



链滴

# 学习笔记 --MySQL

作者: [KioLuo](#)

原文链接: <https://ld246.com/article/1492568687994>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

<p><strong>//安装 mysql5.7 解压版</strong></p>  
<p>1. 解压</p>  
<p>2. 设置环境变量</p>  
<p>PATH 中加入 mysql 的 bin 路径</p>  
<p>3. 修改 my-default.ini</p>  
<p>[mysqld]</p>  
<p>basedir = D:\MySQL\Server\mysql-5.6.20-win32<br>datadir = D:\MySQL\Server\data<br>port = 3306</p>  
<p>4. 安装 mysql 服务</p>  
<p>在 bin 下执行: </p>  
<p>D:\MySQL\Server\mysql-5.6.20-win32\bin>mysqld -install<br>Service successfully installed.</p>  
<p><strong>//MySQL 管理</strong></p>  
<p><strong>//启动及关闭 MySQL</strong></p>  
<p>启动 mysql 服务</p>  
<p>若在 myql 目录中没有 data 文件夹, 则会启动失败, 需要在 bin 中执行: </p>  
<p>mysqld --initialize-insecure --user=mysql</p>  
<p>会自动创建 data 文件夹, 现在可以启动 mysql 服务: </p>  
<p>net start mysql</p>  
<p>登录 mysql: </p>  
<p>mysql -u root -p</p>  
<p>默认密码为空</p>  
<p>添加密码: </p>  
<p>mysqladmin -u root password "new\_password" </p>  
<p>关闭 mysql 服务</p>  
<p>mysqladmin -u root -p shutdown</p>  
<p><strong>//用户设置</strong></p>  
<p>创建用户</p>  
<p>CREATE USER 'username'@'host' IDENTIFIED BY 'password';</p>  
<p>说明:username - 你将创建的用户名, host - 指定该用户在哪个主机上可以登陆,如果是本地用户用 localhost, 如果想让该用户可以从任意远程主机登陆,可以使用通配符 %. password - 该用户的密码,密码可以为空,如果为空则该用户不需要密码登陆服务器.</p>  
<p>注意: % 为任意远程主机, localhost 和 127.0.0.1 权限不一样</p>  
<p>例子: CREATE USER 'dog'@'localhost' IDENTIFIED BY '123456';</p>  
<p>CREATE USER 'pig'@'192.168.1.101\_' IDENTIFIED BY '123456';</p>  
<p>CREATE USER 'pig'@'%' IDENTIFIED BY '123456';</p>  
<p>CREATE USER 'pig'@'%' IDENTIFIED BY '';</p>  
<p>CREATE USER 'pig'@'%';</p>  
<p>授权</p>  
<p>GRANT privileges ON databasename.tablename TO 'username'@'host'</p>  
<p>说明: privileges - 用户的操作权限,如 SELECT , INSERT , UPDATE 等(详细列表见该文最后面). 果要授予所的权限则使用 ALL.;databasename - 数据库名,tablename-表名,如果要授予该用户对所数据库和表的相应操作权限则可用<em>表示, 如</em>.\*.</p>  
<p>注意:用以上命令授权的用户不能给其它用户授权,如果想让该用户可以授权,用以下命令:</p>  
<p>GRANT privileges ON databasename.tablename TO 'username'@'host' WITH GRANT OPTION;</p>  
<p>设置与更改用户密码</p>  
<p>SET PASSWORD FOR 'username'@'host' = PASSWORD('newpassword');如果是当前登陆用 SET PASSWORD = PASSWORD("newpassword");</p>  
<p>例子: SET PASSWORD FOR 'pig'@'%' = PASSWORD("123456");</p>  
<p>撤销用户权限</p>  
<p>REVOKE privilege ON databasename.tablename FROM 'username'@'host';</p>  
<p>注意: 假如你在给用户'pig'@'%'授权的时候是这样的(或类似的):GRANT SELECT ON test.user T

'pig'@%', 则在使用 REVOKE SELECT ON <em>.</em> FROM 'pig'@'%';命令并不能撤销该用户 test 数据库中 user 表的 SELECT 操作.相反,如果授权使用的是 GRANT SELECT ON <em>.</em> O 'pig'@'%';则 REVOKE SELECT ON test.user FROM 'pig'@'%';命令也不能撤销该用户对 test 数据库中 user 表的 Select 权限.</p>

