

Compétences	Oo5 : Charger la base de données au moyen d'un utilitaire (Bulk charge) ou de commandes SQL avec des données de test.
	<p>Table des matières</p> <p>Chargement des données _____ 1</p> <p>1 Syntaxe de la commande Load Data _____ 1</p> <p>2 Situation _____ 2</p> <p>3 Remarques _____ 2</p> <p>4 Contenu du fichier .CSV _____ 2</p> <p>5 Exemple 1 : CHARACTER SET ... _____ 3</p> <p>6 Exemple 2 : COLUMNS TERMINATED BY _____ 3</p> <p>7 Exemple 3 : ENCLOSED BY _____ 4</p> <p>8 Exemple 4 : ESCAPED BY _____ 4</p> <p>9 Exemple 5 : LINES STARTING BY _____ 5</p> <p>10 Exemple 6 : LINES TERMINATED BY _____ 5</p> <p>11 Exemple 7 : IGNORE 1 LINES _____ 6</p> <p>12 Exemple 8 : Ordre dans les colonnes _____ 6</p> <p>13 Exemple 9 : SET colonne=NULL _____ 7</p> <p>14 Considération avec phpMyAdmin _____ 7</p>
Durée estimée	30 min
Répertoire de travail	.
Fichiers sources	Collection.xlsx
A produire	Charger les contenus des fichiers dans la BD DB-RedWire
Référence	http://dev.mysql.com/doc/refman/5.7/en/load-data.html

Chargement des données

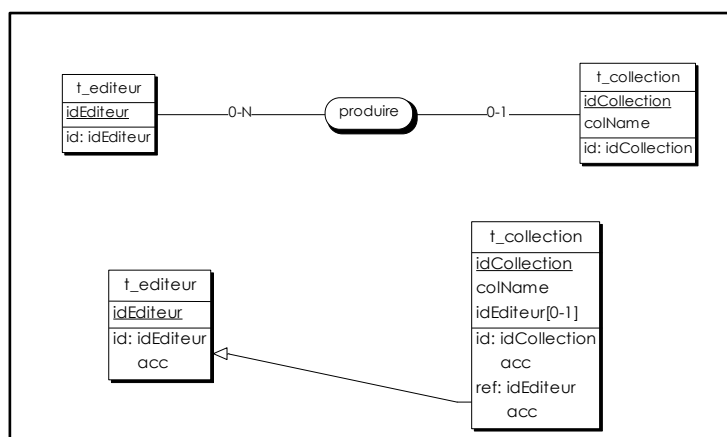
1 Syntaxe de la commande Load Data

```

LOAD DATA [LOW_PRIORITY | CONCURRENT] [LOCAL] INFILE 'file_name'
  [REPLACE | IGNORE]
  INTO TABLE tbl_name
  [PARTITION (partition_name,...)]
  [CHARACTER SET charset_name] -- exemple 1
  [{FIELDS | COLUMNS}
    [TERMINATED BY 'string'] -- exemple 2
    [[OPTIONALLY] ENCLOSED BY 'char'] -- exemple 3
    [ESCAPED BY 'char'] -- exemple 4
  ]
  [LINES
    [STARTING BY 'string'] -- exemple 5
    [TERMINATED BY 'string'] -- exemple 6
  ]
  [IGNORE number {LINES | ROWS}] -- exemple 7
  [(col_name_or_user_var, ...)] -- exemple 8
  [SET col_name = expr, ...] -- exemple 9

```

2 Situation



```

CREATE TABLE IF NOT EXISTS `t_collection` (
  `idCollection` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `colNom` varchar(60) NOT NULL,
  `idEditeur` int(10) unsigned DEFAULT NULL,
  PRIMARY KEY (`idCollection`),
  KEY `idEditeur` (`idEditeur`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=1 ;
  
```

3 Remarques

Dans les exemples qui suivent, il faut considérer les points suivants :

