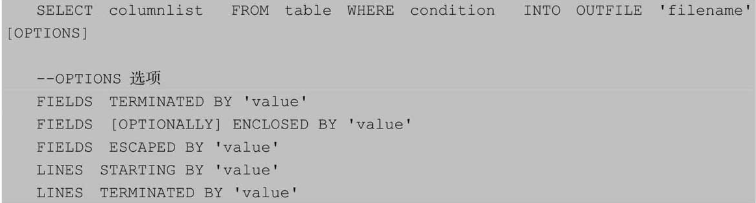
# 概述

有时会需要将MySQL数据库中的数据导出到外部存储文件中，MySQL数据库中的数据可以导出成sql文本文件、xml文件或者html文件。同样，这些导出文件也可以导入到MySQL数据库中。

# 导入

## SELECT INTO OUTFILE

MySQL数据库导出数据时，允许使用包含导出定义的SELECT语句进行数据的导出操作。该文件被创建到服务器主机上，因此必须拥有文件写入权限（FILE权限）才能使用此语法。“SELECT...INTOOUTFILE 'filename'”形式的SELECT语句可以把被选择的行写入一个文件中，并且filename不能是一个已经存在的文件。SELECT...INTO OUTFILE语句的基本格式如下：



可以看到SELECT columnlist FROM table WHERE condition为一个查询语句，查询结果返回满足指定条件的一条或多条记录；INTO OUTFILE语句的作用就是把前面SELECT语句查询出来的结果导出到名称为“filename”的外部文件中。[OPTIONS]为可选参数选项，OPTIONS部分的语法包括FIELDS和LINES子句，其可能的取值有：

●　FIELDS　TERMINATED BY 'value'：设置字段之间的分隔字符，可以为单个或多个字符，默认情况下为制表符‘\t’。

●　FIELDS　[OPTIONALLY] ENCLOSED BY 'value'：设置字段的包围字符，只能为单个字符，若使用了OPTIONALLY则只有CHAR和VERCHAR等字符数据字段被包括。

●　FIELDS　ESCAPED BY 'value'：设置如何写入或读取特殊字符，只能为单个字符，即设置转义字符，默认值为‘\’。

●　LINES　STARTING BY 'value'：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 'value'：设置每行数据结尾的字符，可以为单个或多个字符，默认值为‘\n’。

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

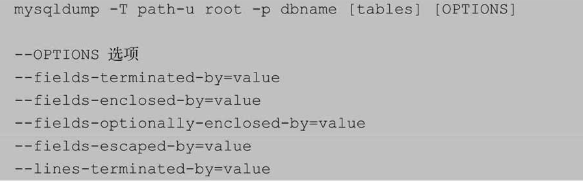
SELECT...INTO OUTFILE语句可以非常快速地把一个表转储到服务器上。如果想要在服务器主机之外的部分客户主机上创建结果文件，不能使用SELECT...INTO OUTFILE。在这种情况下，应该在客户主机上使用比如“MySQL –e "SELECT ..." > file\_name”的命令来生成文件。

SELECT...INTO OUTFILE是LOAD DATA INFILE的补语。用于语句的OPTIONS部分的语法包括部分FIELDS和LINES子句，这些子句与LOAD DATA INFILE语句同时使用。

## Mysqldump

除了使用SELECT… INTO OUTFILE语句导出文本文件之外，还可以使用MySQLdump。本章开始介绍了使用MySQLdump备份数据库，该工具不仅可以将数据导出为包含CREATE、INSERT的sql文件，也可以导出为纯文本文件。

MySQLdump创建一个包含创建表的CREATE TABLE语句的tablename.sql文件和一个包含其数据的tablename.txt文件。MySQLdump导出文本文件的基本语法格式如下：



只有指定了-T参数才可以导出纯文本文件；path表示导出数据的目录；tables为指定要导出的表名称，如果不指定，将导出数据库dbname中所有的表；[OPTIONS]为可选参数选项，这些选项需要结合-T选项使用。使用OPTIONS常见的取值有：

●　--fields-terminated-by=value：设置字段之间的分隔字符，可以为单个或多个字符，默认情况下为制表符“\t”。

●　--fields-enclosed-by=value：设置字段的包围字符。

●　--fields-optionally-enclosed-by=value：设置字段的包围字符，只能为单个字符，只能包括CHAR和VERCHAR等字符数据字段。

