Unidad 2

Importación de Archivos

Bases de Datos Aplicada

v1.0 – Abril 2024

Universidad Nacional de La Matanza

DIIT
Departamento de Ingeniería e Investigaciones Tecnológicas

Importar un archivo CSV

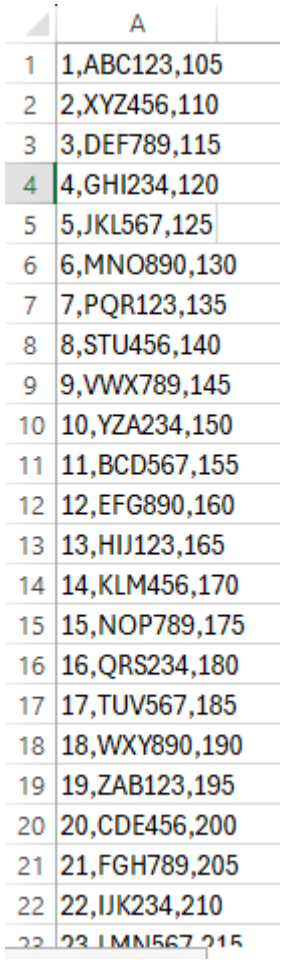
```
USE TestingDB
GO

DROP TABLE IF EXISTS multas
GO
CREATE TABLE multas (ID_Multa int primary key, Patente NVARCHAR(50), Velocidad int);
GO

BULK INSERT TestingDB..multas
FROM 'C:\Importar\multas.csv'
WITH
(
    FIELDTERMINATOR = ',',  -- Especifica el delimitador de campo (coma en un archivo CSV)
    ROWTERMINATOR = '\n',   -- Especifica el terminador de fila (salto de línea en un archivo CSV)
    CODEPAGE = 'ACP'        -- Especifica la página de códigos del archivo
)
GO
SELECT *
FROM multas
```

Importar un archivo CSV

Dado el archivo “multas” de formato CSV ubicado en la ruta 'C:\Importar\multas.csv’, obtenemos la siguiente tabla luego de ejecutar la query.



	A
1	1,ABC123,105
2	2,XYZ456,110
3	3,DEF789,115
4	4,GHI234,120
5	5,JKL567,125
6	6,MNO890,130
7	7,PQR123,135
8	8,STU456,140
9	9,VWX789,145
10	10,YZA234,150
11	11,BCD567,155
12	12,EFG890,160
13	13,HIJ123,165
14	14,KLM456,170
15	15,NOP789,175
16	16,QRS234,180
17	17,TUV567,185
18	18,WXY890,190
19	19,ZAB123,195
20	20,CDE456,200
21	21,FGH789,205
22	22,IJK234,210
23	23,LMN567,215

Results Messages

	ID_Multa	Patente	Velocidad
1	1	ABC123	105
2	2	XYZ456	110
3	3	DEF789	115
4	4	GHI234	120
5	5	JKL567	125
6	6	MNO890	130
7	7	PQR123	135
8	8	STU456	140
9	9	VWX789	145
10	10	YZA234	150
11	11	BCD567	155
12	12	EFG890	160
13	13	HIJ123	165
14	14	KLM456	170
15	15	NOP789	175
16	16	QRS234	180
17	17	TUV567	185
18	18	WXY890	190
19	19	ZAB123	195
20	20	CDE456	200
21	21	FGH789	205
22	22	IJK234	210
23	23	LMN567	215

Importar un archivo CSV

```
DROP TABLE IF EXISTS multas
GO
CREATE TABLE multas (ID_Multa int primary key, Patente
NVARCHAR(50), Velocidad int);
GO
```

- **DROP TABLE IF EXISTS multas:** Si la tabla existe la elimina, el IF EXISTS nos previene de un error en caso de que la tabla no exista.
- Es muy importante que la estructura de la tabla en la base de datos coincida con la estructura de los datos en el archivo. Considerando que coincida el orden y el tipo de datos de las columnas, o si la tabla tiene una clave primaria o restricciones de unicidad los datos deben respetarlas.

Fuente: https://learn.microsoft.com/en-us/sql/t-sql/statements/bulk-insert-transact-sql?view=sql-server-ver16

Importar un archivo CSV

- **BULK INSERT:** Es una operación que importa un archivo de datos a una tabla de base de datos o vista en un formato especificado por el usuario.

```
BULK INSERT  database_name.schema_name.table_or_view_name
FROM 'data_file'
WITH
(
    FIELDTERMINATOR = ',',
    ROWTERMINATOR = '\n',
   CODEPAGE = 'ACP’,
    FIRSTROW = '1'
);
```

Fuente: https://learn.microsoft.com/en-us/sql/t-sql/statements/bulk-insert-transact-sql?view=sql-server-ver16

Importar un archivo CSV

- Si no se especifica el nombre de la base de datos, se usa la actual. Lo mismo con el esquema, son opcionales.
- **DATA_FILE:** la ruta completa del archivo que quiero importar. Por ej: '\\SystemX\DiskZ\Sales\data\orders.dat'
- **FIELDTERMINATOR:** Especifica el terminador de campo que se utilizará para archivos de datos char y Widechar.
  El terminador de campo predeterminado es \t(carácter de tabulación).
- **ROWTERMINATOR:** Especifica el terminador de fila que se utilizará para archivos de datos char y Widechar.
  El terminador de fila predeterminado es \r\n(carácter de nueva línea).
- **CODEPAGE:** Especifica la página de códigos a usar para la interpretación de los caracteres en el archivo de datos. 'ACP' significa "Ansi Code Page". La opciones son 'ACP', 'OEM', 'RAW', 'code_page'(especifica el numero de codepage)
- **FIRSTROW:** En el caso de que existan encabezados se debe especificar en que fila empiezan los datos.

