

Capacitación SQL

Nivel inicial

Unidad 4/8

Indice general

1. Comandos para la gestión de base de datos:
 - La instrucción SQL CREATE DATABASE
 - La instrucción SQL CREATE TABLE
 - Tipos de datos de cadena:
 - Tipos de datos numéricos
 - Tipos de datos de fecha y hora
 - Crear tabla usando otra tabla
 - La instrucción SQL INSERT INTO
 - Insertar datos sólo en columnas especificadas
 - La instrucción DELETE de SQL
 - Eliminar todos los registros
 - La instrucción SQL UPDATE

1. Comandos para la gestión de base de datos:

Permiten la definición de la base de datos especificando la estructura y el tipo de los datos, así como las restricciones sobre éstos. Todo se almacenará en la base de datos.

Adicionalmente permiten la inserción, actualización, eliminación y consulta de datos.

SQL CREATE DATABASE

La instrucción CREATE DATABASE se utiliza para crear una nueva base de datos.

Sintaxis:

```
CREATE DATABASE databasename
```

Nota: Es necesario asegurarse de tener privilegios de administrador antes de crear cualquier base de datos. Una vez que se crea una base de datos, puede verificar su creación en la lista de bases de datos con el siguiente comando SQL: SHOW DATABASES

En SQL Server, el equivalente es:

```
SELECT name, database_id, create_date  
FROM sys.databases ;
```

SQLQuery1.sql - qg..._dev_user_dw (98))*

```
Select Name, Database_Id, Create_Date
FROM Sys.databases
```

100 %

Results Messages

	Name	Database_Id	Create_Date
1	master	1	2003-04-08 09:13:36.390
2	tempdb	2	2019-11-08 17:45:35.280
3	model	3	2003-04-08 09:13:36.390
4	msdb	4	2017-08-22 19:39:22.887
5	rdsadmin	5	2019-06-21 15:32:52.443
6	BOT_QG_DATA	6	2020-01-29 18:03:22.800
7	QG_SIGI_App_Data	7	2019-08-21 18:42:22.487
8	Potenciar_Test_Pentaho	8	2019-08-29 13:44:57.203
9	QG_COLPPY_STAGE	9	2020-04-21 18:26:36.277
10	QG_DW	10	2020-02-13 13:27:22.720
11	Amgen_Test	11	2019-08-29 13:45:32.670
12	QG_BOT_BI	12	2019-10-25 16:24:35.623
13	QG_DM_Comercial	13	2019-11-28 17:20:36.250
14	BOT_BI_GRAL	14	2019-12-05 12:45:03.173
15	BOT_RANDAZZO_DATA	15	2020-01-28 15:41:39.030
16	QG_Sentinel_App_Data_Fynpal	16	2020-01-10 13:39:24.090
17	QG_Sentinel_Telefonica_App_Data	17	2020-01-14 15:07:22.437
18	QG_QUALLIE_COVID	18	2020-04-06 12:16:40.680
19	QG_AdventureWorksDW2017	19	2020-05-05 17:56:22.790

SQL CREATE TABLE

La instrucción *CREATE TABLE* se usa para crear una nueva tabla en una base de datos.

