

Temas

- 1 Introducción
 - 1.1 In-Process vs. Out-of-Process Operation
 - 1.2 Archivos de formato
 - 1.3 Consideraciones de rendimiento
- 2 Importar y exportar datos con el utilitario bcp
 - 2.1 Copiar filas de una tabla en un archivo de datos utilizando conexión de confianza
 - 2.2 Copiar filas de una tabla en un archivo de datos utilizando autenticación de modo mixta
 - 2.3 Copiar datos de un archivo en una tabla
 - 2.4 Copiar una columna específica en un archivo de datos
 - 2.5 Copiar una fila específica en un archivo de datos
 - 2.6 Copiar datos de una consulta en un archivo de datos
 - 2.7 Crear un archivo de formato no XML
 - 2.8 Crear un archivo de formato XML
 - 2.9 Usar un archivo de formato para importar de forma masiva con bcp
- 3 Importación utilizando BULK INSERT
 - 3.1 Importar datos desde un archivo de datos utilizando un archivo de formato.
 - 3.2 Importar datos desde un archivo de texto
 - 3.3 Importar datos desde un archivo de datos en su formato nativo
- 4 Importar datos utilizando OPENROWSET(BULK...)
 - 4.1 Importar datos desde un archivo de datos utilizando un archivo de formato.

1 Introducción

En muchos casos se necesita transferir datos de un servidor de SQL Server u otro (que no necesariamente es SQL Server) o viceversa, esta operación puede resultar muy difícil si no se conoce las opciones que nos brinda SQL Server.

Microsoft SQL Server admite la exportación masiva de datos desde una tabla de SQL Server y la importación masiva de datos en una tabla o vista. Los métodos básicos disponibles son:

Método	Descripción	Importa datos	Exporta datos
bcp, utilidad	Utilidad de línea de comandos (Bcp.exe) que importa y exporta datos masivamente y genera archivos de formato.	Sí	Sí
Instrucción BULK INSERT	Instrucción Transact-SQL que importa datos directamente de un archivo de datos en una tabla o vista de una base de datos.	Sí	No
INSERT ... SELECT * FROM OPENROWSET(BULK...)	Instrucción Transact-SQL que usa el proveedor de conjunto de filas bulk OPENROWSET para importar masivamente datos en una tabla de SQL Server especificando la función OPENROWSET(BULK...) para seleccionar datos en una instrucción INSERT.	Sí	No

1.1 In-Process vs. Out-of-Process Operation

La instrucción BULK INSERT y la función OPENROWSET(BULK) se ejecutan in-process con SQL Server, comparten el mismo espacio de direcciones de memoria. Debido a que los archivos de datos son abiertos por un proceso de SQL Server, los datos no se copian entre el proceso del cliente y los procesos de SQL Server.

Por el contrario, la utilidad bcp se ejecuta fuera de proceso. Para mover datos entre los espacios de memoria de proceso, bcp debe utilizar el cálculo de referencias de datos entre procesos. El cálculo de referencias de datos entre procesos consiste en convertir los parámetros de una llamada de método en una secuencia de bytes. Ello puede incrementar significativamente la carga del procesador. Sin embargo, como bcp analiza los datos y los convierte en el formato de almacenamiento nativo en el proceso del cliente, pueden descargar el análisis y la conversión de datos del proceso de SQL Server. En consecuencia, si tiene restricciones de CPU, puede obtener mejor rendimiento en la importación masiva en un equipo que tenga más de una CPU o en diferentes equipos, utilizando bcp en lugar de utilizar BULK INSERT o INSERT ... SELECT * FROM OPENROWSET(BULK).

1.2 Archivos de formato

La utilidad bcp, BULK INSERT e INSERT ... SELECT * FROM OPENROWSET(BULK...) admiten el uso de un archivo de formato especializado que almacena información de formato para cada campo de un archivo de datos. El archivo de formato puede contener también información acerca de la tabla de SQL Server correspondiente. El archivo de formato se puede utilizar para proporcionar toda la información de formato necesaria para la exportación e importación masivas de datos en una instancia de SQL Server.

Los archivos de formato proporcionan una forma flexible de interpretar los datos con el formato que tienen en el archivo de datos durante la importación, así como para dar formato a los datos del archivo de datos durante la exportación. Esta flexibilidad elimina la necesidad de escribir código para propósitos especiales con el fin de interpretar los datos o volver a darles formato según los requisitos específicos de SQL Server o la aplicación externa. Por ejemplo, si va a exportar masivamente datos que se van a cargar en una aplicación que requiere valores separados por comas, puede usar un archivo de formato para insertar comas como terminadores de campo en los datos exportados.