Fuente: https://learn.microsoft.com/en-us/sql/t-sql/statements/bulk-insert-transact-sql?view=sql-server-ver16

Importar un Archivo JSON

```
USE TestingDB
GO

DROP TABLE IF EXISTS multas
GO
CREATE TABLE multas (ID_Multa int primary key, Patente NVARCHAR(50), Velocidad int);
GO

-- Utilizar OPENROWSET con OPENJSON para leer el archivo JSON y cargar los datos en la tabla
INSERT INTO multas (ID_Multa, Patente, Velocidad)
SELECT ID_Multa, Patente, Velocidad
FROM OPENROWSET (BULK 'C:\Importar\infomultas.json', SINGLE_CLOB) as j
CROSS APPLY OPENJSON(BulkColumn)
WITH (
    ID_Multa INT '$.ID_Multa',
    Patente NVARCHAR(50) '$.Patente',
    Velocidad INT '$.Velocidad'
);
GO
SELECT *
FROM multas
```

Importar un Archivo JSON

infomultas.json: Bloc de notas

Archivo Edición Formato Ver Ayuda

```
[
  {
    "ID_Multa": 1,
    "Patente": "ABC123",
    "Velocidad": 105
  },
  {
    "ID_Multa": 2,
    "Patente": "XYZ456",
    "Velocidad": 110
  },
  {
    "ID_Multa": 3,
    "Patente": "DEF789",
    "Velocidad": 115
  },
  {
    "ID_Multa": 4,
    "Patente": "GHI234",
    "Velocidad": 120
  },
  {
    "ID_Multa": 5,
    "Patente": "JKL567",
    "Velocidad": 125
  },
  {
    "ID_Multa": 6,
    "Patente": "MNO890",
    "Velocidad": 130
  },
  {
    "ID_Multa": 7,
    "Patente": "PQR123",
    "Velocidad": 135
  },
  {
```

Dado el archivo “infomultas” ubicado en la ruta 'C:\Importar\infomultas.json’ obtenemos la tabla multas.

Results Messages

	ID_Multa	Patente	Velocidad
1	1	ABC123	105
2	2	XYZ456	110
3	3	DEF789	115
4	4	GHI234	120
5	5	JKL567	125
6	6	MNO8...	130
7	7	PQR1...	135
8	8	STU456	140
9	9	VWX7...	145
10	10	YZA234	150
11	11	BCD567	155
12	12	EFG890	160
13	13	HIJ123	165
14	14	KLM456	170
15	15	NOP7...	175
16	16	QRS2...	180
17	17	TUV567	185
18	18	WXY8...	190
19	19	ZAB123	195

Importar un Archivo JSON

```
FROM OPENROWSET (BULK 'C:\Importar\infomultas.json', SINGLE_CLOB) as j
CROSS APPLY OPENJSON(BulkColumn)
WITH (
    ID_Multa INT '$.ID_Multa',
    Patente NVARCHAR(50) '$.Patente',
    Velocidad INT '$.Velocidad'
);
```

- **OPENROWSET(BULK...)** se utiliza para acceder a archivos externos.
- **SINGLE_CLOB:** indica que el archivo se debe leer como un objeto grande de caracteres. Por eso es ideal para archivos JSON, donde todo el contenido se trata como una sola cadena grande.
- **CROSS APPLY** funciona de manera similar a un JOIN, pero se usa para aplicar una función table-valued (que devuelve una tabla) a cada fila que resulta de la consulta a la que se aplica.

Importar un Archivo JSON

```
FROM OPENROWSET (BULK 'C:\Importar\infomultas.json', SINGLE_CLOB) as j
CROSS APPLY OPENJSON(BulkColumn)
WITH (
    ID_Multa INT '$.ID_Multa',
    Patente NVARCHAR(50) '$.Patente',
    Velocidad INT '$.Velocidad'
);
```


- **OPENJSON** es una función que transforma un texto JSON en una tabla relacional. Es decir, convierte valores JSON a los tipos que se especifican en la cláusula WITH.
- **$:** se usa para referirse a la raíz del objeto JSON, y luego se especifican las propiedades del objeto (en este caso ID_Multa, Patente, y Velocidad. Estas propiedades del JSON se mapean directamente a las columnas correspondientes en la tabla “multas”.

¿Dudas?

Universidad Nacional
de La Matanza

DIIT
Departamento de Ingeniería e
Investigaciones Tecnológicas