

Práctica A: Modificación de datos

Objetivos

Después de realizar esta práctica, el alumno será capaz de:

- Modificar los datos de las tablas con las instrucciones INSERT, DELETE y UPDATE.
- Insertar filas en una tabla con las palabras clave DEFAULT y DEFAULT VALUES.
- Modificar datos en función de los datos de otras tablas.

Requisitos previos

- El archivo lab04.zip contiene los archivos de comandos necesarios para la realización de la práctica, así como las correspondientes soluciones.
- Para la realización de las prácticas es necesario crear la base de datos library, ejecutando el script bldlib.cmd del directorio Library.

Ejercicio 1

Uso de la instrucción INSERT

En este ejercicio va a utilizar la instrucción INSERT para agregar filas a las tablas de la base de datos **library**. A continuación, va a ejecutar una consulta para comprobar que se han agregado las filas nuevas. La carpeta Soluciones contiene las secuencias de comandos completas para este ejercicio.

✍ Insertar valores en la tabla **item**

En este procedimiento va a insertar en la tabla **item** filas que representan un libro de los fondos de la biblioteca.

1. Abra el Analizador de consultas SQL y, si así se le pide, inicie una sesión en el servidor local con autenticación de Microsoft Windows®.
2. En la lista **BD**, haga clic en **library**.
3. Inserte dos filas en la tabla **item** para el título número 8, *The Cherry Orchard*. Especifique los nombres de las columnas para las que proporciona valores. Indique que el primer elemento tiene tapas duras y que el segundo está encuadernado en rústica, y utilice los valores siguientes.
Answer_InsValues1.sql es la secuencia de comandos completa de este paso.

Nombre de columna	Datos
Isbn	10001 para HARDBACK (tapas duras); 10101 para SOFTBACK (rústica)
title_no	8
Cover	HARDBACK y SOFTBACK
Loanable	Y
Translation	ENGLISH

```
USE library
INSERT item (isbn, title_no, cover, loanable, translation)
VALUES (10001, 8, 'HARDBACK', 'Y', 'ENGLISH')
INSERT item (isbn, title_no, cover, loanable, translation)
VALUES (10101, 8, 'SOFTBACK', 'Y', 'ENGLISH')
GO
```

4. Ejecute la consulta y compruebe que se inserta una sola fila en dos transacciones distintas.

✍ Insertar valores en la tabla **copy**

En este procedimiento va a insertar en la tabla **copy** filas que representan un libro de los fondos de la biblioteca.

5. Agregue una fila a la tabla **copy** para el libro con tapas duras agregado en el paso 4 y utilice los valores siguientes. Answer_InsValues2.sql es la secuencia de comandos completa de este paso.

Nombre de columna	Datos
Isbn	10001 (número de ISBN del elemento con tapas duras agregado en el paso 1)
copy_no	1
title_no	8 (número de título de <i>The Cherry Orchard</i>)
On_loan	N

```
USE library
INSERT copy (isbn, copy_no, title_no, on_loan)
VALUES (10001,1,8,'N')
GO
```

6. Ejecute la consulta y compruebe que se inserta una sola fila.

✍ Determinar el idioma de traducción de un elemento

En este procedimiento escribirá una consulta que devuelva el idioma al que se ha traducido un elemento.

1. Escriba una consulta que obtenga de la tabla **item** el idioma de uno de los elementos insertados en el paso 4. Answer_Translate.sql es la secuencia de comandos completa de este paso.

```
USE library
SELECT translation
FROM item
WHERE isbn = 10001
GO
```

2. Ejecute la consulta para comprobar que devuelve los resultados deseados.

Ejercicio 2

Uso de la instrucción INSERT con la palabra clave DEFAULT

En este ejercicio va a utilizar la instrucción INSERT con la palabra clave DEFAULT para agregar dos filas a la tabla **title** y especificar datos parciales para las columnas que admiten valores NULL o que tienen valores predeterminados. La carpeta Soluciones contiene las secuencias de comandos completas para este ejercicio.

✍ Para determinar las columnas que permiten valores NULL

En este procedimiento determinará las columnas que permiten valores NULL.

1. Ejecute el procedimiento almacenado del sistema **sp_help** para averiguar qué columnas de la tabla **title** admiten valores NULL. No es necesario especificar valores para las columnas que admiten valores NULL, que tienen valores predeterminados o que tienen la propiedad IDENTITY. Answer_WhichNull.sql es la secuencia de comandos completa de este paso.

```
USE library
EXEC sp_help title
GO
```

2. Revise los segundos resultados devueltos para determinar qué columnas admiten valores NULL.

✍ Para insertar valores en la tabla title

En este procedimiento insertará valores en la tabla **title**.