<p>删除用户: </p>

<p>DROP USER 'username'@'host';</p>

<p>重新载入授权表: </p>

<p>FLUSH PRIVILEGES;</p>

<p>在 mysql 数据库中的 user 表添加新用户: </p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">INSERT INTO user
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">          (host, user,
</span></span><span class="highlight-line"><span class="highlight-cl">          assword,
</span></span><span class="highlight-line"><span class="highlight-cl">          select_priv,
</span></span><span class="highlight-line"><span class="highlight-cl">          nsert_priv, update_priv)
</span></span><span class="highlight-line"><span class="highlight-cl">          VALUES ('l
</span></span><span class="highlight-line"><span class="highlight-cl">          calhost', 'guest',
</span></span><span class="highlight-line"><span class="highlight-cl">          PASSWOR
</span></span><span class="highlight-line"><span class="highlight-cl">          ('guest123'), 'Y', 'Y', 'Y');
</span></span></code></pre>
```

<p>注意: 在 MySQL5.7 中 user 表的 password 已换成了 authentication\_string</p>

<p>通过 GRANT 命令添加用户并添加权限: </p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">mysql> GRANT SELECT,INSERT,UPDATE,DELETE,CREATE,DROP
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">  -&gt; ON TUT
</span></span><span class="highlight-line"><span class="highlight-cl">  RIALS.*
</span></span><span class="highlight-line"><span class="highlight-cl">  -&gt; TO 'zara'
</span></span><span class="highlight-line"><span class="highlight-cl">  'localhost'
</span></span><span class="highlight-line"><span class="highlight-cl">  -&gt; IDENTIFI
</span></span><span class="highlight-line"><span class="highlight-cl">  D BY 'zara123';
</span></span></code></pre>
```

<p><strong>示例: </strong></p>

```
<pre><code class="language-mysql highlight-chroma"><span class="highlight-line"><span class="highlight-cl"><span class="highlight-n">mysql</span><span class="highlight-w"> <
</span></span><span class="highlight-line"><span class="highlight-cl"><span class="highlight-o">&gt;</span><span class="highlight-w"> </span><span class="highlight-cl"><span class="highlight-k">create</span><span class="highlight-w"> </span><span class="highlight-cl"><span class="highlight-k">database</span><span class="highlight-w"> </span><span class="highlight-cl"><span class="highlight-n">examp
</span></span><span class="highlight-line"><span class="highlight-cl"><span class="highlight-w"> </span><span class="highlight-cl"><span class="highlight-k">default</span><span class="highlight-w"> </span><span class="highlight-cl"><span class="highlight-k">character</span><span class="highlight-w"> </span><span class="highlight-cl"><span class="highlight-kt">set</span><span class="highlight-w"> </span><span class="highlight-cl"><span class="highlight-n">utf8</span><span class="highlight-p">;</span><span class="highlight-w">
</span></span></code></pre>
```





```

pan><span class="highlight-n">userName</span><span class="highlight-o">`</span><sp
n class="highlight-p">,</span><span class="highlight-w"> </span><span class="highlight-
">`</span><span class="highlight-n">userPassword</span><span class="highlight-o">`</
pan><span class="highlight-p">,</span><span class="highlight-w"> </span><span class=
highlight-o">`</span><span class="highlight-n">userDesc</span><span class="highlight-o
">`</span><span class="highlight-p">)</span><span class="highlight-w"> </span><span c
ass="highlight-k">VALUES</span><span class="highlight-w">
</span></span></span><span class="highlight-line"><span class="highlight-cl"><span cla
s="highlight-w">
</span></span></span><span class="highlight-line"><span class="highlight-cl"><span cla
s="highlight-w"> </span><span class="highlight-p">(</span><span class="highlight-s1
">'test_user'</span><span class="highlight-p">,</span><span class="highlight-w"> </span>
<span class="highlight-s1">'test_password'</span><span class="highlight-p">,</span><sp
n class="highlight-w"> </span><span class="highlight-s1">'Test user for server example'</
pan><span class="highlight-p">);</span><span class="highlight-w">
</span></span></span></code></pre>
<p><strong>//管理 MySQL 的常用命令</strong></p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">mysqladmin -u root -p create ; //创建数据库
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">mysqladmin -u ro
t -p drop ; //删除数据库
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">CREATE database
xample default character set utf8; //创建数据库
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">USE &lt;数据库名
gt;; //选择要操作的数据库，以后的MySQL命令都只针对该数据库
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">SHOW DATABASE
; //列出MySQL数据库管理系统所管理的数据库列表
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">SHOW TABLES;
//列出指定数据库中的所有表，需要先使用use 数据库
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">SHOW INDEX FR
M &lt;数据表&gt;; //显示数据表的详细索引信息，包括PRIMARY KEY（主键）
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">SHOW COLUMNS
FROM &lt;数据表&gt;; //显示数据表的详细列头信息
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">SHOW TABLE STA
US FROM &lt;数据库&gt; LIKE '需要匹配的字符串'\G; //该命令将输出Mysql数据库管理系统
性能及统计信息
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">CREATE TABLE tab
e_name (column_name column_type); //创建数据表
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">SHOW PROCESSL
ST; //显示数据库连接线程列表
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">SHOW CREATE T
BLE table_name; //显示表信息
</span></span><span class="highlight-line"><span class="highlight-cl">