SQL Server admite dos tipos de archivos de formato: archivos de formato XML y archivos de formato no XML. Los archivos de formato no XML se admiten en versiones anteriores de SQL Server; los archivos de formato XML son admitidos desde SQL Server 2005.

La única herramienta que puede generar un archivo de formato es la utilidad bcp.

1.3 Consideraciones de rendimiento

Las consideraciones de rendimiento pueden ser importantes si va a importar grandes cantidades de datos. En determinados casos, el rendimiento puede mejorar si modifica la forma en que una operación de importación o exportación masiva controla uno o varios de los siguientes elementos:

- Comprobación de restricciones CHECK
- Forma de registrar las transacciones masivas. Relevante para bases de datos que normalmente utilizan el modelo de recuperación completa.
- Orden de los datos exportados
- Bloqueo de tablas
- Ejecución de desencadenadores

2 Importar y exportar datos con el utilitario bcp

La utilidad bcp (Bcp.exe) es una herramienta de línea de comandos que utiliza la API de importación masiva de Bulk Copy Program o Programa de copia masiva (BCP). La utilidad bcp realiza las tareas siguientes:

- Exportaciones masivas de datos de una tabla de SQL Server a un archivo de datos.
- Exportaciones masivas de una consulta.
- Importaciones masivas de datos de un archivo de datos a una tabla de SQL Server.
- Genera archivos de formato.

```
C:\>bcp -help
usage: bcp {dbtable | query} {in | out | queryout | format} datafile
      [-m maxerrors]          [-f formatfile]          [-e errfile]
      [-F firstrow]           [-L lastrow]             [-b batchsize]
      [-n native type]        [-c character type]       [-w wide character type]
      [-N keep non-text native] [-v file format version] [-q quoted identifier]
      [-C code page specifier] [-t field terminator]    [-r row terminator]
      [-i inputfile]          [-o outfile]             [-a packetsize]
      [-S server name]        [-U username]            [-P password]
      [-T trusted connection] [-v version]             [-R regional enable]
      [-k keep null values]   [-E keep identity values]
      [-h "load hints"]       [-x generate xml format file]
      [-d database name]     [-K application intent]
```

2.1 Copiar filas de una tabla en un archivo de datos utilizando conexión de confianza

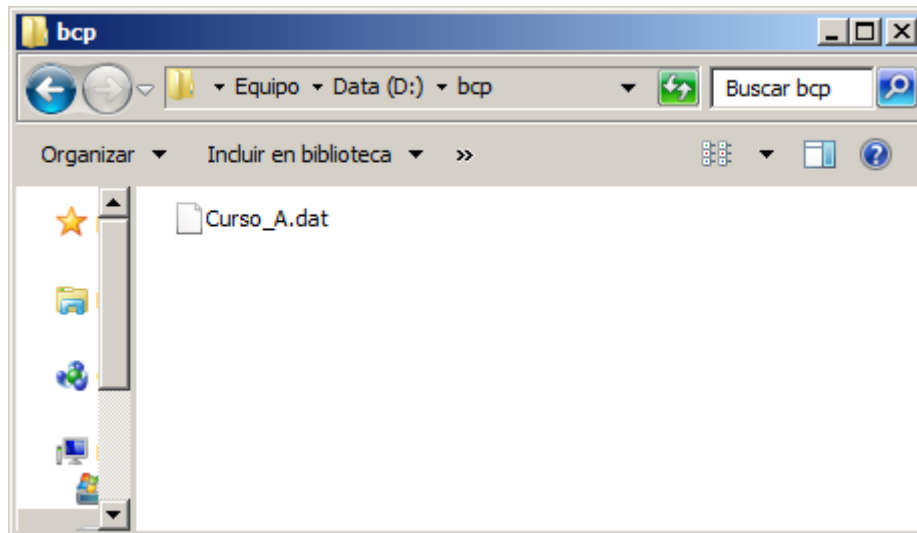
El siguiente ejemplo ilustra la opción **out** sobre la tabla EduTec.dbo.Cursor. En este ejemplo se crea un archivo de datos con el nombre Curso_A.dat y se usa el formato de caracteres para copiar los datos de la tabla en ese archivo. En el ejemplo se supone que usa la autenticación de Windows y que existe una conexión de confianza con la instancia del servidor en la que se ejecuta el comando bcp.