1. Inserte una fila en la tabla **title** para el libro *The Art of Lawn Tennis*, de William T. Tilden. Utilice la palabra clave DEFAULT para las columnas que admitan valores NULL o que tengan valores predeterminados. No debe especificar ningún valor para la columna **title_no**, ya que tiene la propiedad IDENTITY. Answer_InsDefault1.sql es la secuencia de comandos completa de este paso.

```
USE library
INSERT title (title, author, synopsis)
VALUES ('The Art of Lawn Tennis', 'William T. Tilden',
        ,DEFAULT )
GO
```

2. Ejecute la consulta y compruebe que se inserta una sola fila.

✍ Para determinar el último valor de identidad utilizado

En este procedimiento, determinará el último valor de identidad utilizado.

1. Escriba una consulta para determinar el **title_no** del título que agregó en el paso 1 del procedimiento anterior. Answer_Identity.sql es la secuencia de comandos completa de este paso.

```
USE library
SELECT @@identity
GO
```

2. Ejecute la consulta y tome nota del valor devuelto.

✍ Para recuperar la última fila insertada en la tabla **title**

En este procedimiento recuperará la última fila insertada en la tabla **title**.

1. Escriba una consulta para comprobar que el título nuevo se ha agregado a la tabla **title**. Utilice el valor obtenido en el paso 1 del procedimiento anterior para la columna **title_no**. `Answer_LastRow.sql` es la secuencia de comandos completa de este paso.

```
USE library
SELECT *
  FROM title
 WHERE title_no = @@identity
GO
```

2. Ejecute la consulta para comprobar que devuelve los resultados deseados.

✍ Para insertar más valores en la tabla **title**

En este procedimiento insertará más valores en la tabla **title**.

1. Inserte una fila en la tabla **title** para el título *Riders of the Purple Sage*, de Zane Grey. Especifique *listaColumns* y los valores de las columnas que no admitan valores NULL ni tengan valores predeterminados. `Answer_InsValues3.sql` es la secuencia de comandos completa de este paso.

```
USE library
INSERT title (title, author)
  VALUES ('Riders of the Purple Sage', 'Zane Grey')
GO
```

2. Ejecute la consulta y compruebe que se inserta una sola fila.

✍ Para comprobar que los valores se han insertado en la tabla **title**

En este procedimiento comprobará que los valores se han insertado en la tabla **title**.

1. Escriba y ejecute una consulta para comprobar que se ha agregado el nuevo título a la tabla **title**. `Answer_ChkValues3.sql` es la secuencia de comandos completa de este paso.

```
USE library
SELECT *
  FROM title
 WHERE title = 'Riders of the Purple Sage'
GO
```

2. Ejecute la consulta para comprobar que devuelve los resultados deseados.

Ejercicio 3

Uso de la instrucción INSERT con la palabra clave DEFAULT VALUES

En este ejercicio va a utilizar la instrucción INSERT con la palabra clave DEFAULT VALUES para agregar filas a una tabla sin especificar valores. En primer lugar, va a crear y trabajar con una tabla de ejemplo de la base de datos **library**. La carpeta Soluciones contiene las secuencias de comandos completas para este ejercicio.

✍ Para crear la tabla sample1

En este procedimiento va a crear una tabla nueva en la base de datos **library** que admita valores NULL y en la que algunas columnas tengan valores predeterminados.

1. Ejecute la secuencia de comandos MakeSample1.sql para crear una tabla nueva llamada **sample1** en la base de datos **library**, con las características siguientes.

Nombre de columna	Tipo de datos	¿Propiedad IDENTITY?	¿Admite NULL?
Cust_id	Int	Sí (100,5)	No
Name	char(10)	No	Sí

```
USE library
CREATE TABLE sample1 (
    Cust_id int NOT NULL IDENTITY(100,5)
    ,Name char(10) NULL
)
GO
```

2. Ejecute la consulta para comprobar que crea la tabla **sample1**.

✍ Para insertar una fila de valores predeterminados en la tabla sample1

En este procedimiento va a insertar una fila en la tabla **sample1**. Para ello, utilizará la palabra clave DEFAULT VALUES. A continuación, va a escribir y ejecutar una consulta para comprobar que se ha agregado la fila nueva a la tabla. Answer_Insdefault2.sql es la secuencia de comandos completa de este procedimiento.

1. Escriba una consulta que inserte una fila nueva en la tabla **sample1** sin especificar nombres de columna. Utilice la palabra clave DEFAULT VALUES con la instrucción INSERT.

```
USE library
INSERT sample1
DEFAULT VALUES
GO
```

2. Ejecute la consulta y compruebe que se inserta una sola fila.

✍ Para comprobar que los valores se han insertado en la tabla sample1

En este procedimiento comprobará que los valores se han insertado en la tabla **sample1**.