```

```

</span></span><span class="highlight-line"><span class="highlight-cl">SHOW VARIABLES
LIKE '%character%';      //显示字符集编码信息
</span></span></code></pre>
<p>实例: </p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">mysql> CREATE TABLE runoob_tbl(-&gt; runoob_id INT NOT NULL AUTO_INCREMENT,
&gt; runoob_title VARCHAR(100) NOT NULL,-&gt; runoob_author VARCHAR(40) NOT NULL,-
&gt; submission_date DATE,-&gt; PRIMARY KEY ( runoob_id )-&gt; ) ENGINE=InnoDB DEFAULT
CHARSET = utf8;Query OK, 0 rows affected (0.16 sec)
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">DROP TABLE table
name;      //删除数据表
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">INSERT INTO tabl
_name (field1, field2, ... fieldN) VALUES (value1, value2, ...valueN);      //插入数据
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">SELECT column_n
me, column_name FROM table_name [WHERE Clause] [OFFSET M] [LIMIT N]; //查询数据
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">UPDATE table_na
e SET field1=new-value1, field2=new-value2 [WHERE Clause];      //更新数据
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">DELETE FROM tab
e_name [WHERE Clause];      //删除数据
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">SELECT field1, fiel
2, ... fieldN table_name1, table_name2,... table_nameN WHERE field1 LIKE condition1;      /
where从句中like和%的使用
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">SELECT field1, fiel
2, ...fieldN table_name1, table_name2, ...table_nameN ORDER BY field1, [field2...] ASC [DESC]
</span></span></code></pre>
<p>//排序</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">set names utf8;      //设置为utf8编码
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">SET sql_mode='N
_AUTO_VALUE_ON_ZERO';      //设置AUTO_INCREMENT的值从0开始
</span></span></code></pre>
<p>//GROUP BY 语法, 在分组的列上我们可以使用 COUNT, SUM, AVG,等函数。 </p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">SELECT column_name, function(column_name) FROM table_name
</span></span><span class="highlight-line"><span class="highlight-cl">WHERE column_n
me operator value
</span></span><span class="highlight-line"><span class="highlight-cl">GROUP BY colum
_name;
</span></span></code></pre>
<p>实例: </p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">mysql> SELECT name, COUNT(*) FROM employee_tbl GROUP BY name;
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">mysql> SELECT
name, SUM(singin) as singin_count FROM employee_tbl GROUP BY name WITH ROLLUP;
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span></code></pre>

```

```
</span></span><span class="highlight-line"><span class="highlight-cl">mysql> SELECT coalesce(name, '总数'), SUM(singin) as singin_count FROM employee_tbl GROUP BY name WITH ROLLUP;
```

```
</span></span></code></pre>
```