- Le chemin est donné avec des séparateurs « UNIXiens ».
- Les données dans les colonnes sont séparées par le caractère ";", mais d'autres caractères sont utilisables, comme la virgule ",", la tabulation "\t", etc
- L'encodage du fichier texte .CSV est en ANSI, ce qui correspond à l'encodage typique lors de la production d'un fichier CSV avec Excel lorsqu'elle est réalisée en faisant "Enregistrer sous...", puis
CSV (séparateur: point-virgule) (*.csv)
- La clé primaire, idcoll, n'est pas renseignée dans le classeur, car elle est l'objet d'un champ auto-incrémenté, d'où le marqueur NULL
- La clé étrangère, idEditeur, ne peut/doit pas non plus être renseignée, d'où aussi le marqueur NULL.
 Ceci n'est possible que si la colonne a été définie comme ici, `idEditeur` int(10) unsigned **DEFAULT NULL**.

4 Contenu du fichier .CSV

IDcoll	Collection	IDediteur
	La vie quotidienne	
	Le goût de l'Asie	
	Le livre d'or de la SF	
	L'épreuve des faits	
	Les petits libres	
	L'évolution de l'humanité	
	Librio	

5 Exemple 1: CHARACTER SET ...

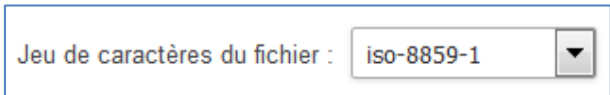
LOAD DATA

```
INFILE 'D://collection.csv'  
INTO TABLE t_collection  
CHARACTER SET "latin1"  
COLUMNS TERMINATED BY ';' ;  
ENCLOSED BY ""  
ESCAPED BY '\t'  
LINES TERMINATED BY '\r\n'  
IGNORE 1 LINES  
SET idCollection=NULL, idEditeur=NULL ;
```

Attention à l'encodage du fichier csv, il doit être le même que celui de la base de données. Dans cet exemple, les colonnes textuelles de la table t_collection qui seraient par exemple encodées en UTF8, ne sont pas comme celles du fichier de données qui est encodé en ANSI. Il faut donc informer le serveur sur la façon dont il doit travailler lors de l'exécution de la commande.

CHARACTER SET est suivi en général de "latin1" ou "utf8"

5.1 Avec PhpMyAdmin, menu Importer :



6 Exemple 2 : COLUMNS TERMINATED BY

```
LOAD DATA INFILE 'D://collection.csv'  
INTO TABLE t_collection  
CHARACTER SET "latin1"  
COLUMNS TERMINATED BY ';' ;  
ENCLOSED BY ""  
ESCAPED BY '\t'  
LINES TERMINATED BY '\r\n'  
IGNORE 1 LINES  
SET idCollection=NULL, idEditeur=NULL ;
```

Les données dans le fichier sont séparées par un caractère spécifique. Les fichiers .CSV (Comma Separated Value) séparent les données d'un enregistrement, habituellement, par une virgule ",".

Toutefois, il apparaît que le caractère virgule soit remplacé par un point-virgule ";", comme dans les exemples de ce document, ou un caractère de tabulation. Dans ce dernier cas, il faut exprimer **COLUMNS TERMINATED BY '\t'** pour renseigner le serveur comment il doit faire son travail lors de l'importation.

6.1 Avec PhpMyAdmin, menu Importer :



Options spécifiques au format :

☐ Remplacer les données de la table avec le fichier

Colonnes séparées par :

;

7 Exemple 3 : ENCLOSED BY

```
LOAD DATA INFILE 'D://collection.csv'  
  INTO TABLE t_collection  
  CHARACTER SET "latin1"  
  COLUMNS TERMINATED BY ';'   
  ENCLOSED BY '' /* le caractère guillemet entre 2 apostrophes */  
  ESCAPED BY '\t'  
  LINES TERMINATED BY '\r\n'  
  IGNORE 1 LINES  
  SET idCollection=NULL, idEditeur=NULL ;
```

Il s'agit d'indiquer au serveur si les champs sont entourés par des caractères comme ici les guillemets (ou autre caractère comme les apostrophes)

Cet entourage apparaît dans certains cas, mais cela est peu fréquent...

