MySQL数据库的学习

**（一）安装**

搜索：mysql-5.6.17-winx64.zip 下载安装

类型MySQL，5.6绿色版

安装参考这个博客：http://blog.csdn.net/zhuxiaowei716/article/details/19405955

注意：

①不要删除已有的内容。

②需要修改路径，路径使用自己解压缩后（文件）的路径

备注：卸载重装请参考本链接：

http://www.jb51.net/article/32016.htm

1。在“运行”里键入regedit进入注册表。   
2。HEKY\_LOCAL\_MACHINE/SOFTWARE/MYSQL目录删除   
3。 HKEY\_LOCAL\_MACHINE/SYSTEM/ControlSet001/Services/Eventlog/Application/MySQL 目录删除   
4。HKEY\_LOCAL\_MACHINE/SYSTEM/ControlSet002/Services/Eventlog /Application/MySQL 目录删除   
5。HKEY\_LOCAL\_MACHINE/SYSTEM/CurrentControlSet /Services/Eventlog/Application/MySQL目录删除注册表中的   
ControlSet001,ControlSet002,不一定是001和002,可能是ControlSet005、006之类,删除的时候,都删除就可以   
6。删除C:/Documents and Settings/All Users/Application Data/下的MySQL文件夹   
7。最后卸载MySQL。   
8。删除MySQL安装路径下的文件夹。   
9。最好重启下电脑。

**（二）启动**

我是直接启动：D:\python\MySQL Server 5.6\bin\mysql.exe

就可以的（进入了正常使用的页面）。

D:\python\MySQL Server 5.6是我的数据库安装目录。

以下是我的启动页面。如果无法正常启动的话，请评论说明，我再重新研究一下。

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 5

Server version: 5.6.17 MySQL Community Server (GPL)

Copyright (c) 2000, 2014, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its

affiliates. Other names may be trademarks of their respective

owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>

**（三）显示数据库**

我启动时自动进入页面，当时页面为：

mysql>

在这个后面输入命令，以**分号为结尾**。

输入

show databases;

显示：

mysql> show databases;

+--------------------+

| Database |

+--------------------+

| information\_schema |

| test |

+--------------------+

2 rows in set (0.00 sec)

**（四）进入某个数据库**

以上面的为例，进入test的命令为：

use test;

显示：

mysql> use test;

Database changed

**（五）显示该数据库内的表**

输入：

show tables;

显示：

mysql> show tables;

Empty set (0.00 sec)

原因是是一个新数据库，所以里面没有表。

**（六）创建一个表**

输入：（可以多行输入，只要连续，且中间不写分号就行）

create table sales\_rep(

employee\_number int,

surname varchar(40),

first\_name varchar(30),

commission tinyint);

显示：（这里没换行）

mysql> create table sales\_rep(

-> employee\_number int,

-> surname varchar(40),

-> first\_name varchar(30),

-> commission tinyint);

Query OK, 0 rows affected (0.23 sec)

效果：

创建一个标明为users的表，

第一项：employee\_number，类型为int

第二项：surname，类型为varchar(40)

第三项：first\_name，类型为varchar(30)

第四项：comisision，类型为tinyint

**（七）检查表结构**

输入：

describe sales\_rep;

显示：

mysql> describe sales\_rep;

+-----------------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+-----------------+-------------+------+-----+---------+-------+

| employee\_number | int(11) | YES | | NULL | |

| surname | varchar(40) | YES | | NULL | |

| first\_name | varchar(30) | YES | | NULL | |

| commission | tinyint(4) | YES | | NULL | |

+-----------------+-------------+------+-----+---------+-------+

4 rows in set (0.00 sec)

**（八）类型说明**

**int：**范围-2 147 483 648 到 2 147 483 647（包括负号所以11个字符）

**varchar(40)：**表示可变长字符，40表示字符串的最大长度是40

**tinvint：**表示小整数范围是-128~127之间（所以是4个字符）。

**（九）更改表内某一项的名字和类型**

如上面所见：第一项实际应该是employes\_num，我错写为了employee\_number

可以通过命令修改，使其名字变更为正确的名字。

原型：

>alter table 表名称 change 字段原名称 字段新名称 字段类型 [是否允许非空];

输入：

alter table sales\_rep change employee\_number employes\_number int;