<p><strong>//MySQL 连接的使用</strong></p>

<p>//使用 INNER JOIN</p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">例: SELECT a.runoob_id, a.runoob_author, b.runoob_count FROM runoob_tbl a INNER JOIN tcount_tbl b ON a.runoob_author = b.runoob_author;
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">等价于: SELECT a.runoob_id, a.runoob_author, b.runoob_count FROM runoob_tbl a, tcount_tbl b WHERE a.runoob_author = b.runoob_author;
```

```
</span></span></code></pre>
```

<p>//使用 LEFT JOIN</p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">例: SELECT a.runoob_id, a.runoob_author, b.runoob_count FROM runoob_tbl a LEFT JOIN tcount_tbl b ON a.runoob_author = b.runoob_author;
```

```
</span></span></code></pre>
```

<p>//使用 RIGHT JOIN</p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">例: SELECT b.runoob_id, b.runoob_author, a.runoob_count FROM tcount_tbl a RIGHT JOIN runoob_tbl b ON a.runoob_author = b.runoob_author;
```

```
</span></span></code></pre>
```

<p>//使用 NULL</p>

<p>必须使用 IS NULL, 或 IS NOT NULL, 或 &lt;=;&gt; 用于比较两个 NULL</p>

<p><strong>//MySQL 正则表达式</strong></p>

<p>MySQL 中使用 REGEXP 操作符来进行正则表达式匹配。</p>

<p>| 模式 | 描述 |<br>

| ^ | 匹配输入字符串的开始位置。如果设置了 RegExp 对象的 Multiline 属性, ^ 也匹配 '\n' 或 '\r' 后的位置。 |<br>

| <span class="language-math"> | 匹配输入字符串的结束位置。如果设置了 RegExp 对象的 Multiline 属性, </span> 也匹配 '\n' 或 '\r' 之前的位置。 |<br>

| . | 匹配除 "\n" 之外的任何单个字符。要匹配包括 '\n' 在内的任何字符, 请使用象 '[\n]' 的模式。 |<br>

| [...] | 字符集合。匹配所包含的任意一个字符。例如, '[abc]' 可以匹配 "plain" 中的 'a'。 |<br>

| [^...] | 负值字符集合。匹配未包含的任意字符。例如, '[^abc]' 可以匹配 "plain" 中的 'p'。 |<br>

| p1|p2|p3 | 匹配 p1 或 p2 或 p3。例如, 'z|food' 能匹配 "z" 或 "food"。'(z|f)ood' 则匹配 "zood" 或 "food"。 |<br>

| \* | 匹配前面的子表达式零次或多次。例如, zo\* 能匹配 "z" 以及 "zoo"。\* 等价于 {0,}。 |<br>

| + | 匹配前面的子表达式一次或多次。例如, 'zo+' 能匹配 "zo" 以及 "zoo", 但不能匹配 "z"。+ 等价于 {1,}。 |<br>

| {n} | n 是一个非负整数。匹配确定的 n 次。例如, 'o{2}' 不能匹配 "Bob" 中的 'o', 但是能匹配 "foo" 中的两个 o。 |<br>

| {n,m} | m 和 n 均为非负整数, 其中 n &lt;= m。最少匹配 n 次且最多匹配 m 次。 |</p>

<p><strong>//MySQL 事务</strong></p>

<p>1, 开始一个事务</p>

<p>start transaction</p>

<p>2, 做保存点</p>

<p>savepoint 保存点名称</p>

<p>3, 操作</p>

<p>4, 可以回滚, 可以提交, 没有问题, 就提交, 有问题就回滚。</p>

<p>例: </p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">
```



```

cl">SET AUTOCOMMIT = 0;          //设置为不自动提交，因为MYSQL默认立即执行
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">BEGIN;          //
始事务定义
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">...;          //操作
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">ROLLBACK;
//有问题回滚
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">COMMIT;
//成功则执行事务
</span></span></code></pre>
<p><strong>//MySQL 的 ALTER 命令</strong></p>
<p>//添加删除表字段</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">ALTER TABLE testalter_tbl DROP i;          //删除i字段，不能删除最后一个字段
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">ALTER TABLE testa
ter_tbl ADD i INT;          //增加i字段并定义类型
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">ALTER TABLE testa
ter_tbl ADD i INT FIRST;          //使用MySQL提供的关键字 FIRST (设定位第一列)
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">ALTER TABLE testa
ter_tbl ADD i INT AFTER c;          //AFTER 字段名（设定位于某个字段之后）
</span></span></code></pre>
<p>//修改字段类型和名称</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">ALTER TABLE testalter_tbl MODIFY c CHAR(10);          //把字段 c 的类型从 CHAR(1) 改为
CHAR(10)
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">ALTER TABLE testa
ter_tbl CHANGE i j BIGINT;          //把字段i名称改为j，并将类型改为BIGINT
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">ALTER TABLE testa
ter_tbl MODIFY j BIGINT NOT NULL DEFAULT 100;          //指定字段 j 为 NOT NULL 且默认值为10
，若不指定默认值为NULL
</span></span></code></pre>
<p>//修改字段默认值</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">ALTER TABLE testalter_tbl ALTER i SET DEFAULT 1000;          //使用 ALTER 来修改字段的
认值
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">ALTER TABLE testa
ter_tbl ALTER i DROP DEFAULT;          //使用 ALTER 命令及 DROP子句来删除字段的默认值
</span></span></code></pre>
<p>//修改表名</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">ALTER TABLE testalter_tbl RENAME TO alter_tbl;          //将数据表 testalter_tbl 重命名为 alter
tbl
</span></span></code></pre>
<p><strong>//MySQL 创建索引</strong></p>
<p>//普通索引</p>