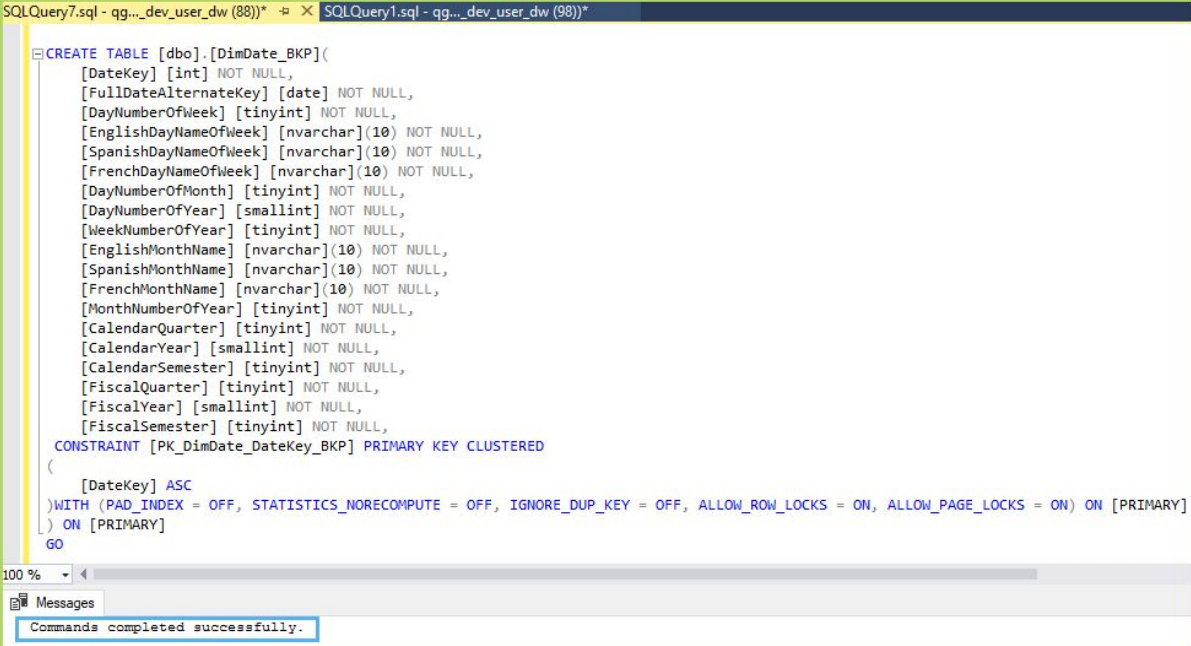
Sintaxis:

```
CREATE TABLE table_name (
    column1 datatype,
    column2 datatype,
    column3 datatype,
    ....
)
```

Los parámetros de columna especifican los nombres de las columnas de la tabla.

El parámetro datatype especifica el tipo de datos que puede contener la columna (por ejemplo, varchar, entero, fecha, etc.).

A continuación se detalla un ejemplo de la creación de una tabla denominada **"DimDate_BKP"**



```

CREATE TABLE [dbo].[DimDate_BKP](
    [DateKey] [int] NOT NULL,
    [FullDateAlternateKey] [date] NOT NULL,
    [DayNumberOfWeek] [tinyint] NOT NULL,
    [EnglishDayNameOfWeek] [nvarchar](10) NOT NULL,
    [SpanishDayNameOfWeek] [nvarchar](10) NOT NULL,
    [FrenchDayNameOfWeek] [nvarchar](10) NOT NULL,
    [DayNumberOfMonth] [tinyint] NOT NULL,
    [DayNumberOfYear] [smallint] NOT NULL,
    [WeekNumberOfYear] [tinyint] NOT NULL,
    [EnglishMonthName] [nvarchar](10) NOT NULL,
    [SpanishMonthName] [nvarchar](10) NOT NULL,
    [FrenchMonthName] [nvarchar](10) NOT NULL,
    [MonthNumberOfYear] [tinyint] NOT NULL,
    [CalendarQuarter] [tinyint] NOT NULL,
    [CalendarYear] [smallint] NOT NULL,
    [CalendarSemester] [tinyint] NOT NULL,
    [FiscalQuarter] [tinyint] NOT NULL,
    [FiscalYear] [smallint] NOT NULL,
    [FiscalSemester] [tinyint] NOT NULL,
    CONSTRAINT [PK_DimDate_DateKey_BKP] PRIMARY KEY CLUSTERED
(
    [DateKey] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
    
```

100 %

Messages

Commands completed successfully.

EC2AMAZ-JK8JGFO....dbo.DimDate_BKP SQLQuery1.sql - qg..._dev

Column Name	Data Type	Allow Nulls
DateKey	int	<input type="checkbox"/>
FullDateAlternateKey	date	<input type="checkbox"/>
DayNumberOfWeek	tinyint	<input type="checkbox"/>
EnglishDayNameOfWeek	nvarchar(10)	<input type="checkbox"/>
SpanishDayNameOfWeek	nvarchar(10)	<input type="checkbox"/>
FrenchDayNameOfWeek	nvarchar(10)	<input type="checkbox"/>
DayNumberOfMonth	tinyint	<input type="checkbox"/>
DayNumberOfYear	smallint	<input type="checkbox"/>
WeekNumberOfYear	tinyint	<input type="checkbox"/>
EnglishMonthName	nvarchar(10)	<input type="checkbox"/>
SpanishMonthName	nvarchar(10)	<input type="checkbox"/>
FrenchMonthName	nvarchar(10)	<input type="checkbox"/>
MonthNumberOfYear	tinyint	<input type="checkbox"/>
CalendarQuarter	tinyint	<input type="checkbox"/>
CalendarYear	smallint	<input type="checkbox"/>
CalendarSemester	tinyint	<input type="checkbox"/>
FiscalQuarter	tinyint	<input type="checkbox"/>
FiscalYear	smallint	<input type="checkbox"/>
FiscalSemester	tinyint	<input type="checkbox"/>