显示：

mysql> alter table sales\_rep change employee\_number employes\_number int

-> ;

Query OK, 0 rows affected (0.13 sec)

Records: 0 Duplicates: 0 Warnings: 0

这时再查询表结构describe sales\_rep;，则正确了（结果略）

**（十）向表单内插入**

**注：这个不能用中文（可能是没设置），varchar的值可以用双引号**

**方法一：（按自定义顺序插入）**

输入：

**insert into sales\_rep(employes\_number, surname, first\_name, commission) values(3,'Jop','Peter',12);**

显示：

mysql> insert into sales\_rep(employes\_number, surname, first\_name, commission)

values(3,'Jop','Peter',12);

Query OK, 1 row affected (0.04 sec)

效果：

插入一个值，前面4个参数分别对应values后面的4个值。后面值的对应是参数的类型。

**方法二：（按默认顺序插入）**

输入：

**insert into sales\_rep values(4,"dong", "wang", 20);**

显示：

mysql> insert into sales\_rep values(4,"dong", "wang", 20);

Query OK, 1 row affected (0.03 sec)

效果：插入一个值，值的顺序是默认的（按照之前表的预设）

**方法三：（一次插入多个）**

输入：

**insert into sales\_rep values(5,"aaa","AAA",1),**

**(6,"bbb","BB",2),**

**(7,"ccc","CC",3);**

显示：

mysql> insert into sales\_rep values(5,"aaa","AAA",1),

-> (6,"bbb","BB",2),

-> (7,"ccc","CC",3);

Query OK, 3 rows affected (0.15 sec)

Records: 3 Duplicates: 0 Warnings: 0

效果：按默认顺序插入3个值。

**方法四：（用load data读取文本插入大量数据）**

输入：

**load data local infile "eee.txt" into table sales\_rep;**

显示：

mysql> load data local infile "eee.txt" into table sales\_rep;

Query OK, 4 rows affected, 4 warnings (0.04 sec)

Records: 4 Deleted: 0 Skipped: 0 Warnings: 4

效果：

将eee.txt文件内的3项读取进去

**问题：**

**为什么会有4个warnings？并没有读取失败啊？**

注：显示（十一），删除（二十一），修改（二十二）

**（十一）使用select显示**

输入：

**select \* from sales\_rep where commission>1;**

显示：

mysql> select \* from sales\_rep where commission>1;

+-----------------+---------+------------+------------+

| employes\_number | surname | first\_name | commission |

+-----------------+---------+------------+------------+

| 1 | Rive | Sol | 10 |

| 2 | John | char | 11 |

| 3 | Jop | Peter | 12 |

| 4 | dong | wang | 20 |

| 6 | bbb | BB | 2 |

| 7 | ccc | CC | 3 |

| 8 | ddd | DDD | 4 |

| 9 | ee | EEEE | 5 |

| 10 | ff | FFF | 6 |

+-----------------+---------+------------+------------+

9 rows in set (0.00 sec)

解释：

①select：表示使用select语句

②\*：表示通配符（显示所有的）。可以用某一项替换，如surname，则只显示surname这一项；

③from sales\_rep：指从表名为sales\_rep的这一张表读取内容；

④where commission>1：指的是，读取表内commission这一个属性大于1的所有项；

也可以改为：where first\_name='cc' ：那么就只显示first\_name的值是cc的项了（注意需要加引号）（并且不区分大小写，注意，mysql几乎是不区分大小写的）

⑤;：以分号为结尾。

**（十二）where从句**

基于（十一）

输入：

**select \* from sales\_rep where first\_name!='cc' or first\_name="cc";**

显示：

mysql> select \* from sales\_rep where first\_name!='cc' or first\_name="cc";

+-----------------+---------+------------+------------+

| employes\_number | surname | first\_name | commission |

+-----------------+---------+------------+------------+

| 1 | Rive | Sol | 10 |

| 2 | John | char | 11 |

| 3 | Jop | Peter | 12 |

| 4 | dong | wang | 20 |

| 5 | aaa | AAA | 1 |

| 6 | bbb | BB | 2 |

| 7 | ccc | CC | 3 |

| 8 | ddd | DDD | 4 |

| 9 | ee | EEEE | 5 |

| 10 | ff | FFF | 6 |

+-----------------+---------+------------+------------+

10 rows in set (0.00 sec)