7.1 Avec PhpMyAdmin, menu Importer :

Colonnes entourées par :	<input type="text" value=""/>
--------------------------	-------------------------------

8 Exemple 4 : ESCAPED BY

```
LOAD DATA INFILE 'D://collection.csv'  
  INTO TABLE t_collection  
  CHARACTER SET "latin1"  
  COLUMNS TERMINATED BY ';'   
  ENCLOSED BY ''  
  ESCAPED BY '\t'  
  LINES TERMINATED BY '\r\n'  
  IGNORE 1 LINES  
  SET idCollection=NULL, idEditeur=NULL;
```

Parfois, les textes à importer contiennent des caractères de contrôles (par exemple, '\t' pour la tabulation), et il conviendrait de les ignorer. Pour cela, la clause ESCAPED BY '\t' indique au serveur quel caractère il devra ignorer lors de l'importation.

8.1 Avec PhpMyAdmin, menu Importer :

Caractère d'échappement :	<input type="text" value="\t"/>
---------------------------	---------------------------------

9 Exemple 5 : LINES STARTING BY

```
LOAD DATA INFILE 'D://collection.csv'  
  INTO TABLE t_collection  
  CHARACTER SET "latin1"  
  COLUMNS TERMINATED BY ';'   
  ENCLOSED BY ""  
  ESCAPED BY '\t'  
  LINES STARTING BY 'string'  
  LINES TERMINATED BY '\r\n'  
  IGNORE 1 LINES  
  SET idCollection=NULL, idEditeur=NULL;
```

Il y a moyen d'indiquer au serveur quelles lignes il devra importer, dans un fichier de données hétérogènes. Par exemple, si la commande contient LINES STARTING BY 'ok', et que le fichier de données contient :

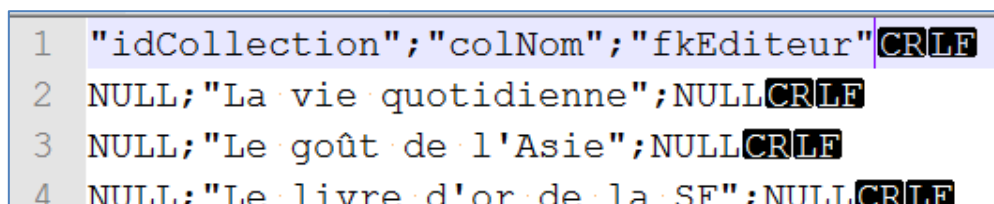
```
"idCollection";"colNom";"idEditeur"  
NULL;"La vie quotidienne";NULL  
ok;"Le goût de l'Asie";NULL  
NULL;"Le livre d'or de la SF";NULL  
ok;"L'épreuve des faits";NULL  
NULL;"Les petits libres";NULL  
NULL;"L'évolution de l'humanité";NULL  
NULL;"Librio";NULL
```

Les lignes 3 et 5 seront importées, mais pas les autres ...

10 Exemple 6 : LINES TERMINATED BY

```
LOAD DATA INFILE 'D://collection.csv'  
  INTO TABLE t_collection  
  CHARACTER SET "latin1"  
  COLUMNS TERMINATED BY ';'   
  ENCLOSED BY ""  
  ESCAPED BY '\t'  
  LINES TERMINATED BY '\r\n'  
  IGNORE 1 LINES  
  SET idCollection=NULL, idEditeur=NULL ;
```

Il est parfois nécessaire d'indiquer au serveur comment les lignes se terminent. Avec Windows, les fins de lignes sont souvent constituées des caractères CR (Carriage Return) et LF (Line Feed) :



```
1 "idCollection";"colNom";"fkEditeur"CRLF  
2 NULL;"La vie quotidienne";NULLCRLF  
3 NULL;"Le goût de l'Asie";NULLCRLF  
4 NULL;"Le livre d'or de la SF";NULLCRLF
```

Avec MySQL, le CR est représenté par '\r' et le LF par '\n'

11 Exemple 7 : IGNORE 1 LINES

```
LOAD DATA INFILE 'D://collection.csv'  
  INTO TABLE t_collection  
  CHARACTER SET "latin1"  
  COLUMNS TERMINATED BY ';'\  
  ENCLOSED BY ""  
  ESCAPED BY '\t'  
  LINES TERMINATED BY '\r\n'  
  IGNORE 1 LINES  
  SET idCollection=NULL, idEditeur=NULL ;
```

Lorsque le fichier de données contient une ligne avec les titres des colonnes, il faut l'ignorer.

