1.mysql -u root -p 数据库名 </文件路径(在bin下就直接加文件名)   
2.mysqlimport的语法介绍：   
mysqlimport位于mysql/bin目录中，是mysql的一个载入（或者说导入）数据的一个非常有效的工具。这是一个命令行工具。有两个参数以及大量的选项可供选择。这个工具把一个文本文件（text file）导入到你指定的数据库和表中。比方说我们要从文件Customers.txt中把数据导入到数据库Meet\_A\_Geek中的表 Custermers中：   
mysqlimport Meet\_A\_Geek Customers.txt   
注意：这里Customers.txt是我们要导入数据的文本文件，而Meet\_A\_Geek是我们要操作的数据库，数据库中的表名是Customers，这里文本文件的数据格式必须与Customers表中的记录格式一致，否则mysqlimport命令将会出错。   
其中表的名字是导入文件的第一个句号（.）前面文件字符串，另外一个例子：   
mysqlimport Meet\_A\_Geek Cus.to.mers.txt   
那么我们将把文件中的内容导入到数据库Meet\_A\_Geek 中的Cus表中。上面的例子中，都只用到两个参数，并没有用到更多的选项，下面介绍mysqlimport的选项:   
选项 功能   
-d or --delete 新数据导入数据表中之前删除数据数据表中的所有信息   
-f or --force 不管是否遇到错误，mysqlimport将强制继续插入数据   
-i or --ignore mysqlimport跳过或者忽略那些有相同唯一关键字的行， 导入文件中的数据将被忽略。   
-l or -lock-tables 数据被插入之前锁住表，这样就防止了，你在更新数据库时，用户的查询和更新受到影响。   
-r or -replace 这个选项与－i选项的作用相反；此选项将替代表中有相同唯一关键字的记录。   
--fields-enclosed- by= char 指定文本文件中数据的记录时以什么括起的， 很多情况下数据以双引号括起。 默认的情况下数据是没有被字符括起的。   
--fields-terminated- by=char 指定各个数据的值之间的分隔符，在句号分隔的文件中，分隔符是句号。您可以用此选项指定数据之间的分隔符。默认的分隔符是跳格符（Tab）   
--lines-terminated- by=str 此选项指定文本文件中行与行之间数据的分隔字符串或者字符。 默认的情况下mysqlimport以newline为行分隔符。您可以选择用一个字符串来替代一个单个的字符：一个新行或者一个回车。   
mysqlimport命令常用的选项还有-v 显示版本（version）， -p 提示输入密码(password）等。   
例子：导入一个以逗号为分隔符的文件文件中行的记录格式是这样的：   
"1", "ORD89876", "1 Dozen Roses", "19991226"   
我们的任务是要把这个文件里面的数据导入到数据库Meet\_A\_Geek中的表格Orders中，我们使用这个命令:   
bin/mysqlimport –prl –fields-enclosed-by=" –fields-terminated-by=, Meet\_A\_Geek Orders.txt   
3.mysql数据库常用导出导入命令的几个用例   
＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝   
几个常用用例：   
1.导出整个数据库   
mysqldump -u 用户名 -p 数据库名 > 导出的文件名   
mysqldump -u wcnc -p smgp\_apps\_wcnc > wcnc.sql   
2.导出一个表   
mysqldump -u 用户名 -p 数据库名 表名> 导出的文件名   
mysqldump -u wcnc -p smgp\_apps\_wcnc users> wcnc\_users.sql   
3.导出一个数据库结构   
mysqldump -u wcnc -p -d --add-drop-table smgp\_apps\_wcnc >d:\wcnc\_db.sql   
-d 没有数据 --add-drop-table 在每个create语句之前增加一个drop table   
4.导入数据库   
常用source 命令   
进入mysql数据库控制台，   
如mysql -u root -p   
mysql>use 数据库   
然后使用source命令，后面参数为脚本文件（如这里用到的.sql）   
mysql>source d:\wcnc\_db.sql   
4.可参看 MySQL数据的导出和导入工具:mysqldump   
＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝   
批处理是一种非交互式运行mysql程序的方法，如同您在mysql中使用的命令一样，你仍然将使用这些命令。为了实现批处理，您重定向一个文件到 mysql程序中，首先我们需要一个文本文件，这个文本文件包含有与我们在mysql中输入的命令相同的文本。比如我们要插入一些数据，使用包含下面文本的文件(文件名为New\_Data.sql,当然我们也可以取名为New\_Data.txt及任何其他的合法名字，并不一定要以后缀sql结尾)：   
USE Meet\_A\_Geek;   
INSERT INTO Customers (Customer\_ID, Last\_Name) VALUES(NULL, "Block");   
INSERT INTO Customers (Customer\_ID, Last\_Name) VALUES(NULL, "Newton");   
