARRIO OBRERO 1378 - LA VICTORIA	4352378	1

Tabla PRESTAMO

Resultados		Mensajes							
	pub_cod	pre_fec_sal	eje_num	usu_cod	pre_fec_dev	pre_fec_ent	pre_mul_imp	pre_mul_pag	rec_num
1	40	2013-03-14 21:03:15.553	1	971020	2013-03-16 21:03:15.553	2013-03-16 21:03:15.553	NULL	NULL	NULL
2	40	2013-04-01 21:03:15.567	3	971017	2013-04-04 21:03:15.567	2013-04-04 21:03:15.567	NULL	NULL	NULL