解释：

①or：前后各有一条语句**first\_name!='cc'**和**first\_name="cc"**，

②or的效果是两条语句只要有一条成立，便符合条件。由于一个内容要么是cc，要么不是cc（必然的），所以每一项都会成立（因此相当于**显示除了null的**）。

③and：也可以用and来连接，表示两条语句都符合的才行。（有一条不符合就不成立）。

④多个or和and联用：支持联用。

**or的联用：**mysql> select \* from sales\_rep where **first\_name='bb'** or **first\_name="cc"** or **first\_name="aaa"**; 三者符合一个的就可以。

**and联用：**由此可推，可以多个and联用。

**也可以or和and混合使用**，但为了避免理解出错，请使用括号来区分层级。

**（十三）通配符%**

%是一个通配符，类似\*，表示0个或者多个字符。

输入：

**select \* from sales\_rep where first\_name like "%c%";**

显示：

mysql> select \* from sales\_rep where first\_name like "%c%";

+-----------------+---------+------------+------------+

| employes\_number | surname | first\_name | commission |

+-----------------+---------+------------+------------+

| 2 | John | char | 11 |

| 7 | ccc | CC | 3 |

+-----------------+---------+------------+------------+

2 rows in set (0.00 sec)

效果：

first\_name属性里有c字母的项，全部都是符合条件的。

**（十四）分类**

**排序：（正序）**

输入：

**select \* from sales\_rep order by first\_name;**

显示：

mysql> select \* from sales\_rep order by first\_name;

+-----------------+---------+------------+------------+

| employes\_number | surname | first\_name | commission |

+-----------------+---------+------------+------------+

| 0 | NULL | NULL | NULL |

| 8 | NULL | NULL | NULL |

| 5 | aaa | AAA | 1 |

| 6 | bbb | BB | 2 |

| 7 | ccc | CC | 3 |

| 2 | John | char | 11 |

| 8 | ddd | DDD | 4 |

| 9 | ee | EEEE | 5 |

| 10 | ff | FFF | 6 |

| 3 | Jop | Peter | 12 |

| 1 | Rive | Sol | 10 |

| 4 | dong | wang | 20 |

+-----------------+---------+------------+------------+

15 rows in set (0.00 sec)

效果：

在first\_name项，以字母顺序进行排序（会显示null项）；

**排序：（多重排序）**

输入：

**select \* from sales\_rep order by surname,first\_name;**

显示：

+-----------------+---------+------------+------------+

| employes\_number | surname | first\_name | commission |

+-----------------+---------+------------+------------+

| 0 | NULL | NULL | NULL |

| 8 | NULL | NULL | NULL |

| 5 | aaa | AAA | 1 |

| 5 | bbb | AAA | 5 |

| 6 | bbb | BB | 2 |

| 7 | ccc | CC | 3 |

| 8 | ddd | DDD | 4 |

| 4 | dong | wang | 20 |

| 9 | ee | EEEE | 5 |

| 10 | ff | FFF | 6 |

| 2 | John | char | 11 |

| 3 | Jop | Peter | 12 |

| 1 | Rive | Sol | 10 |

+-----------------+---------+------------+------------+

16 rows in set (0.00 sec)

效果说明：

①多了一个surname作为参数。效果是，先以第一个参数所在的项为顺序进行排序显示，假如第一个一样，以第二个参数进行排序；

②由此类推，可以有更多项。

**排序：（倒序）**

输入：

**select \* from sales\_rep order by first\_name desc;**

显示：

mysql> select \* from sales\_rep order by first\_name desc;

+-----------------+---------+------------+------------+

| employes\_number | surname | first\_name | commission |

+-----------------+---------+------------+------------+

| 4 | dong | wang | 20 |

| 1 | Rive | Sol | 10 |

| 3 | Jop | Peter | 12 |

| 10 | ff | FFF | 6 |

| 9 | ee | EEEE | 5 |

| 8 | ddd | DDD | 4 |

| 2 | John | char | 11 |

| 7 | ccc | CC | 3 |

| 6 | bbb | BB | 2 |

| 5 | bbb | AAA | 5 |

| 5 | aaa | AAA | 1 |

| 0 | NULL | NULL | NULL |

| 8 | NULL | NULL | NULL |

+-----------------+---------+------------+------------+

16 rows in set (0.00 sec)

