

Importar tablas desde ficheros de texto delimitados.

Prácticamente todos los sistemas de gestión de bases de datos tienen la capacidad de importar y exportar tablas a ficheros de texto (ASCII) de campos delimitados. Por lo tanto esta utilidad es muy aconsejable para pasar datos de un SGBD a otro además de para hacer copias de seguridad de tablas individuales.

Un ejemplo de fichero de texto de campos delimitados podría ser el siguiente:

```
1112345;"Martínez Salas,;Fernando";"PROFESOR";2200.00;10
4123005;"Bueno Zarco,;Elisa";"PROFESOR";2200.00;10
4122025;"Montes García, M.Pilar";"PROFESOR";2200.00;10
1112346;"Rivera Silvestre, Ana";"PROFESOR";2050.00;15
9800990;"Ramos Ruiz, Luis";"PROFESOR";2050.00;15
8660990;"De Lucas Fdez, M.Angel";"PROFESOR";2050.00;15
7650000;"Ruiz Lafuente, Manuel";"PROFESOR";2200.00;22
43526789;"Serrano Laguía, María";"PROFESOR";2050.00;45
4480099;"Ruano Cerezo, Manuel";"ADMINISTRATIVO";1800.00;10
1002345;"Albarrán Serrano, Alicia";"ADMINISTRATIVO";1800.00;15
7002660;"Muñoz Rey, Felicia";"ADMINISTRATIVO";1800.00;15
5502678;"Marín Marín, Pedro";"ADMINISTRATIVO";1800.00;22
6600980;"Peinado Gil, Elena";"CONSERJE";1750.00;22
4163222;"Sarro Molina, Carmen";"CONSERJE";1750.00;45
```

Fichero personal.txt

Estos ficheros se caracterizan por estar almacenados en formato ASCII habitualmente con las extensiones .txt o .csv. Tienen los registros de longitud variable y los campos están delimitados por un carácter especial (en este caso “;”). También habitualmente los campos de caracteres vienen entrecomillados.

MySQL proporciona dos métodos para importar directamente este tipo de ficheros a tablas. Una de ellas es la utilidad *mysqlimport*, el otro método es lanzar la sentencia `LOAD DATA INFILE`. En este apartado se comenta únicamente este último método. La utilidad *mysqlimport* se puede ver en detalle en la sección 8.10 del Manual de MySQL.

Sintaxis genérica de la sentencia `LOAD DATA INFILE` :

```
LOAD DATA [LOCAL] INFILE 'path/nombre_fichero.txt'
[REPLACE | IGNORE]
INTO TABLE nombre_tabla
[FIELDS
  [TERMINATED BY '\t']
  [[OPTIONALLY] ENCLOSED BY '']
  [ESCAPED BY '\\\ ' ]
]
[LINES
  [STARTING BY '']
  [TERMINATED BY '\n']
]
[IGNORE number LINES]
[(col_name,...)]
```

Se observa en esta sintaxis que la mayoría de las cláusulas de esta sentencia son optativas, vamos a comentar las más interesantes.

Si se especifica la cláusula `LOCAL` significa que el fichero es leído por el programa cliente en el equipo donde se está ejecutando el cliente de MySQL desde el que se está

lanzando la sentencia `LOAD DATA`. Si la cláusula `LOCAL` se omite significa que el fichero debe encontrarse en el host del servidor MySQL .

Para poder ejecutar esta sentencia es necesario crear previamente la tabla sobre la que van a ir los datos. La tabla puede o no estar vacía. Habrá que tener cuidado al hacer una carga masiva de datos con esta sentencia el evitar entradas duplicadas en campos marcados como `UNIQUE` o `PRIMARY KEY` .

En el siguiente ejemplo se va a crear una tabla y lanzar la sentencia `LOAD DATA` que permita cargar los datos del fichero de personal (personal.txt) anterior.

```
USE TEST;
CREATE TABLE IF NOT EXISTS personal(
    dni            int(10),
    apellidos      varchar(30),
    funcion        enum('PROFESOR','ADMINISTRATIVO','CONSERJE'),
    salario        float(6,2),
    cod_centros    int(5),
    CONSTRAINT pk_codigo PRIMARY KEY (dni));

LOAD DATA INFILE 'C:\\temp\\personal.txt'
INTO TABLE personal
FIELDS TERMINATED BY ';' ENCLOSED BY '"'
LINES TERMINATED BY '\r\n';
```

Muy importante es construir correctamente las cláusulas `FIELDS` y `LINES` de la sentencia `LOAD DATA`. En la cláusula `FIELDS` indicamos que el carácter delimitador es el “;” y que las cadenas de texto van entrecomilladas. En la cláusula `LINES` indicamos que el carácter de final de línea es ‘\r\n’. (Los ficheros de texto de Windows suelen llevar \r\n como carácter de final de línea mientras que los ficheros de texto en Linux es únicamente \n)

Para poder lanzar una sentencia `LOAD DATA` es necesario ser administrador de la base de datos o al menos tener concedido el privilegio `FILE` .

Exportar tablas a ficheros de texto delimitados

La sentencia que permite exportar tablas a ficheros ASCII delimitados tiene la siguiente sintaxis

```
SELECT columns INTO OUTFILE 'path/nombre_fichero.txt'
[ FIELDS
    [ TERMINATED BY '\t' ]
    [[ OPTIONALLY ] ENCLOSED BY ' ' ]
    [ ESCAPED BY '\\' ]
]
[ LINES
    [ STARTING BY ' ' ]
    [ TERMINATED BY '\n' ]
]
FROM nombre_tabla;
```

A diferencia de `LOAD DATA`... la sentencia `SELECT...INTO OUTFILE` siempre genera el fichero en el host donde esta ejecutándose el servidor. Si lo que se desea es que el fichero se cree en el equipo del cliente esta sentencia no nos sirve. Podemos usar en su lugar el siguiente comando

```
shell> mysql -e "SELECT ..." > nombre_fichero
```

En el siguiente ejemplo se va a crear un fichero llamado `personal2.txt` a partir de la tabla a tabla `personal` del punto anterior.

```
SELECT * INTO OUTFILE 'C:\\temp\\personal2.txt'
FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"'
LINES TERMINATED BY '\r\n'
FROM personal;
```

Al igual que ocurre con `LOAD DATA` para poder lanzar una sentencia `SELECT ... INTO OUTFILE` es necesario ser administrador de la base de datos o al menos tener concedido el privilegio `FILE`.