●　--fields-escaped-by=value：控制如何写入或读取特殊字符，只能为单个字符，即设置转义字符，默认值为反斜线“\”。

●　--lines-terminated-by=value：设置每行数据结尾的字符，可以为单个或多个字符，默认值为“\n”。

**提示：**

与SELECT…INTO OUTFILE语句中的OPTIONS各个参数设置不同，这里OPTIONS各个选项等号后面的value值不要用引号括起来。

## Mysql

MySQL是一个功能丰富的工具命令，使用MySQL还可以在命令行模式下执行SQL指令，将查询结果导入到文本文件中。相比MySQLdump，MySQL工具导出的结果可读性更强。

如果MySQL服务器是单独的机器，用户是在一个client上进行操作，用户要把数据结果导入到client机器上。可以使用MySQL -e语句。

使用MySQL导出数据文本文件语句的基本格式如下：



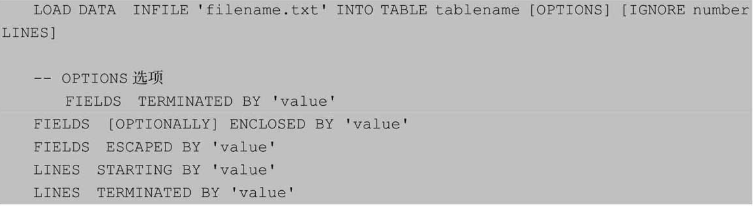
该命令使用--execute选项，表示执行该选项后面的语句并退出，后面的语句必须用双引号括起来，dbname为要导出的数据库名称；导出的文件中不同列之间使用制表符分隔，第1行包含了各个字段的名称。

# 导出

## LOAD DATA FROM

MySQL允许将数据导出到外部文件，也可以从外部文件导入数据。MySQL提供了一些导入数据的工具，包括LOAD DATA语句、source命令和mysql命令。LOAD DATA INFILE语句用于高速地从一个文本文件中读取行，并装入一个表中。文件名称必须为文字字符串。

LOAD DATA语句的基本格式如下：



可以看到LOAD DATA语句中，关键字INFILE后面的filename文件为导入数据的来源；tablename表示待导入的数据表名称；[OPTIONS]为可选参数选项，OPTIONS部分的语法包括FIELDS和LINES子句，其可能的取值有：

●　FIELDS　TERMINATED BY 'value'：设置字段之间的分隔字符，可以为单个或多个字符，默认情况下为制表符“\t”。

●　FIELDS　[OPTIONALLY] ENCLOSED BY 'value'：设置字段的包围字符，只能为单个字符。如果使用了OPTIONALLY，则只有CHAR和VERCHAR等字符数据字段被包括。

●　FIELDS　ESCAPED BY 'value'：控制如何写入或读取特殊字符，只能为单个字符，即设置转义字符，默认值为“\”。

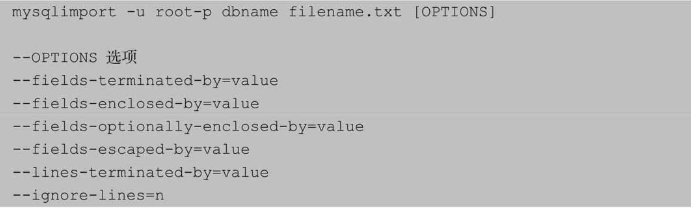
●　LINES　STARTING BY 'value'：设置每行数据开头的字符，可以为单个或多个字符，默认情况下不使用任何字符。

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

IGNORE number LINES选项表示忽略文件开始处的行数，number表示忽略的行数。执行LOADDATA语句需要FILE权限。

## mysqlimport

使用MySQLimport可以导入文本文件，并且不需要登录MySQL客户端。MySQLimport命令提供许多与LOAD DATA INFILE语句相同的功能，大多数选项直接对应LOAD DATA INFILE子句。使用MySQLimport语句需要指定所需的选项、导入的数据库名称以及导入的数据文件的路径和名称。MySQLimport命令的基本语法格式如下：



dbname为导入的表所在的数据库名称。注意，MySQLimport命令不指定导入数据库的表名称，数据表的名称由导入文件名称确定，即文件名作为表名，导入数据之前该表必须存在。[OPTIONS]为可选参数选项，其常见的取值有：