```

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">CREATE INDEX indexName ON tableName(columnName(length)); //创建索引, 方法1
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">ALTER TABLE tabl
Name ADD INDEX indexName(columnName(length)); //修改表结构, 方法2
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">CREATE TABLE my
able( ID INT NOT NULL, username VARCHAR(16) NOT NULL, INDEX [indexName] (username(
ength)); //创建表时直接指定, 方法3
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">DROP INDEX inde
Name ON tableName; //删除索引
</span></span></code></pre>
```

<p>//唯一索引</p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">ALTER TABLE tbl_name ADD PRIMARY KEY (column_list); //添加一个主键, 要确保
键不为NULL
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">CREATE UNIQUE
NDEX indexName ON mytable(username(length)); //创建索引, 方法1
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">ALTER table mytab
e ADD UNIQUE [indexName] (username(length)); //修改表结构, 方法2
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">CREATE TABLE my
able( ID INT NOT NULL, username VARCHAR(16) NOT NULL, UNIQUE [indexName] (usernam
(length)); //创建表时指定, 方法3
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> ALTER TABLE tbl_
ame ADD FULLTEXT index_name (column_list); //添加全文索引
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">ALTER TABLE tbl_
ame DROP INDEX (column_list); //删除索引
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">ALTER TABLE tbl_
ame DROP PRIMARY KEY; //删除主键
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">SHOW INDEX FR
M tableName\G; //显示索引信息
</span></span></code></pre>
```