En la consola, escriba el siguiente comando:

```
C:\>bcp edutech.dbo.Cursor out d:\bcp\Curso_A.dat -T -c

Starting copy...

14 rows copied.
Network packet size (bytes): 4096
Clock Time (ms.) Total      : 1      Average : (14000.00 rows per sec.)
```



2.2 Copiar filas de una tabla en un archivo de datos utilizando autenticación de modo mixta

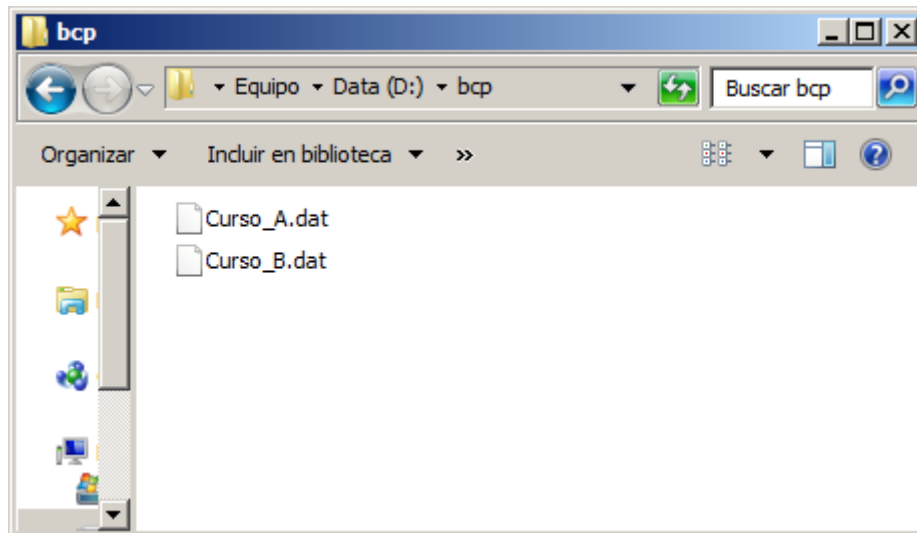
El siguiente ejemplo ilustra la opción **out** sobre la tabla EduTec.dbo.Cursor. En este ejemplo se crea un archivo de datos con el nombre Curso_B.dat y se usa el formato de caracteres para copiar los datos de la tabla en ese archivo.

En el ejemplo se supone que usa la autenticación de modo mixto; debe utilizar el modificador **-U** para especificar su identificador de inicio de sesión. Además, a menos que se esté conectando a la instancia predeterminada de SQL Server en el equipo local, use el modificador **-S** para especificar el nombre del servidor y, opcionalmente, un nombre de instancia, el modificador **-P** se utiliza para especificar la contraseña.

```
C:\>bcp edutec.dbo.Cursor out d:\bcp\Curso_B.dat -c -Usa -Psql

Starting copy...

14 rows copied.
Network packet size (bytes): 4096
Clock Time (ms.) Total      : 1      Average : (14000.00 rows per sec.)
```

**Nota**

Si no utiliza el modificar **-P** el sistema le solicitará la contraseña.

2.3 Copiar datos de un archivo en una tabla

El siguiente ejemplo ilustra la opción **in** y usa el archivo creado en el ejemplo anterior (Curso_B.dat). No obstante, en primer lugar, este ejemplo crea una copia vacía de la tabla de EduTec.dboCurso, dboCurso2, en la que se copian los datos. En el ejemplo se supone que se usa la autenticación de Windows y que existe una conexión de confianza con la instancia del servidor en la que se ejecuta el comando bcp.

Para crear la tabla vacía, en el Editor de consultas, escriba el siguiente comando:

```
USE EduTec;
GO

SELECT * INTO dbo.Curso2
FROM dbo.Curso WHERE 1=2;
GO

SELECT * FROM dbo.Curso2;
GO

IdCurso IdTarifa NomCurso
-----
(0 filas afectadas)
```

Para realizar una copia masiva de los datos de caracteres en la nueva tabla, es decir, para importar los datos, en la consola escriba el siguiente comando:

```
C:\>bcp edutec.dbo.Cursor2 in d:\bcp\Curso_B.dat -T -c

Starting copy...

14 rows copied.
Network packet size (bytes): 4096
Clock Time (ms.) Total      : 343      Average : (40.82 rows per sec.)
```