Bien évidemment, s'il y a plusieurs lignes non-significatives au début du fichier d'importation, il suffit de les dénombrer en spécifiant le nombre de lignes à ignorer durant l'importation.

11.1 Avec PhpMyAdmin, menu Importer :

Ignorer ce nombre de requêtes (pour SQL) ou de lignes (autres formats), à partir du début :



12 Exemple 8 : Ordre dans les colonnes


```
LOAD DATA INFILE 'D://collection.csv'  
  INTO TABLE t_collection  
  CHARACTER SET "latin1"  
  COLUMNS TERMINATED BY ';'\  
  ENCLOSED BY ""  
  ESCAPED BY '\t'  
  LINES TERMINATED BY '\r\n'  
  IGNORE 1 LINES  
  (idCollection, idEditeur, colNom)  
  SET idCollection=NULL, idEditeur=NULL;
```

Lorsque le fichier de données contient des données pour des colonnes qui ne sont pas dans le même ordre que les colonnes de la table, il est possible de le spécifier au serveur.

Par exemple, la ligne ci-dessus **idCollection, idEditeur, colNom** correspond à un fichier CSV de données qui contiendrait les données suivantes

```
"idCollection";"idEditeur";"colNom"  
;;"La vie quotidienne"  
;;"Le goût de l'Asie"  
;;"Le livre d'or de la SF"  
;;"L'épreuve des faits"  
;;"Les petits livres"  
;;"L'évolution de l'humanité"
```

12.1 Avec PhpMyAdmin, menu Importer :

Nom des colonnes : 

13 Exemple 9 : SET colonne=NULL

```
LOAD DATA INFILE 'D://collection.csv'
  INTO TABLE t_collection
  CHARACTER SET "latin1"
  COLUMNS TERMINATED BY ';'
  ENCLOSED BY '"'
  ESCAPED BY '\\'
  LINES TERMINATED BY '\r\n'
  IGNORE 1 LINES
  (idCollection, idEditeur, colNom)
  SET idCollection=NULL, idEditeur=NULL;
```

Lorsque le fichier de données contient un champ de clé primaire auto-incrémentée, ici idCollection, et/ou une clé étrangère non renseignée, ici idEditeur, il faut informer le serveur de quelle(s) colonne(s) il s'agit, afin que l'auto-incrémentation puisse se faire correctement durant le chargement, ou que le marqueur NULL soit mis à la place de la clé étrangère qui est inconnue.

Remarquons que l'on ne renseigne pas le serveur qu'il doit "mettre" un NULL, mais qu'il doit juste ignorer l'interprétation des données de la colonne à insérer dans la table.

13.1 Avec PhpMyAdmin, menu Importer :

Apparemment, il n'y a aucune option à ce sujet.

14 Considération avec phpMyAdmin

Avec phpMyadmin, l'importation de données provenant d'un fichier .csv peut se faire de la manière suivante :

- Menu Importer
- Fichier (sélection)
- Jeu de car (utf-8 ou ISO-8859-1)
- Ignorer les n premières lignes

A ce stade, il y a 2 options exclusives possibles : CSV ou CSV using LOAD DATA

1. Format CSV

Lorsque l'on choisit CSV, phpMyAdmin choisira lui-même la façon dont il va insérer les enregistrements dans la table.

2. CSV using LOAD DATA

Lorsque l'on choisit CSV using LOAD DATA, phpMyAdmin va fonctionner comme décrit tout au long de ce document.

Toutefois, il a été observé que, jusqu'à la version 4.5.4.1 de phpMyAdmin y compris, cette fonction s'arrête en **erreur** si le fichier csv est encodé en **ANSI**.

Il faut donc artificiellement le convertir en **utf-8**.

A ce stade, aucune explication pertinente n'a été trouvée ... (26 janvier 2018 GGZ)