Tipos de datos de cadena:

Data Type	Descripción	Max Size	Almacenamiento
char(n)	Cadena de caracteres de ancho fijo.	8000 Caracteres	Ancho definido
varchar(n)	Cadena de caracteres de ancho variable.	8000 Caracteres	2 bytes + número de caracteres
varchar(max)	Cadena de caracteres de ancho variable.	1.073.741.824 Caracteres	2 bytes + número de caracteres
text	Cadena de caracteres de ancho variable.	2 GB de datos de texto	4 bytes + número de caracteres

nchar	Ancho fijo cadena de un sólo caracter.	4000 Caracteres	Ancho definido x 2
nvarchar	Cadena de un solo caracter de ancho variable	4000 Caracteres	
nvarchar(max)	Cadena de un solo caracter de ancho variable	536.870.912 Caracteres	
ntext	Cadena de un solo caracter de ancho variable	2 GB de datos de texto	
binary(n)	Cadena binaria de ancho fijo.	8000 bytes	
varbinary	Cadena Binaria de ancho variable	8000 bytes	
varbinary(max)	Cadena Binaria de ancho variable	2GB	
image	Cadena Binaria de ancho variable	2GB	

Tipos de datos numéricos:

bit	Entero que puede ser 0, 1 o NULL	
tinyint	Permite números enteros del 0 al 255	1 byte
smallint	Permite números enteros entre -32,768 y 32,767	2 bytes
int	Permite números enteros entre -2,147,483,648 y 2,147,483,647	4 bytes
bigint	Permite números enteros entre -9,223,372,036,854,775,808 y 9,223,372,036,854,775,807	8 bytes
decimal(p,s)		5-17 bytes
numeric(p,s)	Precisión fija y números de escala.	5-17 bytes

	Permite números de $-10^{38} + 1$ a $10^{38} - 1$. El parámetro p indica el número total máximo de dígitos que se pueden almacenar (tanto a la izquierda como a la derecha del punto decimal). p debe ser un valor de 1 a 38. El valor predeterminado es 18. El parámetro s indica el número máximo de dígitos almacenados a la derecha del punto decimal. s debe ser un valor de 0 a p. El valor predeterminado es 0	
smallmoney	Datos monetarios de -214,748.3648 a 214,748.3647	4 bytes
float(n)	Datos de números de precisión flotantes de $-1.79E + 308$ a $1.79E + 308$. El parámetro n indica si el campo debe contener 4 u 8 bytes. float (24) contiene un campo de 4 bytes y float (53) contiene un campo de 8 bytes. El valor predeterminado de n es 53.	4 u 8 bytes
real	Datos de números de precisión flotantes de $-3.40E + 38$ a $3.40E + 38$	4 bytes

Tipos de datos de fecha y hora:

Data Type	Descripción	Almacenamiento
datetime	Del 1 de enero de 1753 al 31 de diciembre de 9999 con una precisión de 3,33 milisegundos.	8 bytes
datetime2	Del 1 de enero de 0001 al 31 de diciembre de 9999 con una precisión de 100 nanosegundos	6-8 bytes
smalldatetime	Del 1 de enero de 1900 al 6 de junio de 2079 con una precisión de 1 minuto	4 bytes

date	Almacene una fecha solamente. Del 1 de enero de 0001 al 31 de diciembre de 9999	3 bytes
time	Almacene un tiempo solo con una precisión de 100 nanosegundos	3-5 bytes
datetimeoffset	Lo mismo que datetime2 con la adición de un desplazamiento de zona horaria	8-10 bytes
timestamp	Almacena un número único que se actualiza cada vez que se crea o modifica una fila. El valor de la marca de tiempo se basa en un reloj interno y no corresponde al tiempo real. Cada tabla puede tener solo una variable de marca de tiempo.	

Crear tabla usando otra tabla

También se puede crear una copia de una tabla existente usando CREATE TABLE.