Para comprobar que el comando se ha ejecutado correctamente, muestre el contenido en el editor de consultas utilizando el siguiente comando:

```
USE EduTec;
GO

SELECT * INTO dbo.Cursor2
FROM dbo.Curso WHERE 1=2;
GO

IdCurso IdTarifa NomCurso
-----
C001    A        SO windows
C002    B        Visual Basic .NET - Fundamentos
C003    C        Visual Basic .NET - Cliente/Servidor
C004    C        Visual Basic .NET - web
C005    B        Java - POO
C006    C        Java - Cliente/Servidor
C007    C        Java - web
C008    C        Java - Frameworks
C009    B        PHP - Fundamentos
C010    C        PHP - POO
C011    C        PHP - Frameworks
C012    B        C# - POO
C013    C        C# - web Services
C014    C        Java - web Services

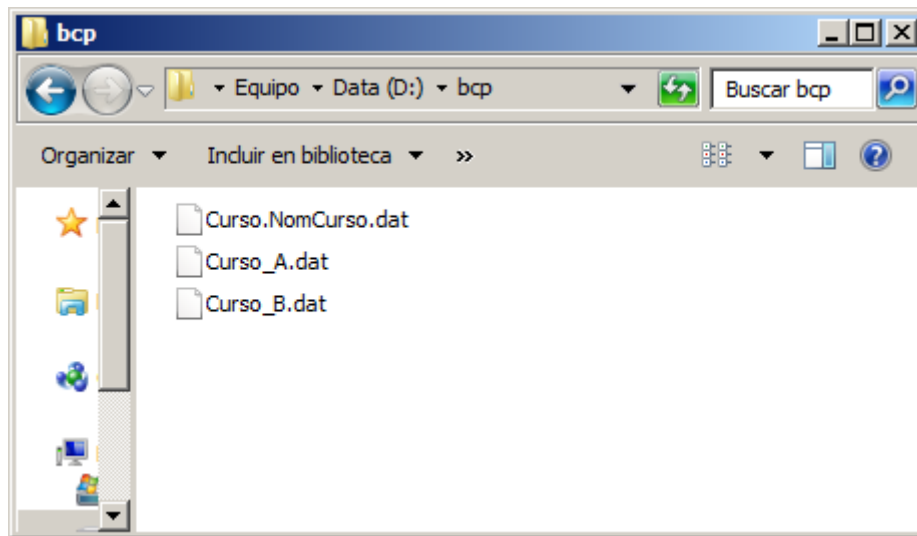
(14 filas afectadas)
```

2.4 Copiar una columna específica en un archivo de datos

Para copiar una columna específica, puede usar la opción **queryout**. El siguiente ejemplo copia únicamente la columna **NomCurso** de la tabla **dbo.Curso** en un archivo de datos. En el ejemplo se supone que usa la autenticación de Windows y que existe una conexión de confianza con la instancia del servidor en la que se ejecuta el comando **bcp**.

En la consola ejecute el siguiente comando:

```
C:\>bcp "select NomCurso from EduTec.dbo.Curso" queryout  
d:\bcp\Curso.NomCurso.dat -T -c  
  
Starting copy...  
  
14 rows copied.  
Network packet size (bytes): 4096  
Clock Time (ms.) Total      : 1      Average : (14000.00 rows per sec.)
```

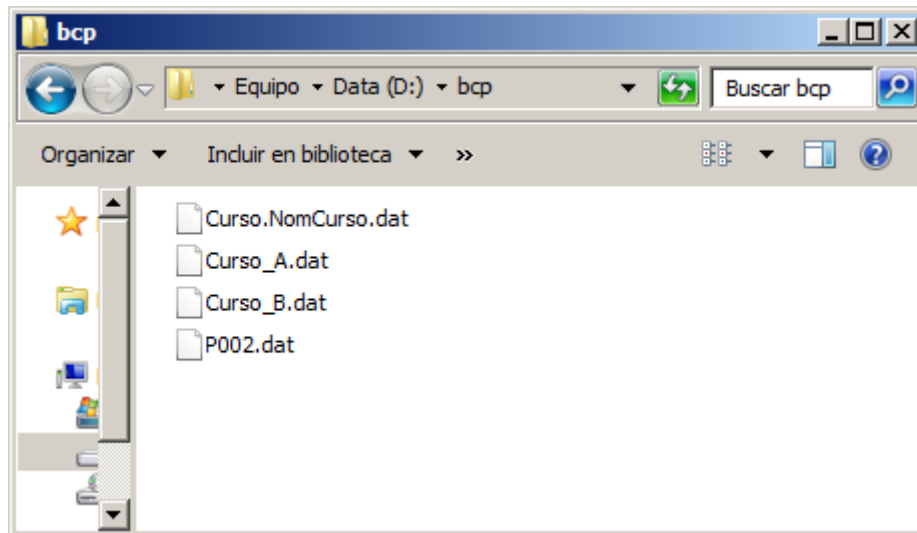


2.5 Copiar una fila específica en un archivo de datos

Para copiar una fila específica, se puede usar la opción **queryout**. En el siguiente ejemplo se copia únicamente la fila correspondiente al profesor de código P002 de la tabla EduTec.dbo.Profesor en un archivo de datos de nombre P002.dat. En el ejemplo se supone que usa la autenticación de Windows y que existe una conexión de confianza con la instancia de servidor en la que se ejecuta el comando bcp.

