

CONNECT CSV Table Type:

CONNECT CSV Table Type Syntax:

CREATE TABLE Nombre_Tabela

(
 Nombre_ColumnaN | VARCHAR(N°) | NOT NULL | ((FLAG = N°))
 | INT(N°) | NULL |
 | [...] | [...] |
 [...])

Nombre_ColumnaN | DATE | date-format = 'YYYY-MM-DD' | ((FLAG = N°))
 | DATETIME | date-format = 'YYYY-MM-DD HH:MM:SS' |
 [...])

ENGINE = CONNECT

TABLE-TYPE = CSV

FILE-NAME = ' C:\Nombre-Carpeta-Rais\Nombre-Subcarpeta\Nombre-Archivo.
 Nombre-Archivo.txt | .csv | .zip | .txt |
 C:/Nombre-Carpeta-Rais/Nombre-Subcarpeta/Nombre-Archivo.txt | .zip |
 C:/Nombre-Carpeta-Rais/Nombre-Subcarpeta/Nombre-Archivo.* | .csv | .zip | .txt |

HEADER = 1 // YES
 0 // NO

SEP-CHAR = ','

QUCHAR = ''' '''
 [...]]

QUOTED = 0
 1
 2
 3
 -1

MULTIPLE = 3
 1 // YES
 0 // NO

(*) Multiple = 3 is to retrieve multiple files with subfolders option enabled. (An equivalent of C:/.../Nombre-Archivo.csv)

ZIPPED = 1 // YES
 0 // NO

OPTION_LIST = 'maxlen = N°, accept = 1 // YES
 0 // NO (Entry = Nombre-Archivo-Dentro-ZIP.csv | Nombre-Archivo-Dentro-ZIP*.csv |) [...]

[...] multientry = 1 // YES
 0 // NO , load = 'C:/ [...] / Nombre-Archivo-Cargado.csv | C:/ [...] / Nombre-Archivo-Cargado*.csv |

[...] append = 1 // YES
 0 // NO

FILE-NAME = ' C:/ [...] / Nombre-Archivo-ZIP-Creado.zip '

READONLY = 1

(#) zipped File Tables

CONNECT CSV Table Type Syntax: (...) (Cont.)

Characteristics:

▶ HEADER: "Used to specify if the CSV file first record is the list of column names, which are skipped if set to HEADER = 1"

{ (*) NOTE: On writing, if the file is empty, the column names are automatically written.

▶ FLAG: "The flag column option is the rank of the column into the file starting from 1 for the leftmost column (For CSV Tables). This is to enable:

- ▶ Having columns displayed in a different order than in the file.
- ▶ And/or to define the table specifying only some columns of the CSV file."

▶ QUOTED: "It is used to indicate the level of quoting. values can be:

▶ -1: "No quoting".

▶ 0: "The fields between quotes are read, and the quotes discarded. On writing, fields will be quoted only if they contain the separator character or begin with the quoting character. If they contain the quoting character, it will be doubled."

▶ 1: "Only Text fields will be written between quotes, except NULL fields. This includes also the column names of an eventual header."

▶ 2: "All fields will be written between quotes, except NULL fields."

▶ 3: "All fields will be written between quotes, including NULL fields."

▶ QCHAR: "Used to specify the eventual quoting character."

{ (*) NOTE: If only QCHAR option is specified, the QUOTED option will be default to 1.

▶ SEP_CHAR: "Used to specify the separator character between fields. By default: ','"

{ (*) NOTE: When creating a table on an existing CSV file, you can let CONNECT analyze the file and make the column description. However, CONNECT does not perform an elaborate analysis of the file and, for instance, DATE fields will not be recognized as such, but will be regarded as string fields.

CONNECT CSV Table Type Syntax: (--) (Cont.)

Bad record error processing:

- ▷ "When no match is found for a column field, the process aborts with a message such as: "Bad format line N° field N° of Nombre-Archivo.txt"
- ▷ This can mean that one line of the input file is ill formed or that the column format for this field has been wrongly specified.