●　--fields-terminated-by= 'value'：设置字段之间的分隔字符，可以为单个或多个字符，默认情况下为制表符“\t”。

●　--fields-enclosed-by= 'value'：设置字段的包围字符。

●　--fields-optionally-enclosed-by= 'value'：设置字段的包围字符，只能为单个字符，包括CHAR和VERCHAR等字符数据字段。

●　--fields-escaped-by= 'value'：控制如何写入或读取特殊字符，只能为单个字符，即设置转义字符，默认值为反斜线“\”。

●　--lines-terminated-by= 'value'：设置每行数据结尾的字符，可以为单个或多个字符，默认值为“\n”。

●　--ignore-lines=n：忽视数据文件的前n行。

# Oracle导入导出

有时会需要将Oracle数据库中的数据导出到外部存储文件中，Oracle数据库中的数据表可以导出，同样这些导出文件也可以导入到Oracle数据库中。

## 导出

### 用EXP工具导出数据

在DOS窗口下，输入以下语句，然后根据提示即可导出表。



其中username是登录数据库的用户名，password为用户密码。注意这里的用户不能为SYS。

【例17.6】导出数据表books，代码如下：



这里指出了导出文件的名称和路径，然后指出导出表的名称。如果要导出多个表，可以在各个表之间用逗号隔开。

导出表空间和导出表不同，导出表空间的用户必须是数据库的管理员角色。导出表空间的命令如下：



其中参数username/password表示具有数据库管理员权限的用户名和密码，filename.dmp表示存放备份的表空间的数据文件，tablespaces\_name表示要备份的表空间名称。

【例17.7】导出表空间MYTEM，代码如下：



### 用EXPDP导出数据

EXPDP是从ORCALE 10g开始提供的导入导出工具，采用的是数据泵技术，该技术是在数据库之间或者数据库与操作系统之间传输数据的工具。

数据泵技术的主要特性如下：

支持并行处理导入、导出任务。

支持暂停和重启导入、导出任务。

支持通过联机的方式导出或导入远端数据库中的对象。

支持在导入时实现导入过程中自动修改对象属主、数据文件或数据所在表空间。

导入／导出时提供了非常细粒度的对象控制，甚至可以详细制定是否包含或不包含某个对象。

下面开始讲述使用EXPDP导出数据的过程。

1. 创建目录对象

使用EXPDP工具之前，必须创建目录对象，具体的语法规则如下：



其中参数directory\_name为创建目录的名称，file\_name表示存放数据的文件夹名。

【例17.8】创建目录对象MYDIR，代码如下：



结果如下：



1. 给使用目录的用户赋权限

新创建的目录对象不是所有用户都可以使用，只有拥有该目录权限的用户才可以使用。假设备份数据库的用户是SCOTT，那么赋予权限的具体语法如下：



其中参数directory\_name表示目录的名称。

【例17.9】将目录对象MYDIR权限赋予SCOTT，代码如下：

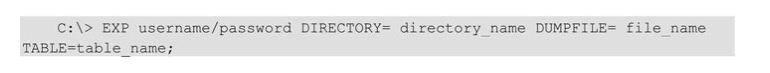


运行结果如下：



1. 导出指定的表

创建完目录后，即可使用EXPDP工具导出数据，操作也是在DOS的命令窗口中完成。指定备份表的语法格式如下：



其中参数directory\_name表示存放导出数据的目录名称。file\_name表示导出数据存放的文件名。table\_name表示准备导出的表名，如果导出多个表，可以用逗号隔开。

【例17.10】导出数据表BOOKS，代码如下：



## 导入

### 用IMP导入数据

导入数据是导出数据的逆过程，使用EMP导出的数据，可以使用IMP导入数据。

【例17.11】使用EXP导出fruits表，命令如下：



【例17.12】使用IMP导入fruits表，命令如下：



### 用IMPDP导入数据

使用EXPDP导出数据后，可以使用IMPDP将数据导入。

【例17.13】使用IMPDP导入BOOKS表，命令如下：



如果数据库中BOOKS表已经存在，此时会报错，解决方式是在上面代码后加上ignore=y即可。