En la consola ejecute el siguiente comando:

```
C:\>bcp "select * from EduTec.dbo.Profesor where IdProfesor='P002'" queryout  
d:\bcp\P002.dat -T -c  
  
Starting copy...  
  
1 rows copied.  
Network packet size (bytes): 4096  
Clock Time (ms.) Total      : 31      Average : (32.26 rows per sec.)
```

2.6 Copiar datos de una consulta en un archivo de datos

Para copiar el conjunto de resultados de una instrucción Transact-SQL en un archivo de datos, use la opción **queryout**. El siguiente ejemplo copia los nombres de la tabla de EduTec.dbo.Profesor, ordenados por el apellido y después por el nombre, en el archivo de datos Profesores.txt. En el ejemplo se supone que usa la autenticación de Windows y que existe una conexión de confianza con la instancia del servidor en la que se ejecuta el comando bcp.

En la consola ejecute el siguiente comando:

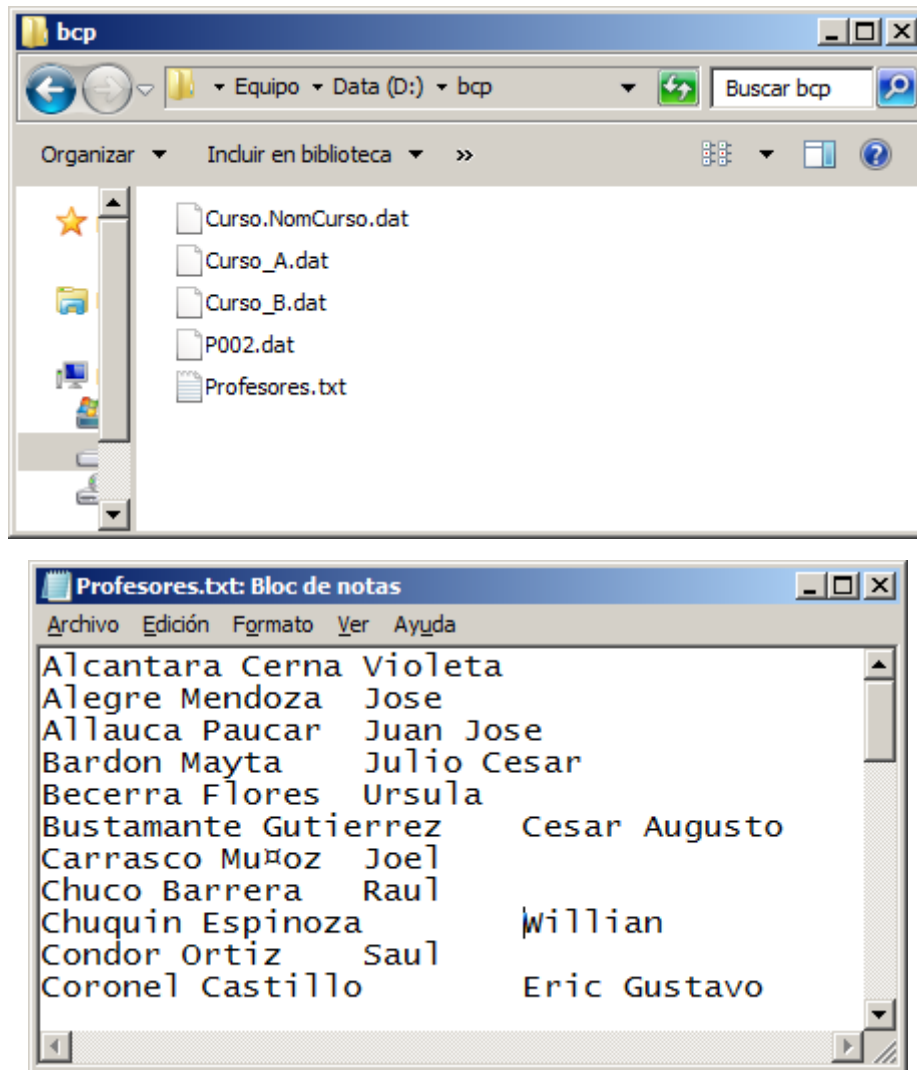
```
C:\>bcp "select ApeProfesor, NomProfesor from EduTec.dbo.Profesor order by 1, 2" queryout d:\bcp\Profesores.txt -T -c
```

```
Starting copy...
```

```
42 rows copied.
```

```
Network packet size (bytes): 4096
```

```
Clock Time (ms.) Total      : 31      Average : (1354.84 rows per sec.)
```