⇒ When you know that your file contains rows that are ill formatted and should be eliminated from normal processing, set "maxerr" option in the CREATE TABLE statement:

{ Option-list = 'maxerr = N°'

- This will indicate that no error message will be raised for the first N° wrong lines.
- You can set "maxerr" to a number greater than the number of wrong lines in your files to ignore and get no errors.

▷ "Accept" Option: Permits to keep those ill formatted lines with the bad field, and all succeeding fields of the record, nullified. If "Accept" is specified without "maxerr", all formatted lines will be accepted.

CONNECT Zipped File Tables: "It applies when the table file or files are compressed in one or several zip files".

Specific options when creating tables based on ZIP files:

▷ ZIPPED = $\begin{vmatrix} 1 \\ 0 \end{vmatrix}$: "Required to be set as true".

▷ ENTRY = ' (Text) ': "The optional name or pattern of the zip entry or entries to be used with the table."

{ (*) NOTE: If not specified, all entries or only the first one will be used depending on the "MULENTRY" option setting.

▷ MULENTRY = $\begin{vmatrix} 1 \\ 0 \end{vmatrix}$: "True if several entries are part of the table".

CONNECT Zipped File Tables:

[4]

Specific Options when creating tables based on ZIP files: (...) (Cont.)

▶ MULENTRY = $\begin{matrix} 1 \\ 0 \end{matrix}$: (...) (Cont.)

If not specified, it defaults to:

▶ 0: If the "ENTRY" option is not specified.

▶ If the "ENTRY" option is specified, and it DOES NOT contain wildcard (*) characters.

▶ 1: If the "ENTRY" option is specified, and it DOES contain wildcard (*) characters.

▶ LOAD = ' (Text) ': "Used when creating new zipped tables, to specify the ZIP file from which another ZIP file is created".

CONNECT Zipped File CSV Table Syntax:

▶ Single CSV File included in a Single ZIP File:

```
CREATE TABLE Nombre-Tabla
```

```
( [..] (Column Definition)
```

```
)
```

```
ENGINE = CONNECT
```

```
TABLE_TYPE = CSV
```

```
FILE_NAME = 'C:/Nombre-Carpeta-Rais / Nombre-Subcarpeta / Nombre-Archivo.zip'
```

```
SEP_CHAR = ' ; '
```

(*) Path or name of the ZIP file.

```
HEADER =  $\begin{matrix} 1 \\ 0 \end{matrix}$ 
```

```
ZIPPED =  $\begin{matrix} 1 \\ 0 \end{matrix}$ 
```

```
OPTION_LIST = ' Entry = Nombre-Archivo.csv '
```

(*) Name of the entry inside the ZIP file.

(*) It can be omitted if there is only one entry file inside the file.

[..] (Cont.)

CONNECT Zipped File CSV Table Syntax: (..) (Cont.)

General CSV Files included in a Single ZIP File:

```
CREATE TABLE Nombre-Tabla
(
  [...] (Column Definition)
)
ENGINE = CONNECT
TABLE-TYPE = CSV
FILE = 'C:/Nombre-Carpeta/Nombre-Subcarpeta/Nombre-Archivo.zip'
SEP-CHAR = '|'
HEADER = 1
ZIPPED = 1
OPTION-LIST = 'Entry = Nombre-ArchivoCSV*.csv'
```

(*) Pattern that the files inside the ZIP file must match.

(*) If all entry files are correct, the Entry option can be omitted but the Boolean option `mulentry` must be specified as true: `mulentry = 1`.

Single CSV Files included in multiple ZIP Files:

```
CREATE TABLE Nombre-Tabla
(
  [...] (Column Definition)
)
ENGINE = CONNECT
TABLE-TYPE = CSV
FILE-NAME = 'C:/Nombre-Carpeta/Nombre-Subcarpeta/Nombre-Archivo*.zip'
SEP-CHAR = '|'
HEADER = 1
MULTIPLE = 1 // YES
ZIPPED = 1 // YES
OPTION-LIST = 'Entry = Nombre-ArchivoCSV.csv'
```

(*) The "Entry" option is used to restrict the entry file(s) inside the ZIP. If all are ok it can be omitted.