La nueva tabla obtiene las mismas definiciones de columna. Se pueden seleccionar todas las columnas o columnas específicas.

Si crea una nueva tabla utilizando una tabla existente, la nueva tabla se generará con los valores existentes de la tabla anterior.

Sintaxis:

Nota: variará según el motor de base de datos que se esté utilizando

Para SQL Server:

```
SELECT column1, column2,... INTO new_table_name
FROM existing_table_name
WHERE ....
```

Otros motores:

```
CREATE TABLE new_table_name AS
SELECT column1, column2,...
FROM existing_table_name
WHERE ....
```

SQLQuery1.sql - qq...dev_user_dw (98)*

```
-- Copia todos los campos de la tabla DimDate a la nueva tabla DimDate_BKP
Select * INTO DimDate_BKP
From DimDate

-- Seleccionamos la tabla creada para ver los resultados
Select * From DimDate_BKP
```

100 %

Results Messages

	DateKey	FullDateAlternateKey	DayNumberOfWeek	EnglishDayNameOfWeek	SpanishDayNameOfWeek	FrenchDayNameOfWeek	DayNumberOfMonth	DayNumberOfYear	Week
1	20050101	2005-01-01	7	Saturday	Sábado	Samedi	1	1	1
2	20050102	2005-01-02	1	Sunday	Domingo	Dimanche	2	2	2
3	20050103	2005-01-03	2	Monday	Lunes	Lundi	3	3	2
4	20050104	2005-01-04	3	Tuesday	Martes	Mardi	4	4	2
5	20050105	2005-01-05	4	Wednesday	Miércoles	Mercredi	5	5	2
6	20050106	2005-01-06	5	Thursday	Jueves	Jeudi	6	6	2
7	20050107	2005-01-07	6	Friday	Viernes	Vendredi	7	7	2
8	20050108	2005-01-08	7	Saturday	Sábado	Samedi	8	8	2
9	20050109	2005-01-09	1	Sunday	Domingo	Dimanche	9	9	3
10	20050110	2005-01-10	2	Monday	Lunes	Lundi	10	10	3
11	20050111	2005-01-11	3	Tuesday	Martes	Mardi	11	11	3
12	20050112	2005-01-12	4	Wednesday	Miércoles	Mercredi	12	12	3
13	20050113	2005-01-13	5	Thursday	Jueves	Jeudi	13	13	3
14	20050114	2005-01-14	6	Friday	Viernes	Vendredi	14	14	3
15	20050115	2005-01-15	7	Saturday	Sábado	Samedi	15	15	3
16	20050116	2005-01-16	1	Sunday	Domingo	Dimanche	16	16	4
17	20050117	2005-01-17	2	Monday	Lunes	Lundi	17	17	4
18	20050118	2005-01-18	3	Tuesday	Martes	Mardi	18	18	4

Query executed successfully. qq-db-dev.c1kzfhyaypx.us-e... qq_dev_user_dw (98) QG_AdventureWorksDW2017 00:00:03 3652 rows

SQL INSERT INTO

La instrucción INSERT INTO, se utiliza para insertar nuevos registros en una tabla.

Sintaxis:

Es posible escribir la instrucción INSERT INTO de dos maneras.

La primera forma especifica los nombres de columna y los valores a insertar:

```
INSERT INTO table_name (column1, column2, column3, ...)
```

```
VALUES (value1, value2, value3, ...)
```

```
SQLQuery8.sql - qg..._dev_user_dw (77)* X SQLQuery1.sql - qg..._dev_user_dw (98)

INSERT INTO [dbo].[DimDate_BKP]
(
    [DateKey]
    , [FullDateAlternateKey]
    , [DayNumberOfWeek]
    , [EnglishDayNameOfWeek]
    , [SpanishDayNameOfWeek]
    , [FrenchDayNameOfWeek]
    , [DayNumberOfMonth]
    , [DayNumberOfYear]
    , [WeekNumberOfYear]
    , [EnglishMonthName]
    , [SpanishMonthName]
    , [FrenchMonthName]
    , [MonthNumberOfYear]
    , [CalendarQuarter]
    , [CalendarYear]
    , [CalendarSemester]
    , [FiscalQuarter]
    , [FiscalYear]
    , [FiscalSemester]
)
VALUES
(
    1234,
    '2020-05-05',
    5,
    'Tuesday',
    'Martes',
    'Mardi',
    5,
    5,
    1,
    'May',
    'Mayo',
    'Mai',
    2020,
    2,
    2020,
    1,
    2,
    2020,
    2
)
```