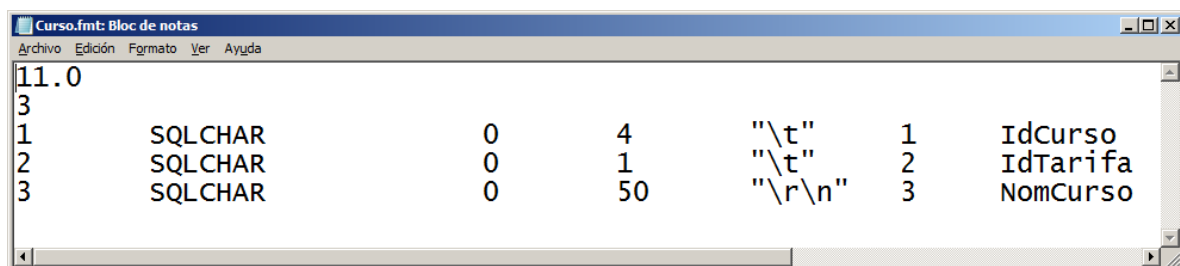
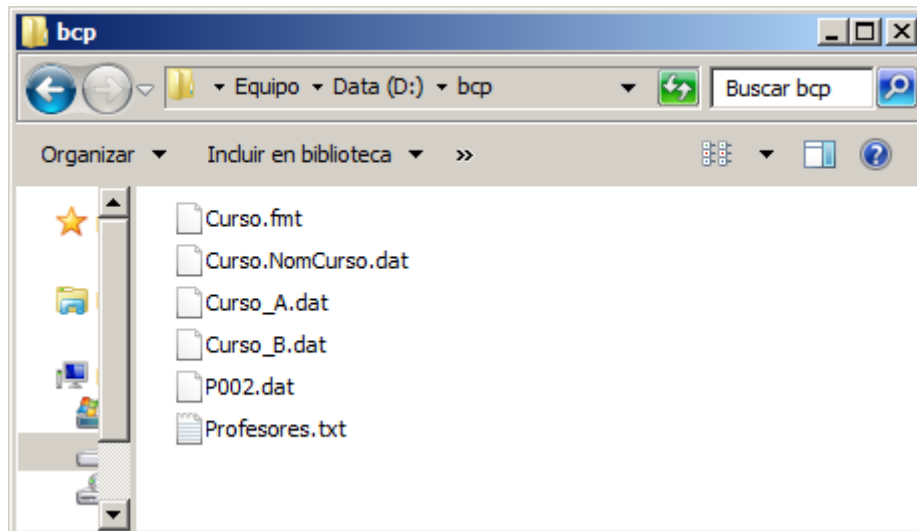


2.7 Crear un archivo de formato no XML

En el siguiente ejemplo se crea un archivo de formato no XML, Curso.fmt, para la tabla dbo.Curso de la base de datos EduTec. En el ejemplo se supone que usa la autenticación de Windows y que existe una conexión de confianza con la instancia del servidor en la que se ejecuta el comando bcp.

En la consola ejecute el siguiente comando:

```
C:\>bcp EduTec.dbo.Curso format nul -T -c -f d:\bcp\Curso.fmt
```