<p><strong>//MySQL 临时表</strong></p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">CREATE TEMPORARY TABLE tableName (...); //创建临时表
</span></span></code></pre>
```

<p>删除临时表与删除其他表一样, 但 show tables 不能显示临时表</p>

<p><strong>//MySQL 复制表</strong></p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">SHOW CREATE TABLE runoob_tbl \G; //获取创建数据表(CREATE TABLE) 语句
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">复制以下命令显示
SQL语句, 修改数据表名, 并执行SQL语句, 通过以上命令 将完全的复制数据表结构。
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">INSERT INTO ... SE
ECT... 语句来实现复制表的内容
```

```

</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">mysql&gt; INSERT
INTO clone_tbl (runoob_id,-&gt; runoob_title,-&gt; runoob_author,-&gt; submission_date)-&
t; SELECT runoob_id,runoob_title,-&gt; runoob_author,submission_date -&gt; FROM runoob_t
l;
</span></span></code></pre>
<p><strong>//获取服务器元数据</strong></p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">SELECT VERSION();          //服务器版本信息
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">SELECT DATABASE
);          //当前数据库名
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">SELECT USER();
//当前用户名
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">SHOW STATUS;
//服务器状态
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">SHOW VARIABLES
//服务器配置变量
</span></span></code></pre>
<p><strong>//MySQL 序列使用</strong></p>
<p>使用 AUTO_INCREMENT 定义列</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">LAST_INSERT_ID()          //获取最后的插入表中的自增的值的函数
</span></span></code></pre>
<p>重置序列: </p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">mysql&gt; ALTER TABLE insect DROP id; mysql&gt; ALTER TABLE insect -&gt; ADD id INT
UNSIGNED NOT NULL AUTO_INCREMENT FIRST,-&gt; ADD PRIMARY KEY (id);
</span></span></code></pre>
<p>设置序列的开始值: </p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">mysql&gt; ALTER TABLE t AUTO_INCREMENT = 100;
</span></span></code></pre>
<p>或者在创建时直接 AUTO_INCREMENT = 100</p>
<p><strong>//MySQL 处理重复数据</strong></p>
<p>防止表中出现重复数据</p>
<p>防止重复插入: </p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">INSERT IGNORE INTO
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">REPLACE INTO
</span></span></code></pre>
<p>设置唯一索引: </p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">CREATE TABLE person_tbl ( first_name CHAR(20) NOT NULL, last_name CHAR(20) NOT N
LL, sex CHAR(10) UNIQUE (last_name, first_name));
</span></span></code></pre>
<p>统计重复数据: </p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">mysql&gt; SELECT COUNT(*) as repetitions, last_name, first_name -&gt; FROM person_tbl
-&gt; GROUP BY last_name, first_name -&gt; HAVING repetitions &gt; 1;

```

```

</span></span></code></pre>
<p>过滤重复数据: </p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">mysql> SELECT DISTINCT last_name, first_name -&gt; FROM person_tbl -&gt; ORDER
Y last_name;
</span></span></code></pre>
<p>或使用 GROUP BY 来读取</p>
<p>删除重复数据: </p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">mysql> CREATE TABLE tmp SELECT last_name, first_name, sex -&gt; FROM person_tbl;
&gt; GROUP BY (last_name, first_name); mysql> DROP TABLE person_tbl; mysql> ALTER
TABLE tmp RENAME TO person_tbl;
</span></span></code></pre>
<p>或者通过添加主键/索引的方法: </p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">mysql> ALTER IGNORE TABLE person_tbl -&gt; ADD PRIMARY KEY (last_n
me);
</span></span></code></pre>
<p><strong>//MySQL 导出数据</strong></p>
<p><strong>使用 SELECT ... INTO OUTFILE ...语句导出数据</strong></p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">mysql> SELECT * FROM runoob_tbl -&gt; INTO OUTFILE '/tmp/tutorials.txt';
</span></span></code></pre>
<p>指定格式, 如 CSV 格式: </p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">mysql> SELECT * FROM passwd INTO OUTFILE '/tmp/tutorials.txt'-&gt; FIELDS TERMI
ATED BY ',' ENCLOSED BY "''-&gt; LINES TERMINATED BY '\r\n';
</span></span></code></pre>
<p>或</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">SELECT a,b,a+b INTO OUTFILE '/tmp/result.text' FIELDS TERMINATED BY ',' OPTIONALLY
NCLOSED BY "' ' LINES TERMINATED BY '\n' FROM test_table;
</span></span></code></pre>
<p>使用 LOAD DATA INFILE 可以将文件读回数据库</p>
<p><strong>导出表作为原始数据: </strong></p>
<p>mysqldump 是 mysql 用于转存储数据库的实用程序。它主要产生一个 SQL 脚本, 其中包含从
重新创建数据库所必需的命令 CREATE TABLE INSERT 等。</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">$ mysqldump -u root -p --no-create-info \ --tab=/tmp RUNOOB runoob_tbl
</span></span><span class="highlight-line"><span class="highlight-cl">password *****
</span></span></code></pre>
<p>导出 SQL 格式的数据: </p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">$ mysqldump -u root -p RUNOOB runoob_tbl &gt; dump.txt
</span></span><span class="highlight-line"><span class="highlight-cl">password *****
</span></span></code></pre>
<p>导出整个数据库: </p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">$ mysqldump -u root -p RUNOOB &gt; database_dump.txt
</span></span><span class="highlight-line"><span class="highlight-cl">password *****
</span></span></code></pre>
<p>导出所有数据库: </p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">$ mysqldump -u root -p --all-databases &gt; database_dump.txt