Si se están agregando valores para todas las columnas de la tabla, no necesita especificar los nombres de las columnas en la consulta SQL. Sin embargo, es necesario asegurarse que el orden de los valores esté en el mismo orden que las columnas de la tabla. La sintaxis INSERT INTO sería la siguiente:

```
INSERT INTO table_name
VALUES (value1, value2, value3, ...)
```

```

SQLQuery8.sql - qq..._dev_user_dw (77)*
INSERT INTO [dbo].[DimDate_BKP]
VALUES
    (1234,
     '2020-05-05',
     5,
     'Tuesday',
     'Martes',
     'Mardi',
     5,
     5,
     1,
     'May',
     'Mayo',
     'Mai',
     2,
     2,
     2020,
     1,
     2,
     2020,
     2)
    
```

100 %

Messages

(1 row affected)

Insertar datos sólo en columnas especificadas

También es posible insertar datos en columnas específicas.

La siguiente instrucción SQL insertará un nuevo registro, insertando solo algunas columnas:

Nota: Es necesario tener en cuenta que en los campos que no se permiten nulos, es obligatorio insertar datos. A continuación un ejemplo.

```
SQLQuery8.sql - qq..._dev_user_dw (77)* X
INSERT INTO [dbo].[DimDate_BKP]
(DateKey,
FullDateAlternateKey,
DayNumberOfWeek,
DayNumberOfMonth,
-- FiscalSemester -- Se trató de insertar valores sin este campo, lo que tiró por error que no se permiten nulos
)
VALUES
(1235,
'2020-04-12',
12,
4,
1
)

Select * from DimDate_BKP
Where DateKey = '1235'
```

100 %

Results Messages

Msg 515, Level 16, State 2, Line 2
Cannot insert the value NULL into column 'FiscalSemester', table 'QG_AdventureWorksDW2017.dbo.DimDate_BKP'; column does not allow nulls.
The statement has been terminated.

(0 rows affected)

Ejemplo sin excluir la carga de campos que no permite nulos:

```
SQLQuery8.sql - qq..._dev_user_dw (77)* X
INSERT INTO [dbo].[DimDate_BKP]
(DateKey,
FullDateAlternateKey,
DayNumberOfWeek,
DayNumberOfMonth,
FiscalSemester
)
VALUES
(1235,
'2020-04-12',
12,
4,
1
)
```

Instrucción DELETE

La instrucción DELETE se utiliza para eliminar registros existentes en una tabla. A continuación listamos el registro que deseamos eliminar.



SQLQuery8.sql - qg..._dev_user_dw (77))*

```

SELECT * FROM DimDate_BKP
WHERE DateKey = '20050201'
  
```

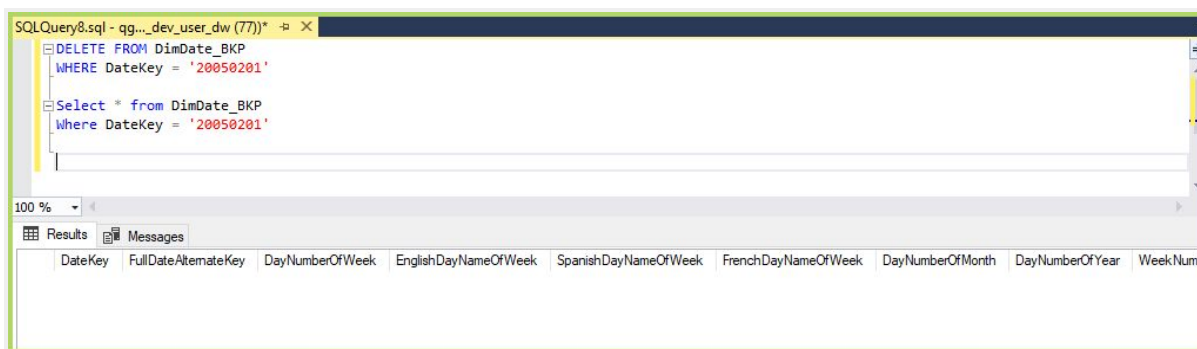
100 %

Results Messages

	DateKey	FullDateAlternateKey	DayNumberOfWeek	EnglishDayNameOfWeek	SpanishDayNameOfWeek	FrenchDayNameOfWeek	DayNumberOfMonth	DayNumberOfYear	WeekNum
1	20050201	2005-02-01	3	Tuesday	Martes	Mardi	1	32	6