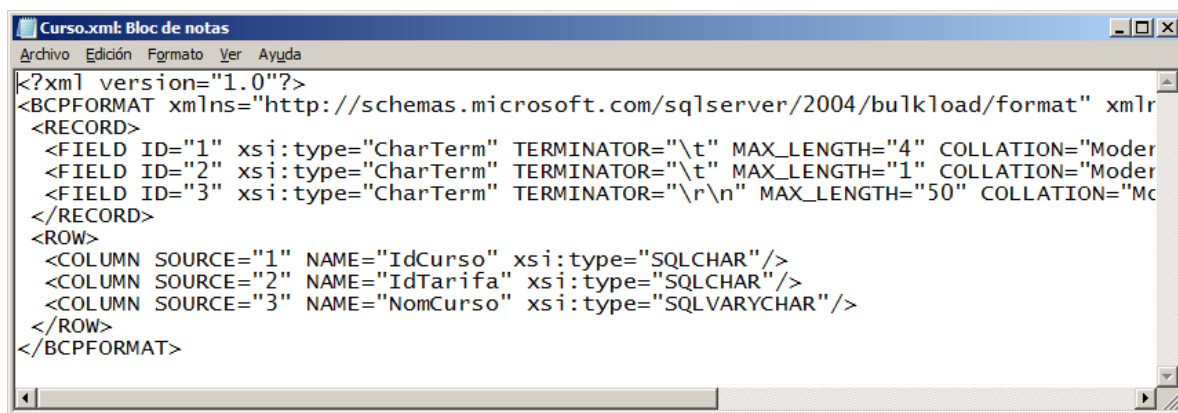
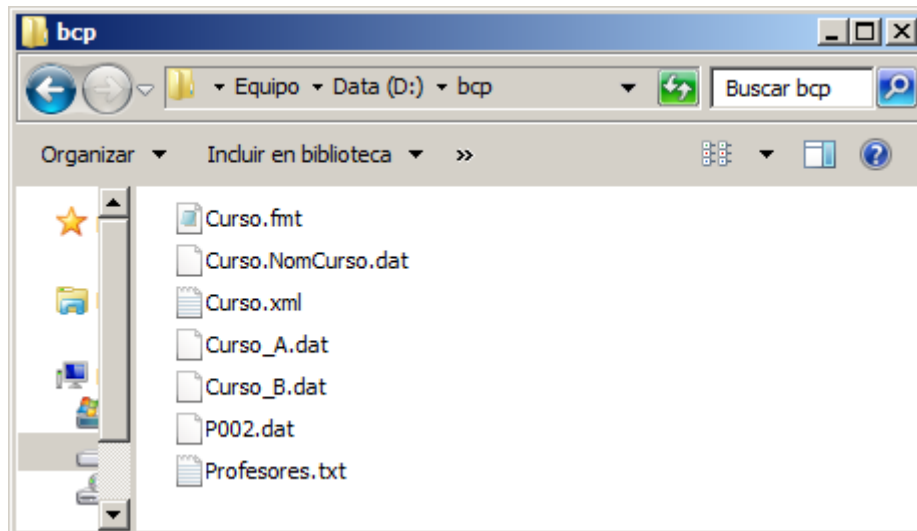


2.8 Crear un archivo de formato XML

El siguiente ejemplo crea un archivo de formato XML con el nombre Curso.xml para la tabla dbo.Curso de la base de datos EduTec. En el ejemplo se supone que usa la autenticación de Windows y que existe una conexión de confianza con la instancia del servidor en la que se ejecuta el comando bcp.

En la consola ejecute el siguiente comando:

```
C:\>bcp EduTec.dbo.Curso format nul -T -c -x -f d:\bcp\Curso.xml
```



2.9 Usar un archivo de formato para importar de forma masiva con bcp

Para usar un archivo de formato creado anteriormente al importar datos a una instancia de SQL Server, se debe usar el modificador **-f** con la opción **in**. Por ejemplo, el siguiente comando copia de forma masiva el contenido del archivo de datos, `Curso_A.dat`, en una copia de la tabla `dbo.Curso` de nombre `dbo.Curso3` mediante el archivo de formato creado anteriormente (`Curso.xml`). En el ejemplo se supone que usa la autenticación de Windows y que existe una conexión de confianza con la instancia del servidor en la que se ejecuta el comando `bcp`.

Primero crearemos la tabla `dbo.Curso3`:

```
USE EduTec;
go

SELECT * INTO dbo.Curso3
FROM dbo.Curso WHERE 1=2;
GO

select * from dbo.Curso3;
```

```
go
```

```
IdCurso IdTarifa NomCurso
```

```
-----
```

```
(0 filas afectadas)
```

En el símbolo del sistema de Windows, escriba:

```
C:\>bcp EduTec.dbo.Curso3 in D:\bcp\Curso_B.dat -T -f D:\bcp\Curso.xml
```

```
Starting copy...
```

```
14 rows copied.
```

```
Network packet size (bytes): 4096
```

```
Clock Time (ms.) Total      : 234      Average : (59.83 rows per sec.)
```

Ahora, consultamos la tabla dbo.Curso3:

```
select * from dbo.Curso3;
```

```
go
```

```
IdCurso IdTarifa NomCurso
```

```
-----
```

```
C001    A      SO windows
C002    B      Visual Basic .NET - Fundamentos
C003    C      Visual Basic .NET - Cliente/Servidor
C004    C      Visual Basic .NET - web
C005    B      Java - POO
C006    C      Java - Cliente/Servidor
C007    C      Java - web
C008    C      Java - Frameworks
C009    B      PHP - Fundamentos
C010    C      PHP - POO
C011    C      PHP - Frameworks
C012    B      C# - POO
C013    C      C# - web Services
C014    C      Java - web Services
```

```
(14 filas afectadas)
```

3 Importación utilizando BULK INSERT

BULK INSERT carga datos de un archivo de datos a una tabla. Esta funcionalidad es parecida a la que ofrece la opción **in** del comando **bcp**, aunque el que lee el archivo de datos es el proceso de SQL Server.