○ CONNECT Zipped File CSV Table Syntax: (...) (cont.)

○ Characteristics:

- ▷ Column Definitions can be retrieved by the discovery process for table types allowing it, although it cannot be done for multiple tables or multiple entries.
- ▷ A Catalog Table can be created by adding as table option "CATFUNC = COLUMNS". This can be used to show the column definitions of multiple tables, although "MULTIPLE" option must be set to false, and column definitions will be the ones of the first table or entry.

▷ Restrictions:

- 1) Read-Only Implementation. No insert, update or delete.
- 2) The inside files are decompressed into memory. Memory problems may arise with huge files.
- 3) Only file types that can be handled from memory are eligible to this. This includes DOS, FIX, BIN, CSV, FMT, JSON and XML table types.

- ▷ Optimisation by indexing or block indexing is possible for table types supporting it. However, it applies on the uncompressed table. This means that the whole table is always uncompressed.
- ▷ Partitioning also supported.

○ Creating New Zipped Tables: "To make a new ZIP file from an existing file or table"

Ⓐ INSERT Method: "The inserted records are not added, but replace existing ones"

Only three ways are available to do so:

- 1) Using only one INSERT statement to make the new table, which is possible only for small tables:

```
CREATE TABLE Nombre-Tabla
(
  [...] (Column Definitions)
)
ENGINE = CONNECT TABLE-TYPE = CSV ZIPPED = 1
FILE-NAME = 'C:/Nombre-Carpeta / Nombre-Subcarpeta / Nombre-Nuevo-Archivo.zip'
OPTION-LIST = 'Entry = Nombre-Archivo-CSV.csv'
```

□ CONNECT Zipped File Tables: (...) (cont.)

[7]

● Creating new zipped Tables: (...) (cont.)

Ⓐ INSERT Method: (...) (cont.)

1.) Using only one INSERT statement to make the new table, which is possible only for small tables: (...) (cont.)

⋮ [...] (cont.)

```
INSERT INTO Nombre-Tabla
( ( 'N°', (Texto) ), ( '...', ( 'N°', (Texto) ) ),
  [...],
( '...', ( 'N°', (Texto) ) ),
  [...], ( '...', ( 'N°', (Texto) ) ) );
```

2.) Making the table from the data of another table:

2.1) INSERT INTO ... SELECT:

```
CREATE TABLE Nombre-Tabla-ZIP
(
  [...] (Column Definitions)
)
ENGINE = CONNECT
TABLE-TYPE = CSV
ZIPPED = 1
FILE_NAME = 'C: / Nombre-Carpeta / Nombre-Subcarpeta / Nombre-Archivo.zip'
OPTION-LIST = 'Entry = Nombre-ArchivoCSV.csv'

INSERT INTO Nombre-Tabla-ZIP
SELECT
  #
FROM
  Nombre-Tabla-de-la-que-Crear-ZIP;
```

2.2) CREATE TABLE ... AS SELECT:

```
CREATE TABLE Nombre-Tabla-ZIP
ENGINE = CONNECT TABLE-TYPE = CSV ZIPPED = 1
FILE_NAME = 'C: / Nombre-Carpeta / Nombre-Subcarpeta / Nombre-Archivo.zip'
OPTION-LIST = 'Entry = Nombre-ArchivoCSV.csv'
AS
SELECT
  #
FROM
  Nombre-Tabla-de-la-que-Crear-ZIP;
```

Creating New Zipped Tables: (...) (Cont.)

(A) INSERT Method: (...) (Cont.)