1. Escriba una consulta para comprobar que se ha agregado la fila nueva a la tabla **sample1**. Answer_ChkDefault2.sql es la secuencia de comandos completa de este procedimiento.

```
USE LIBRARY
SELECT *
FROM sample1
GO
```

2. Ejecute la consulta y compare los resultados con los valores predeterminados definidos en la tabla.

Resultado

Su resultado debería ser similar al siguiente conjunto de resultados.

cust_id	Name
100	NULL

(1 fila afectada)

Ejercicio 4

Uso de la instrucción DELETE

En este ejercicio va a utilizar la instrucción DELETE para quitar el libro con el ISBN 10101 y el número de título 8 de la tabla **item** de la base de datos **library**. La carpeta Soluciones contiene las secuencias de comandos completas para este ejercicio.

✍ Para recuperar una fila de datos que se propone eliminar de la tabla **item**

En este procedimiento va a recuperar una fila de datos que desea eliminar de la tabla **item**. Answer_SelDelete1.sql es la secuencia de comandos completa correspondiente a este procedimiento.

1. Escriba una consulta que devuelva la fila de la tabla **item** que representa una copia en rústica (**isbn** 10101) del libro *The Cherry Orchard* (**title_no** 8).

```
USE library
SELECT *
FROM item
WHERE isbn = 10101
AND title_no = 8
GO
```

2. Ejecute la consulta para comprobar que devuelve los resultados deseados.

✍ Para eliminar una fila de datos específica de la tabla **item**

En este procedimiento va a eliminar una fila de datos específica de la tabla **item**. Answer_Delete1.sql es la secuencia de comandos completa de este procedimiento.

1. Modifique la consulta del paso 1 del procedimiento anterior para eliminar de la tabla **item** la fila que representa una copia en rústica (**isbn** 10101) del libro *The Cherry Orchard* (**title_no** 8).

```
USE library
DELETE FROM item
WHERE isbn = 10101
AND title_no = 8
GO
```

2. Ejecute la consulta y confirme que se elimina una fila de la tabla **item**.

Ejercicio 5

Uso de la instrucción UPDATE

En este ejercicio va a modificar el apellido del miembro número 507 en la tabla **member** de la base de datos **library**. La carpeta Soluciones contiene las secuencias de comandos completas para este ejercicio.

✍ Para recuperar una fila de datos que se propone actualizar de la tabla **member**

En este procedimiento va a recuperar una fila de datos que desea actualizar de la tabla **member**. `Answers_SelUpdate1.sql` es la secuencia de comandos completa para este procedimiento.

1. Escriba una consulta que recupere el apellido del miembro número 507 en la tabla **member**.

```
USE library
SELECT *
FROM member
WHERE member_no = 507
GO
```

2. Ejecute la consulta para comprobar que devuelve los resultados deseados.

✍ Para actualizar una fila de datos específica de la tabla **member**

En este procedimiento va a actualizar una fila de datos específica de la tabla **member**. `Answer_Update1.sql` es la secuencia de comandos completa de este procedimiento.

1. Escriba una consulta que cambie el apellido del miembro número 507 de la tabla **member** por otro distinto.

```
USE library
UPDATE member
SET lastname = 'BENSON'
WHERE member_no = 507
GO
```

2. Ejecute la consulta y confirme que actualiza una fila de la tabla **member**.

Ejercicio 6

Modificación de tablas basada en los datos de otras tablas

En este ejercicio va a escribir consultas que insertan valores de una o varias tablas de la base de datos en una tabla existente, y va a eliminar o actualizar las filas de una tabla en función de los criterios de otras tablas. La carpeta Soluciones contiene las secuencias de comandos completas para este ejercicio.

✍ Para agregar un nuevo miembro joven a la base de datos

En este procedimiento, agregará un nuevo miembro joven a la base de datos **library**.

1. Repase y ejecute el archivo de comandos AddJuvenile.sql para agregar un nuevo miembro joven a la base de datos **library**.

Como este proceso requiere dos instrucciones INSERT, esta acción se trata como una transacción. La instrucción SET IDENTITY_INSERT se utiliza para proporcionar un valor específico a la columna **member.member_no** en lugar de utilizar el valor que suministró la propiedad IDENTITY.

2. Ejecute la consulta y compruebe que se inserta una sola fila en las dos tablas.

✍ Para determinar los registros que se deben mover de la tabla juvenile a la tabla adult

En este procedimiento va a recuperar de las tablas **adult** e **item** los datos correspondientes a todos los miembros jóvenes de más de 18 años.

Answer_SelNewAdult.sql es la secuencia de comandos completa de este procedimiento.

