数据的备份与还原

数据库的管理与维护中，还经常涉及数据的备份与还原。即在重新确定并建立了表的结构基础上，需要将原始大批量数据还原到表结构当中。MySQL 的数据备份与还原一般是采用 load data file 、mysqlimport 、select into outfile 、>/>>重定向等几种方式，这里主要介绍 load data file 和 select into outfile 的方式。MySQL load data 语句能快速将一个文本文件的内容导入到对应的数据库表中(一般文本的一行对应表的一条记录)，load data 语句的效率比一般的 insert 语句要高很多。

1.首先是备份表数据，可以使用 SELECT...INTO OUTFILE 语句将表的内容导出为一个文本文件。其基本的语法格式如下：

```
SELECT [列名] FROM table [WHERE 语句] INTO OUTFILE '目标文件' [OPTION];
```

该语句分为两个部分。前半部分是一个普通的 SELECT 语句，通过这个 SELECT 语句来查询所需要的数据；后半部分是导出数据的。其中，“目标文件”参数指出将查询的记录导出到哪个文件中。

“OPTION”参数为可选参数选项，其可能的取值有：

FIELDS TERMINATED BY '字符串'：设置字符串为字段之间的分隔符，可以为单个或多个字符。默认值是“\t”。

FIELDS ENCLOSED BY '字符'：设置字符来括住字段的值，只能为单个字符。默认情况下不使用任何符号。

FIELDS OPTIONALLY ENCLOSED BY '字符'：设置字符来括住 CHAR、VARCHAR 和 TEXT 等字符型字段。默认情况下不使用任何符号。

FIELDS ESCAPED BY '字符'：设置转义字符，只能为单个字符。默认值为“\”。

LINES STARTING BY '字符串'：设置每行数据开头的字符，可以为单个或多个字符。默认情况下不使用任何字符。

LINES TERMINATED BY '字符串'：设置每行数据结尾的字符，可以为单个或多个字符。默认值是“\n”。

FIELDS 和 LINES 两个子句都是自选的，但是如果两个子句都被指定了，FIELDS 必须位于 LINES 的前面。

如上所述，常用的语法格式可以为：

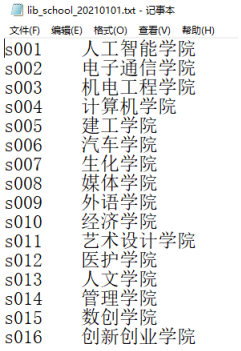
```
SELECT ... FROM TABLE_A
INTO OUTFILE "/path/to/file"
FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"'
LINES TERMINATED BY '\n';
```

【例 9-31】备份 lib 数据库中 school 表的数据到 D 盘文件 lib_school_20210101.txt，数据格式采用系统默认。在终端窗口输入命令：

```
select * from school into outfile  'd:/lib_school_20210101.txt' ;
```

```
mysql> select * from school into outfile 'd:/lib_school_20210101.txt';
Query OK, 16 rows affected (0.02 sec)
```

打开 D 盘文件 lib_school_20210101.txt，可以看到默认数据格式为：

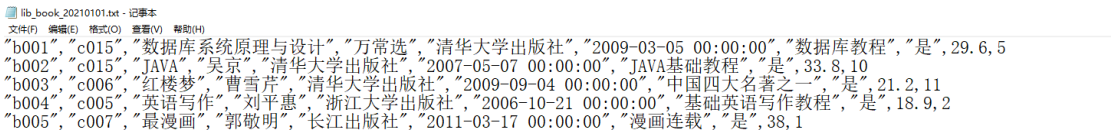


【例 9-32】备份 lib 数据库中 book 表的数据到 D 盘文件 lib_book_20210101.txt，要求字符用”“包含，字段之间用，隔开，每行结束使用\n 换行符。在终端窗口输入命令：

```
select * from book into outfile  'd:/lib_book_20210101.txt'
   FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"'
   LINES TERMINATED BY '\n';
```

```
mysql> select * from book into outfile 'd:/lib_book_20210101.txt'
    ->  FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"'
    -> LINES TERMINATED BY '\n';
Query OK, 205 rows affected (0.00 sec)
```

打开 D 盘文件 lib_book_20210101.txt，可以看到默认数据格式为：



2.其次是还原表数据，即将 select into outfile 导出的文本文件还原到数据库中。

```
LOAD DATA INFILE "/path/to/file" INTO TABLE table_name;
```

LOAD DATA INFILE 语句将从一个文本文件中以很高的速度读入一个表中。当用户一前一后地使用 SELECT ... INTO OUTFILE 和 LOAD DATA INFILE 将数据从一个数据库写到一个文件中，然后再从文件中将它读入数据库中时，两个命令的字段和行处理选项必须匹配。否则，LOAD DATA INFILE 将不能正确地解释文件内容。

【例 9-33】将 D 盘文件 lib_school_20210101.txt 还原到 lib 数据库中 school_bk 表中。首先应创建 school_bk 表结构,然后再还原数据，在终端窗口输入命令：

```
create table school_bk like school;
load data infile  'd:/lib_school_20210101.txt'  into table school_bk;
```

```
mysql> create table school_bk like school;
Query OK, 0 rows affected (0.12 sec)

mysql> load data infile 'd:/lib_school_20210101.txt' into table school_bk;
Query OK, 16 rows affected (0.13 sec)
Records: 16  Deleted: 0  Skipped: 0  Warnings: 0
```

【例 9-34】将 D 盘文件 lib_book_20210101.txt 还原到 lib 数据库中 book_bk 表中，注意数据字符用”“标注，字段之间用，隔开，每行结束使用\n 换行符。在终端窗口输入命令：

```
create table book_bk like book;
load data infile 'd:/lib_book_20210101.txt' into table book_bk
```

```
    FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"'
    LINES TERMINATED BY '\n';
```

```
mysql> create table book_bk like book;
Query OK, 0 rows affected (0.14 sec)

mysql> load data infile 'd:/lib_book_20210101.txt' into table book_bk
    -> FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"'
    -> LINES TERMINATED BY '\n';
Query OK, 205 rows affected (0.06 sec)
Records: 205  Deleted: 0  Skipped: 0  Warnings: 0
```