区别：

在正序的命令后面加一个desc即可。

**排序：（正序关键字）**

假如需要第一个参数按正序，相同的情况下，第二个参数按反序进行排列。

输入：

**select \* from sales\_rep order by surname asc, employes\_number desc;**

显示：

mysql> select \* from sales\_rep order by surname asc, employes\_number desc;

+-----------------+---------+------------+------------+

| employes\_number | surname | first\_name | commission |

+-----------------+---------+------------+------------+

| 10 | NULL | NULL | NULL |

| 9 | NULL | NULL | NULL |

| 8 | NULL | NULL | NULL |

| 0 | NULL | NULL | NULL |

| 0 | NULL | NULL | NULL |

| 5 | aaa | AAA | 1 |

| 6 | bbb | BB | 2 |

| 5 | bbb | AAA | 5 |

| 7 | ccc | CC | 3 |

| 8 | ddd | DDD | 4 |

| 4 | dong | wang | 20 |

| 9 | ee | EEEE | 5 |

| 10 | ff | FFF | 6 |

| 2 | John | char | 11 |

| 3 | Jop | Peter | 12 |

| 1 | Rive | Sol | 10 |

+-----------------+---------+------------+------------+

16 rows in set (0.00 sec)

**（十五）限制返回的项数LIMIT**

**输出行数限制：**

输入：（在**排序：（正序关键字）**基础上修改）

select \* from sales\_rep order by surname asc, employes\_number desc limit 1;

显示：

mysql> select \* from sales\_rep order by surname asc, employes\_number desc limit

1;

+-----------------+---------+------------+------------+

| employes\_number | surname | first\_name | commission |

+-----------------+---------+------------+------------+

| 10 | NULL | NULL | NULL |

+-----------------+---------+------------+------------+

1 row in set (0.00 sec)

解释：

①limit可以设置为0（虽然不显示内容，但是可以用于在大的数据库中进行测试查询）；

②limit后面的数字决定了返回的行数；

**输出行数限制+偏移限制：**

输入：

**select \* from sales\_rep order by surname asc, employes\_number desc limit 1,2;**

显示：

mysql> select \* from sales\_rep order by surname asc, employes\_number desc limit

1,2;

+-----------------+---------+------------+------------+

| employes\_number | surname | first\_name | commission |

+-----------------+---------+------------+------------+

| 9 | NULL | NULL | NULL |

| 8 | NULL | NULL | NULL |

+-----------------+---------+------------+------------+

2 rows in set (0.00 sec)

解释：

①参数1表示从开始偏移几行（即从理论应该显示的结果的第几行开始显示）；

②参数2表示显示几行。

③1,2表示从第2行（偏移1）开始显示，显示2行

**（十六）返回最大值MAX()**

输入：

**select max(commission) from sales\_rep;**

显示：

mysql> select max(commission) from sales\_rep;

+-----------------+

| max(commission) |

+-----------------+

| 20 |

+-----------------+

1 row in set (0.00 sec)

效果：

返回commission这一项中最大的值。（对字母也有效）

**（十七）防止显示重复记录distinct**

输入：

**select distinct first\_name from sales\_rep;**

显示：

mysql> select distinct first\_name from sales\_rep;

+------------+

| first\_name |

+------------+

| Sol |

| char |

| Peter |

| wang |

| AAA |

| BB |

| CC |

| NULL |

| DDD |

| EEEE |

| FFF |

+------------+

11 rows in set (0.00 sec)

效果：

①原本都个NULL，这里只显示一个了；

②原本2个AAA，这里只显示一个了；

**（十八）计数count()**

**显示有效数据的行数（不显示null）：**

输入：

**select count(first\_name) from sales\_rep;**

显示：

mysql> select count(first\_name) from sales\_rep;

+-------------------+

| count(first\_name) |

+-------------------+

| 11 |

+-------------------+

1 row in set (0.00 sec)

效果：

①除了null都显示；

**显示所有数据的行数（貌似除非都是null，否则就显示）：**

输入：

**select count(\*) from sales\_rep;**

显示：

mysql> select count(\*) from sales\_rep;