1. Escriba una instrucción SELECT que devuelva la columna **member_no** de la tabla **juvenile** y las columnas **street**, **city**, **state**, **zip** y **phone_no** de la tabla **adult**. Incluya también en la consulta la fecha de hoy, más un año, mediante la siguiente expresión:

```
DATEADD( YY, 1, GETDATE() )
```

Esta última columna se utilizará posteriormente para proporcionar un valor a la columna **adult.expr_date**. Esta instrucción SELECT combina la tabla **juvenile** con la tabla **member**, de forma que **juvenile.adult_member_no** sea igual que **adult.member_no**.

Incluya una cláusula WHERE para limitar las filas agregadas a aquellos miembros de la tabla **juvenile** que sean mayores de 18 años mediante el uso de la función DATEADD en una expresión: si necesita ayuda adicional, busque “DATEADD” en los Libros en pantalla.

```
USE library
SELECT ju.member_no, ad.street, ad.city, ad.state
      ,ad.zip, ad.phone_no, DATEADD( YY, 1, GETDATE() )
FROM juvenile AS ju
INNER JOIN adult AS ad
  ON ju.adult_member_no = ad.member_no
WHERE ( DATEADD(YY, 18, ju.birth_date) < GETDATE() )
GO
```

2. Ejecute la consulta para comprobar que devuelve los resultados deseados. Observe los valores de **member_no** que se devuelven.

✎ Para insertar filas nuevas en la tabla adult procedentes de la tabla juvenile

1. Escriba una instrucción INSERT que incorpore la instrucción SELECT que creó en el paso 1 del procedimiento anterior para agregar filas a la tabla **adult**. Answer_InsNewAdult.sql es la secuencia de comandos completa de este procedimiento.

```
USE library
INSERT adult( member_no, street, city, state
              ,zip, phone_no, expr_date )
SELECT ju.member_no, ad.street, ad.city, ad.state
      ,ad.zip, ad.phone_no, DATEADD( YY, 1, GETDATE() )
FROM juvenile AS ju
INNER JOIN adult AS ad
  ON ju.adult_member_no = ad.member_no
WHERE ( DATEADD(YY, 18, ju.birth_date) < GETDATE() )
GO
```

2. Ejecute la consulta y compruebe que se inserta una sola fila.

✎ Para comprobar que un registro de juvenile determinado se agregó a la tabla adult

1. Escriba una consulta para comprobar que el miembro joven con número 16101 se ha agregado a la tabla **adult**. Answer_ChkNewAdult.sql es la secuencia de comandos completa de este procedimiento.

```
USE library
SELECT *
FROM adult
WHERE member_no = 16101
GO
```

2. Ejecute la consulta para comprobar que devuelve los resultados deseados.

Resultado

Su resultado será similar al siguiente conjunto de resultados parciales.

member_no	name	expr_date
16101	Walters, B. L.	Feb 7 1998 2:58PM

(1 fila afectada)

✎ Para determinar las filas de la tabla **juvenile** que deben quitarse

En este procedimiento va a crear una consulta que elimina de la tabla **juvenile** las filas que coinciden con las de la tabla **adult**. Cuando los miembros jóvenes se convierten en miembros adultos, deben eliminarse de la tabla **juvenile**. Answer_SelOldJuvenile.sql es la secuencia de comandos completa de este procedimiento.

1. Escriba una instrucción SELECT que combine las tablas **adult** y **juvenile**, de forma que **juvenile.member_no** sea igual que **adult.member_no**.

```
USE library
SELECT *
FROM juvenile
INNER JOIN adult
ON juvenile.member_no = adult.member_no
GO
```

2. Ejecute la consulta para comprobar que devuelve los resultados deseados.

✎ Para eliminar filas de la tabla **juvenile** que coinciden en la tabla **adult**

1. Escriba una instrucción DELETE que utilice la instrucción SELECT que creó en el paso 1 del procedimiento anterior para eliminar estas filas de la tabla **juvenile**. Answer_DelOldJuvenile.sql es la secuencia de comandos completa de este procedimiento.

```
USE library
DELETE juvenile
FROM juvenile
INNER JOIN adult
ON juvenile.member_no = adult.member_no
GO
```

2. Ejecute la consulta y compruebe que se elimina una sola fila.

✎ Para comprobar que ciertos registros se quitaron de la tabla **juvenile**

1. Escriba una instrucción SELECT para comprobar que el miembro con el número 16101 se ha quitado de la tabla **juvenile**. Answer_ChkOldJuvenile.sql es un ejemplo de esta consulta.

```
USE library
SELECT *
FROM juvenile
WHERE member_no = 16101
GO
```

2. Ejecute la consulta y compruebe que no se devuelve ningún registro.