INSERT INTO Customers (Customer\_ID, Last\_Name) VALUES(NULL, "Simmons");   
注意上面的这些句子的语法都必须是正确的，并且每个句子以分号结束。上面的USE命令选择数据库，INSERT命令插入数据。   
下面我们要把上面的文件导入到数据库中，导入之前要确认数据库已经在运行，即是mysqld进程（或者说服务，Windows NT下面称为”服务“，unix下面为”进程“）已经在运行。然后运行下面的命令：   
bin/mysql –p < /home/mark/New\_Data.sql   
接着按提示输入密码，如果上面的文件中的语句没有错误，那么这些数据就被导入到了数据库中。   
命令行中使用LOAD DATA INFILE 从文件中导入数据到数据库:   
现在您可能会问自己，"究竟为什么我要输入所有的这些SQL语句到文件中，然后通过程序运行它们呢？”这样看起来好像需要大量的工作。很好，你这样想很可能就对了。但是假如你有从所有这些命令中产生的log记录呢？现在这样就很棒，嗯，大多数数据库都会自动产生数据库中的事件记录的log。而大部分log 都包含有用过的原始的SQL命令。因此，如果您不能从您现在的数据库中导出数据到新的mysql数据库中使用，那么您可以使用log和mysql的批处理特性，来快速且方便地导入您地数据。当然，这样就省去了打字的麻烦。   
LOAD DATA INFILE   
这是我们要介绍的最后一个导入数据到MySQL数据库中的方法。这个命令与mysqlimport非常相似，但这个方法可以在mysql命令行中使用。也就是说您可以在所有使用API的程序中使用这个命令。使用这种方法，您就可以在应用程序中导入您想要导入的数据。   
使用这个命令之前，mysqld进程（服务）必须已经在运行。启动mysql命令行：   
bin/mysql –p   
按提示输入密码，成功进入mysql命令行之后，输入下面的命令：   
USE Meet\_A\_Geek;   
LOAD DATA INFILE "/home/mark/data.sql" INTO TABLE Orders;   
简单的讲，这样将会把文件data.sql中的内容导入到表Orders中，如mysqlimport工具一样，这个命令也有一些可以选择的参数。比如您需要把自己的电脑上的数据导入到远程的数据库服务器中，您可以使用下面的命令：   
LOAD DATA LOCAL INFILE "C:\MyDocs\SQL.txt" INTO TABLE Orders;   
上面的LOCAL参数表示文件是本地的文件，服务器是您所登陆的服务器。这样就省去了使用ftp来上传文件到服务器，MySQL替你完成了.   
您也可以设置插入语句的优先级，如果您要把它标记为低优先级（LOW\_PRIORITY），那么MySQL将会等到没有其他人读这个表的时候，才把插入数据。可以使用如下的命令：   
LOAD DATA LOW\_PRIORITY INFILE "/home/mark/data.sql" INTO TABLE Orders;   
您也可以指定是否在插入数据的时候，取代或者忽略文件与数据表中重复的键值。替代重复的键值的语法：   
LOAD DATA LOW\_PRIORITY INFILE "/home/mark/data.sql" REPLACE INTO TABLE Orders;   
上面的句子看起来有点笨拙，但却把关键字放在了让您的剖析器可以理解的地方。   
下面的一对选项描述了文件的记录格式，这些选项也是在mysqlimport工具中可以用的。他们在这里看起来有点不同。首先，要用到FIELDS关键字，如果用到这个关键字，MySQL剖析器希望看到至少有下面的一个选项：   
TERMINATED BY character   
ENCLOSED BY character   
ESCAPED BY character   
这些关键字与它们的参数跟mysqlimport中的用法是一样的. The   
TERMINATED BY 描述字段的分隔符，默认情况下是tab字符（\t）   
ENCLOSED BY描述的是字段的括起字符。比方以引号括起每一个字段。   
ESCAPED BY 描述的转义字符。默认的是反些杠（backslash：\ ）.   
下面仍然使用前面的mysqlimport命令的例子，用LOAD DATA INFILE语句把同样的文件导入到数据库中:   
LOAD DATA INFILE "/home/mark/Orders.txt" REPLACE INTO TABLE Orders FIELDS TERMINATED BY ','   
ENCLOSED BY '"';   
LOAD DATA INFILE语句中有一个mysqlimport工具中没有特点：LOAD DATA INFILE 可以按指定的列把文件导入到数据库中。   
当我们要把数据的一部分内容导入的时候，这个特点就很重要。比方说，我们要从Access数据库升级到MySQL数据库的时候，需要加入一些栏目（列/字段 /field）到MySQL数据库中，以适应一些额外的需要。这个时候，我们的Access数据库中的数据仍然是可用的，但是因为这些数据的栏目 (field)与MySQL中的不再匹配，因此而无法再使用mysqlimport工具。尽管如此，我们仍然可以使用LOAD DATA INFILE，下面的例子显示了如何向指定的栏目(field)中导入数据：   
LOAD DATA INFILE "/home/Order.txt" INTO TABLE Orders(Order\_Number, Order\_Date, Customer\_ID);   
如您所见，我们可以指定需要的栏目（fields）。这些指定的字段依然是以括号括起，由逗号分隔的，如果您遗漏了其中任何一个，MySQL将会提醒您^\_^ 。