**1、使用SHOW语句找出在服务器上当前存在什么数据库：**

|  |
| --- |
| mysql> SHOW DATABASES; +----------+ | Database | +----------+ | mysql | | test | +----------+ 3 rows in set (0.00 sec) |

**2、创建一个数据库abccs**   
    mysql> CREATE DATABASE abccs;   
    注意不同操作系统对大小写的敏感。

**3、选择你所创建的数据库**   
    mysql> USE abccs   
    Database changed   
    此时你已经进入你刚才所建立的数据库abccs.

**4、 创建一个数据库表**

    首先看现在你的数据库中存在什么表：   
    mysql> SHOW TABLES;   
    Empty set (0.00 sec)   
    说明刚才建立的数据库中还没有数据库表。下面来创建一个数据库表mytable:   
    我们要建立一个你公司员工的生日表，表的内容包含员工姓名、性别、出生日期、出生城市。

|  |
| --- |
| mysql> CREATE TABLE mytable (name VARCHAR(20), sex CHAR(1), -> birth DATE, birthaddr VARCHAR(20)); Query OK, 0 rows affected (0.00 sec) |

    由于name、birthadd的列值是变化的，因此选择VARCHAR，其长度不一定是20。可以选择从1到255的任何长度，如果以后需要改变它的字长，可以使用ALTER TABLE语句。）;性别只需一个字符就可以表示："m"或"f"，因此选用CHAR(1);birth列则使用DATE数据类型。

    创建了一个表后，我们可以看看刚才做的结果，用SHOW TABLES显示数据库中有哪些表：

|  |
| --- |
| mysql> SHOW TABLES; +---------------------+ | Tables in menagerie | +---------------------+ | mytables | +---------------------+ |

**5、显示表的结构：**

|  |
| --- |
| mysql> DESCRIBE mytable; +-------------+-------------+------+-----+---------+-------+ | Field | Type | Null | Key | Default | Extra | +-------------+-------------+------+-----+---------+-------+ | name | varchar(20) | YES | | NULL | | | sex | char(1) | YES | | NULL | | | birth | date | YES | | NULL | | | deathaddr | varchar(20) | YES | | NULL | | +-------------+-------------+------+-----+---------+-------+ 4 rows in set (0.00 sec) |

**6、 往表中加入记录**

    我们先用SELECT命令来查看表中的数据：   
    mysql> select \* from mytable;   
    Empty set (0.00 sec)  
    这说明刚才创建的表还没有记录。

    加入一条新记录：

|  |
| --- |
| mysql> insert into mytable -> values (′abccs′,′f′,′1977-07-07′,′china′); Query OK, 1 row affected (0.05 sec) |

    再用上面的SELECT命令看看发生了什么变化。我们可以按此方法一条一条地将所有员工的记录加入到表中。

**7、用文本方式将数据装入一个数据库表**

    如果一条一条地输入，很麻烦。我们可以用文本文件的方式将所有记录加入你的数据库表中。创建一个文本文件“mysql.txt”，每行包含一个记录，用定位符(tab)把值分开，并且以在CREATE TABLE语句中列出的列次序给出，例如：

|  |
| --- |
| abccs f 1977-07-07 china 　 mary f 1978-12-12 usa tom m 1970-09-02 usa |

    使用下面命令将文本文件“mytable.txt”装载到mytable表中:mysql> LOAD DATA LOCAL INFILE "mytable.txt" INTO TABLE pet;

    再使用如下命令看看是否已将数据输入到数据库表中：mysql> select \* from mytable;

一, 创建用户:

       命令:CREATE USER ['username'@'host](mailto:%20username%20@%20host)' IDENTIFIED BY 'password';

       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,如果是本地用户可用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 - 该用户的登陆密码,密码可以为空,如果为空则该用户可以不需要密码登陆服务器.

      例子: CREATE USER ['dog'@'localhost](mailto:%20dog%20@%20localhost)' IDENTIFIED BY '123456';  
               CREATE USER ['](mailto:%20pig%20@%20192.168.1.101_)[pig'@'192.168.1.101](mailto:%20pig%20@%20192.168.1.101)\_' IDENDIFIED BY '123456';  
               CREATE USER ['pig'@'%](mailto:%20pig%20@%20%25)' IDENTIFIED BY '123456';  
               CREATE USER ['pig'@'%'](mailto:%20pig%20@%20%25) IDENTIFIED BY '';  
               CREATE USER ['pig'@'%'](mailto:%20pig%20@%20%25);

二,授权:

      命令:GRANT privileges ON databasename.tablename TO ['username'@'host'](mailto:%20username%20@%20host)

     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).如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有数据库和表的相应操作权限则可用\*表示, 如\*.\*.

      例子: GRANT SELECT, INSERT ON test.user TO ['pig'@'%'](mailto:%20pig%20@%20%25);  
               GRANT ALL ON \*.\* TO ['pig'@'%'](mailto:%20pig%20@%20%25);

      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:  
           GRANT privileges ON databasename.tablename TO ['username'@'host'](mailto:%20username%20@%20host) WITH GRANT OPTION;

三.设置与更改用户密码

     命令:SET PASSWORD FOR ['username'@'host'](mailto:%20username%20@%20host) = PASSWORD('newpassword');如果是当前登陆用户用SET PASSWORD = PASSWORD("newpassword");

      例子: SET PASSWORD FOR ['pig'@'%'](mailto:%20pig%20@%20%25) = PASSWORD("123456");

四.撤销用户权限

      命令: REVOKE privilege ON databasename.tablename FROM ['username'@'host'](mailto:%20username%20@%20host);

     说明: privilege, databasename, tablename - 同授权部分.

      例子: REVOKE SELECT ON \*.\* FROM ['pig'@'%'](mailto:%20pig%20@%20%25);

      注意: 假如你在给用户['pig'@'%'](mailto:%20pig%20@%20%25)授权的时候是这样的(或类似的):GRANT SELECT ON test.user TO ['pig'@'%'](mailto:%20pig%20@%20%25), 则在使用REVOKE SELECT ON \*.\* FROM['pig'@'%'](mailto:%20pig%20@%20%25);命令并不能撤销该用户对test数据库中user表的SELECT 操作.相反,如果授权使用的是GRANT SELECT ON \*.\* TO ['pig'@'%'](mailto:%20pig%20@%20%25);则REVOKE SELECT ON test.user FROM ['pig'@'%'](mailto:%20pig%20@%20%25);命令也不能撤销该用户对test数据库中user表的Select 权限.

      具体信息可以用命令SHOW GRANTS FOR ['pig'@'%'](mailto:%20pig%20@%20%25); 查看.

五.删除用户

      命令: DROP USER ['username'@'host'](mailto:%20username%20@%20host);

附表:在MySQL中的操作权限