3) Making a "zipped file CSV table" from other CSV file which enables a LOAD DATA

INFILE:

```

CREATE TABLE Nombre-Tabla-ZIP
ENGINE = CONNECT TABLE-TYPE = CSV ZIPPED = 1
FILE-NAME = 'C:/Nombre-Carpeta / Nombre-Subcarpeta / Nombre-Archivo.zip'
OPTION-LIST = 'Entry = Nombre-Archivo-CSV.csv';

LOAD DATA ((LOCAL)) INFILE 'Nombre-Fichero|.csv|.txt'
C: | /... (path) ... / Nombre-Fichero|.csv|.txt
D: |
... |

INTO TABLE Nombre-Tabla-ZIP
CHARACTER SET | utf-8 |
| latin1 |
| [-] |

FIELDS TERMINATED BY '| '|
| ; |
| [-] |

ENCLOSED BY '| '|
| " |
| [-] |

ESCAPED BY '\\ '

LINES TERMINATED BY '\\n|n'
IGNORE N° LINES

(( Nombre-Columna1, ..., Nombre-ColumnaN )) ;
    
```

(B) File Zipping Method: "Method that enables to make a ZIP file from another file

when creating the table"

Syntax File Zipping Method for a Single File:

```

CREATE TABLE Nombre-Tabla-ZIP-a-crear
(
    [...] (Column Definitions) | (1) Column descriptions of the CSV (Zipped) file to
    ) | be created must be specified (As the zipped file does not
    ) | exist yet)
ENGINE = CONNECT
TABLE-TYPE = | CSV | (2) Path of the created zipped file
| XML |
| [-] |
FILE-NAME = 'C:/Nombre-Carpeta / Nombre-Subcarpeta / Nombre-Archivo-ZIP-a-Crear.zip'
ZIPPED = 1 | (3) Name of the created CSV file.
OPTION-LIST = 'Entry = Nombre-ArchivoCSV-a-Crear|.csv| [-]'
APPEND = | 1 | YES | [-] load = C:/Nombre-Carpeta / Nombre-Subcarpeta / Nombre-ArchivoCSV-a-crear.csv | [-]
| 0 | NO |
    
```

(4) Path of the file to be loaded.

Creating New Zipped Tables: (...) (Cont.)

(B) File Zipping Method: (...) (Cont.)

Syntax File Zipping Method for a Single File: (...) (Cont.)

(*) NOTE: APPEND = 1 can be used to append another CSV to an existing ZIP file.

Syntax to create multi-entries table from several files: (...) (Cont.)

```
CREATE TABLE Nombre-Tabla
(
  [...] (Column Definitions)
)
ENGINE = CONNECT
TABLE-TYPE = | CSV |
              | [...] |
FILE-NAME = ' C: / Nombre-Carpeta / Nombre-Subcarpeta / Nombre-Archivo-ZIP-a-vec.zip '
ZIPPED = 1
OPTION-LIST = ' Load = C: / Nombre-Carpeta / Nombre-Subcarpeta / Nombre-Archivo* | CSV |
              | [...] |
              multentries = 1 // YES '
```

(*) Necessary to specify multiple entries in the zipped file.

Characteristics:

- ▶ The files to load are specified with wildcard (*) characters.
- ▶ "multentries = 1" option must be specified.
- ▶ "entry" option must not be specified, as entry names will be made from the file names.

ZIP Table Type: "The ZIP table type is meant to display information about the zip file contents".

```
CREATE TABLE Nombre-Tabla-ZIP-Info
(
  File-Name VARCHAR (256) NOT NULL,
  Compressed-File-Size BIGINT NOT NULL FLAG = 1,
  Uncompressed-File-Size BIGINT NOT NULL FLAG = 2,
  Compression-Method INT NOT NULL FLAG = 3
)
ENGINE = CONNECT
TABLE-TYPE = ZIP
FILE-NAME = ' C: / Nombre-Carpeta / Nombre-Subcarpeta / Nombre-Archivo.zip ' ;
```

- (*) Of all entries inside the ZIP file.
- (*) Column names are irrelevant, flag values are the ones that indicate what to retrieve.