```

```

</span></span><span class="highlight-line"><span class="highlight-cl">password *****
</span></span></code></pre>
<p>将数据库拷贝到其他主机: </p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">$ mysqldump -u root -p database_name table_name &gt; dump.txt
</span></span><span class="highlight-line"><span class="highlight-cl">password *****
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">$ mysql -u root -p
database_name &lt; dump.txt
</span></span><span class="highlight-line"><span class="highlight-cl">password *****
</span></span></code></pre>
<p>你也可以使用以下命令将导出的数据直接导入到远程的服务器上, 但请确保两台服务器是相通
的是可以相互访问的: p&gt; $ mysqldump -u root -p database_name \ | mysql -h other-host.com
database_name</p>
<p><strong>//MySQL 导入数据</strong></p>
<p><strong>使用 LOAD DATA 导入数据</strong></p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">mysql&gt; LOAD DATA LOCAL INFILE 'dump.txt' INTO TABLE mytbl;
</span></span></code></pre>
<p>如果指定 LOCAL 关键词, 则表明从客户主机上按路径读取文件。如果没有指定, 则文件在服务
上按路径读取文件。</p>
<p>指定 FIELDS 和 LINES</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">mysql&gt; LOAD DATA LOCAL INFILE 'dump.txt' INTO TABLE mytbl -&gt; FIELDS TERMIN
TED BY ':'-&gt; LINES TERMINATED BY '\r\n';
</span></span></code></pre>
<p>指定顺序</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">mysql&gt; LOAD DATA LOCAL INFILE 'dump.txt' -&gt; INTO TABLE mytbl (b, c, a);
</span></span></code></pre>
<p><strong>使用 mysqlimport 导入数据</strong></p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">$ mysqlimport -u root -p --local --fields-terminated-by=":" \ --lines-terminated-by="\r\n
database_name dump.txt
</span></span><span class="highlight-line"><span class="highlight-cl">password *****
</span></span></code></pre>
<p>使用 --columns 选项来设置列的顺序</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">$ mysqlimport -u root -p --local --columns=b,c,a \
</span></span><span class="highlight-line"><span class="highlight-cl"> database_name
dump.txt
</span></span><span class="highlight-line"><span class="highlight-cl">password *****
</span></span></code></pre>
<p>mysqlimport 的常用选项介绍</p>
<p>| 选项 | 功能 |<br>
| -d or --delete | 新数据导入数据表中之前删除数据表中的所有信息 |<br>
| -f or --force | 不管是否遇到错误, mysqlimport 将强制继续插入数据 |<br>
| -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
略。 |<br>
| -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
受到影响。 |<br>
| -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。 |<b
>
| --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

```

号括起。默认的情况下数据是没有被字符括起的。 |<br>

| --fields-terminated-by=char | 指定各个数据的值之间的分隔符，在逗号分隔的文件中，分隔符逗号。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) |<br>

| --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：一个换行或者一个回车。 |</p>

<p><strong>SQL 数据类型</strong></p>

<p>| INTEGER 或 INT | 通常为 32 位整数 |<br>

| SMALLINT | 通常为 16 位整数 |<br>

| NUMERIC(m, n), DECIMAL(m, n) 或 DEC(m, n) | m 位长的定点十进制数，其中小数点后为 n 位 |<br>

| FLOAT(n) | 运算精度为 n 位二进制数的浮点数 |<br>

| REAL | 通常为 32 位浮点数 |<br>

| DOUBLE | 通常为 64 位浮点数 |<br>

| CHARACTER(n)或 CHAR(n) | 固定长度为 n 的字符串 |<br>

| VARCHAR(n) | 最大长度为 n 的字符串 |<br>

| BOOLEAN | 布尔值 |<br>

| DATE | 日历日期 |<br>

| TIME | 当前时间 |<br>

| TIMESTAMP | 当前日期和时间 |<br>

| BLOB | 二进制大对象 |<br>

| CLOB | 字符大对象 |</p>