+----------+

| count(\*) |

+----------+

| 16 |

+----------+

1 row in set (0.00 sec)

效果：

①即使有一项是null（其他项不是），也被计算在内；

**显示行数：（重复项算一项，使用distinct）**

输入：

**select count(distinct first\_name) from sales\_rep;**

显示：

mysql> select count(distinct first\_name) from sales\_rep;

+----------------------------+

| count(distinct first\_name) |

+----------------------------+

| 10 |

+----------------------------+

1 row in set (0.00 sec)

效果：

①该项（first\_name）中如果有重复的，记作一项。

**（十九）平均、最小、总和**

**平均：avg()**

**最小：min()**

**总和：sum()**

输入：

**select sum(commission) from sales\_rep;**

显示：

mysql> select sum(commission) from sales\_rep;

+-----------------+

| sum(commission) |

+-----------------+

| 79 |

+-----------------+

1 row in set (0.00 sec)

效果：

①三个方法类似，结果不同；

②对字符串无效

**（二十）对计算结果修改（不影响原数据）**

输入：

**select employes\_number ,surname, commission+100 from sales\_rep;**

显示：

mysql> select employes\_number ,surname, commission+100 from sales\_rep;

+-----------------+---------+----------------+

| employes\_number | surname | commission+100 |

+-----------------+---------+----------------+

| 1 | Rive | 110 |

| 2 | John | 111 |

| 3 | Jop | 112 |

| 4 | dong | 120 |

| 5 | aaa | 101 |

| 6 | bbb | 102 |

| 7 | ccc | 103 |

| 8 | NULL | NULL |

| 9 | NULL | NULL |

| 10 | NULL | NULL |

| 0 | NULL | NULL |

| 8 | ddd | 104 |

| 9 | ee | 105 |

| 10 | ff | 106 |

| 0 | NULL | NULL |

| 5 | bbb | 105 |

+-----------------+---------+----------------+

16 rows in set (0.00 sec)

效果：

①指定项显示结果 + 100，（**commission+100**）；

②显示选择的项（employes\_number ,surname, commission）；

③不影响原数据；

**（二十一）删除记录**

输入：

**delete from sales\_rep where surname="aaa";**

显示：

mysql> delete from sales\_rep where surname="aaa";

Query OK, 1 row affected (0.06 sec)

效果：

①删除值为null的项：delete from sales\_rep where surname **is null;**

②删除记录将删除所有符合要求的（所以尽量使用唯一项，例如类似id的东西）；

③假如输入**delete from sales\_rep** 将删除**所有项**。

**（二十二）修改记录**

输入：

**update sales\_rep set commission = 20 where employes\_number=1;**

显示：

mysql> update sales\_rep set commission = 20 where employes\_number=1;

Query OK, 1 row affected (0.06 sec)

Rows matched: 1 Changed: 1 Warnings: 0

效果：

①employes\_number的值是1的项，其commission的值被更改为20。

**（二十三）删除一张表**

警告：删除无法被恢复

输入：

**drop table sales\_rep;**

显示：

mysql> drop table sales\_rep;

Query OK, 0 rows affected (0.16 sec)

效果：

①表sales\_rep;被删除

**（二十四）创建、删除一个database**

警告：删除无法被恢复

**创建：（如果通过cmd打开，请确认使用的是管理员权限）**

输入：

create database temp;

注：temp为创建的数据库的名字

**删除：**

drop database temp;

注：temp为创建的数据库的名字

**（二十五）导入一个外部的sql文件**

先到指定的databases目录下；

然后使用命令：

**source 导入文件的路径**

即可。

例如：

source d:/mysql.sql

**（二十六）导出一个表**

  1、导出數據库為dbname的表结构（其中用戶名為root,密码為dbpasswd,生成的脚本名為db.sql）  
    mysqldump -uroot -pdbpasswd -d dbname >db.sql;  
  
    2、导出數據库為dbname某张表(test)结构  
    mysqldump -uroot -pdbpasswd -d dbname test>db.sql;  
  
    3、导出數據库為dbname所有表结构及表數據（不加-d）  
    mysqldump -uroot -pdbpasswd  dbname >db.sql;  
  
    4、导出數據库為dbname某张表(test)结构及表數據（不加-d）  
    mysqldump -uroot -pdbpasswd dbname test>db.sql;