Sintaxis:

```
DELETE FROM table_name WHERE condition
```



SQLQuery8.sql - qg..._dev_user_dw (77))*

```

DELETE FROM DimDate_BKP
WHERE DateKey = '20050201'

SELECT * FROM DimDate_BKP
WHERE DateKey = '20050201'
  
```

100 %

Results Messages

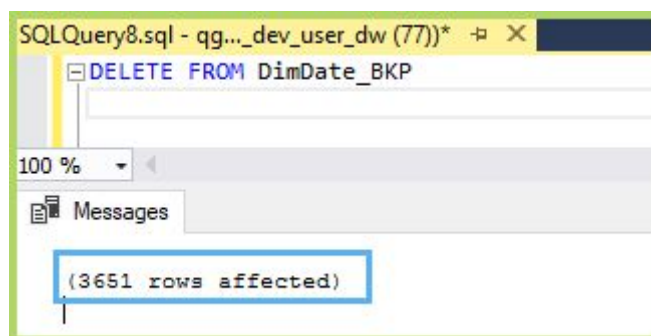
	DateKey	FullDateAlternateKey	DayNumberOfWeek	EnglishDayNameOfWeek	SpanishDayNameOfWeek	FrenchDayNameOfWeek	DayNumberOfMonth	DayNumberOfYear	WeekNum
--	---------	----------------------	-----------------	----------------------	----------------------	---------------------	------------------	-----------------	---------

Nota: Es importante tener cuidado al eliminar registros en una tabla! Observe la cláusula WHERE en la instrucción DELETE. La cláusula WHERE especifica qué registro(s) deben eliminarse. Si omite la cláusula WHERE, ¡se eliminarán todos los registros de la tabla!

Eliminar todos los registros

Es posible eliminar todas las filas de una tabla sin eliminar la tabla. Esto significa que la estructura de la tabla, los atributos y los índices estarán intactos.

```
DELETE FROM table_name
```



SQLQuery8.sql - qg..._dev_user_dw (77))*

```
DELETE FROM DimDate_BKP
```

100 %

Messages

(3651 rows affected)

Nota: En el caso que sea necesario eliminar los registros de una tabla pero no la tabla en sí, puede utilizarse la sentencia truncate. La sentencia a utilizar es la siguiente: **TRUNCATE TABLE nombretabla**. Si bien el objetivo de Delete from es el mismo que Truncate, este último tendrá mejor performance en tablas con mucha cantidad de registro, pues Delete, elimina uno a uno.

Instrucción UPDATE

La instrucción UPDATE se utiliza para modificar los registros existentes en una tabla.

Sintaxis:

```
UPDATE table_name
```

```
SET column1 = value1, column2 = value2, ...
```

```
WHERE condition
```

A continuación, un ejemplo de actualización del valor de un campo para un registro en particular de la tabla **DimDate_BKP**:

The screenshot shows a SQL query window with the following text:

```

Select * from DimDate_BKP
Where DateKey = '20050101'

Update DimDate_BKP
Set EnglishMonthName = 'May'
Where DateKey = '20050101'

Select * from DimDate_BKP
Where DateKey = '20050101'
  
```

Below the query, the 'Results' pane displays a table with the following data:

	DateKey	FullDateAlternateKey	DayNumberOfWeek	EnglishDayNameOfWeek	SpanishDayNameOfWeek	FrenchDayNameOfWeek	DayNumberOfMonth	DayNumberOfYear	WeekNumberOfYear	EnglishMonthName
1	20050101	2005-01-01	7	Saturday	Sábado	Samedi	1	1	1	January
1	20050101	2005-01-01	7	Saturday	Sábado	Samedi	1	1	1	May

Nota: ; Tenga cuidado al actualizar registros en una tabla! Observe la cláusula WHERE en la instrucción UPDATE. La cláusula WHERE especifica qué registro (s) debe actualizarse. Si omite la cláusula WHERE, se actualizarán todos los registros de la tabla.

----- *Seguí reforzando tu conocimiento en la próxima unidad del programa.