3.1 Importar datos desde un archivo de datos utilizando un archivo de formato.

Utilizaremos el archivo de formato Curso.xml para importar el archivo de datos de Curso_A.dat en la tabla dbo.Curso4.

Primero crearemos la tabla dbo.Curso4:

```
USE EduTec;
go

SELECT * INTO dbo.Curso4
FROM dbo.Curso WHERE 1=2;
GO

select * from dbo.Curso4;
go

IdCurso IdTarifa NomCurso
-----
(0 filas afectadas)
```

Ahora, procederemos a importar el archivo de datos:

```
bulk insert EduTec.dbo.Curso4
from 'D:\bcp\Curso_A.dat'
with (formatfile='D:\bcp\Curso.xml');

select * from dbo.Curso4;
go

IdCurso IdTarifa NomCurso
-----
C001     A       SO windows
C002     B       Visual Basic .NET - Fundamentos
C003     C       Visual Basic .NET - Cliente/Servidor
C004     C       Visual Basic .NET - Web
C005     B       Java - POO
C006     C       Java - Cliente/Servidor
C007     C       Java - Web
```

```

C008    C      Java - Frameworks
C009    B      PHP - Fundamentos
C010    C      PHP - POO
C011    C      PHP - Frameworks
C012    B      C# - POO
C013    C      C# - Web Services
C014    C      Java - Web Services

```

(14 filas afectadas)

3.2 Importar datos desde un archivo de texto

En este caso importaremos datos desde Profesores.txt en la tabla dbo.Profesor2.

Primero crearemos la tabla dbo.Profesor2:

```

select NomProfesor, ApeProfesor
into dbo.Profesor2
from dbo.Profesor
where 1=2;
(0 filas afectadas)

```

Ahora, procedemos a importar los datos:

```

bulk insert dbo.Profesor2
from 'D:\bcp\Profesores.txt'
with (fieldterminator='\t',rowterminator='\n');
go

```

```

select * from dbo.Profesor2;
go

```

NomProfesor	ApeProfesor
Alcantara Cerna	Violeta
Alegre Mendoza	Jose
Allauca Paucar	Juan Jose
Bardon Mayta	Julio Cesar
. . .	
. . .	

3.3 Importar datos desde un archivo de datos en su formato nativo

Importaremos los datos desde el archivo Curso_nat.dat en la tabla dbo.Curso5.

En primer lugar crearemos la tabla dbo.Curso5:

```
SELECT * INTO dbo.Curso5
FROM dbo.Curso WHERE 1=2;
GO
```

```
select * from dbo.Curso5;
go
```

```
IdCurso IdTarifa NomCurso
-----
```

```
(0 filas afectadas)
```

Se procederá a crear el archivo con formato nativo utilizando el utilitario bcp:

```
C:\>bcp EduTec.dbo.Curso out D:\bcp\Curso-nat.Dat -n -T
```

```
Starting copy...
```

```
14 rows copied.
```

```
Network packet size (bytes): 4096
```

```
Clock Time (ms.) Total      : 31      Average : (451.61 rows per sec.)
```

Ahora, procedemos a ejecutar la importación:

```
BULK INSERT dbo.Curso5
FROM 'D:\bcp\Curso-nat.Dat'
WITH (DATAFILETYPE='native');
GO
```

```
(14 filas afectadas)
```

4 Importar datos utilizando OPENROWSET(BULK...)

Se utiliza para importaciones masivas de datos. Su sintaxis básica es:

```
INSERT ... SELECT * FROM OPENROWSET(BULK...)
```

4.1 Importar datos desde un archivo de datos utilizando un archivo de formato.

Utilizaremos el archivo de formato Curso.fmt para importar el archivo de datos de Curso_A.dat en la tabla dbo.Curso6.

Primero crearemos la tabla dbo.Curso6:

```
SELECT * INTO dbo.Curso6  
FROM dbo.Curso WHERE 1=2;  
GO
```

(0 filas afectadas)

Ahora, procederemos a ejecutar la importación:

```
INSERT INTO dbo.Curso6  
SELECT a.* FROM OPENROWSET( BULK 'D:\bcp\Curso_A.dat',  
    FORMATFILE = 'D:\bcp\Curso.fmt') AS a;
```

(14 filas afectadas)