第一部分：MySQL数据库

数据库说明:

MySQL是一个轻量级的关系型数据库　可以满足所有中小型企业的数据库

特点是： 免费　可以和绝大部分的语言 进行合作　 开源

（一）数据库的进入

mysql -hIP地址 -uroot -p

-h HOST 主机名

-u USER 用户

-p PASSWORD 设置的密码

root用户　叫做超级管理员

root 不可以进行远程访问

远程访问的话 通过你当前的**root**用户 去创建其他的用户 再次访问

例如：

```
1 mysql -uroot -p
2 mysql -h127.0.0.1 -uroot -p
3 mysql -hlocalhost -uroot -p
```

MySQL数据库--》各种存有数据的库--》装有数据的表--》数据

（二）库的操作

对于库/表的操作的命令

create 创建

show 查看

drop 删除

alter 修改

1. 查看所有的数据库

   show databases

2. 使用某一个库

   use 库名

3. 查看当前库下所有的表

类型	大小	范围（有符号）	范围（无符号）	用途

show tables;

4. 查看当前所在的库

   select database()

5. 创建数据库

   create database 库名

   create database if not exists 库名　　防止执行创建已存在的库　报错

6. 查看所创建的库

   show create database 库名

7. 修改库的字符集

   alter database　库名　　character set 字符集

8. 删除库

   drop database 库名

   drop database if exists 库名　防止执行删除不存在的库　　报错

9. 创建数据库 并设置字符集

   create database if not exists 库名 character set utf8;

注意：

1. mysql数据库 下 命令以;作为结束
2. 在Windows下 命令库名...不区分大小写

MySQL表的创建

一、字段类型

1、数值类型：

类型	大小	范围（有符号）	范围（无符号）	用途
tinyint	1字节	-128-127（4位）	0-255（3位）	最小整数
int	2字节	-2147483648，2147483647（11位）	0，4294967295（10位）	大整数值
float(m,n)	4字节			单精度浮点型（浮点数）
double(m,n)	8字节			双精度浮点型（浮点数）
decimal(m,n)	如果m>n,m+2；否则n+2			浮点数（更加精确）

注意：

类型	类型大小	大小范围	格式	用途

1、int(3)或者tinyint(2)：3或者2不会限制你所存储数据的长度；只有配合zerofill零填充的时候，才有意义

2、如果没有配合zerofill的时候，后面不建议有数值

3、decimal在存储浮点数时，更加精确

4、在存储浮点数的时候，如果小数点超出范围，包含5且5以下的都会被舍去，否则会进一位

5、\c：撤销命令；\G：竖着展示

2、字符串类型

类型	大小	用途
char	0-255字节	存储定长字符串
varchar	0-65535字节	变长字符串
text	0-65535字节	长文本数据
blob	0-65535字节	二进制的文本（不建议用）
enum('w','m')	65535个成员	枚举
set('w','m')	64个成员	集合

注意：

（1）char和varchar的区别：

- char和varchar最大的长度：都是255
- char的执行效率高于varchar；varchar相对于char节省存储空间
- 如果使用char传入的数据的长度小于指定的长度时，实际存储长度不够会用空格来填充
- 如果使用varchar传入的数据的长度小于指定的长度时，实际存储长度为传进来的数据的长度

（2）enum和set的区别：

- enum只能选择其给定值中的某一个
- set可以选择给定值中的某几个

3、日期和时间类型（了解）

类型	大小	范围	格式	用途
date	3	1000-01-01/9999-12-31	YYYY-MM-DD	日期值
time	3	-838:59:59/838:59:59	YYYY-MM-SS	时间值
year	1	1901/2155	YYYY	年份值
datetime	4	1000-01-01 00:00:00/9999-12-31 23:59:59	YYYY-MM-DD YYYY-MM-SS	混合日期和时间

注意：

存储日期时，可以用整型int来存储时间戳，这样可以做到：任意转换和计算

二、字段约束

1、unsigned：无符号

只能用于设置数值类型，不允许负数的出现

最大存储的长度会扩大一倍

2、zerofill：零填充

只能用于设置数值类型的时候，如果位数不足，用0填充

3、auto_increment：自增

用于设置字段，自动增长的属性，常用于主键；每增加一条数据，该字段会自动加1

4、default：默认值

如果在插入数据的时候，某些字段没有插入值，那么它的值为你的默认值

只有在插入新数据时，default才会生效；在已插入过的数据里，设置default没效果

5、null 和 not null

默认为null

如果是not null，那么必须给该字段一个值

null的注意事项：

（1）null意味着没有值 或者 未知的值

（2）可以测试某个字段是否为null

（3）不能对null进行算数运算

（4）对null进行运算，结果还为null

三、MySQL的索引

（一）主键索引 primary key

1、作用：确定数据表里的数据记录的位置

2、一般来说，每个表里都会有一个主键索引，候选的约束条件为（not null 和 auto_increment）

（二）唯一索引 unique

1、主要功能：防止重复的数据插入

2、和主键索引的区别：主键索引每个表只能有一个，但是唯一索引可以有多个

（三）普通/常规索引 index

1、常规索引：是关系型数据中最重要的技术，如果想要提升数据库的效率，首先就是常规索引

2、缺点：（1）多占用磁盘空间

（2）会减慢插入、删除和修改的效率

（四）全文索引 **fulltext**（了解）

例如：

```
注意：在给字段添加索引时，如果没有给对应字段起索引名，那么索引名默认为字段名
Create Table: CREATE TABLE `userinfo` (
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `username` varchar(50) NOT NULL,
  `userpass` varchar(50) NOT NULL,
  `telno` varchar(20) NOT NULL,
  `sex` enum('w','m') NOT NULL DEFAULT 'm',
  `brithday` date NOT NULL DEFAULT '0000-00-00',
  PRIMARY KEY (`id`),
  UNIQUE KEY `telno` (`telno`), #unique key 索引名称（索引字段）
  KEY `username` (`username`),  #index       索引名称（索引字段）
  KEY `userpass` (`userpass`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8
```

四、数据的存储

1、**mysql**的存储是以文件形式进行存储

存储的路径为：C:\ProgramData\MySQL\MySQL Server 5.7\Data\库名\表名

2、**.frm**后缀的文件：存储表的框架结构

3、**innoDB** 表引擎的文件存储结构

.ibd后缀的文件：存储表的数据和索引

4、**MyISAM** 表引擎的文件存储结构

.MYD 后缀的文件：MY Data，用来存储表的数据

.MYI 后缀的文件：MY Index，用来存储表的索引

五、**innoDB**和**MyISAM**引擎的区别

innoDB和MyISAM是mysql非常重要的两个表的存储引擎

1、innoDB 支持事物处理，比较安全

2、MyISAM 引擎执行效率更高，常用于：博客、论坛、微博等；安全性较低，不支持事物处理

3、MySIAM会产生3个文件（.frm 和 .MYD、.MYI），innoDB会产生2个文件（.frm 和 .ibd）

六、事物处理

前期工作：

1、修改当前表的存储引擎是否为**innoDB**

alter table 表名 engine=innodb

2、查询是否为自动提交

select @@autocommit（1为自动提交，0为手动提交）

3、设置手动提交

set autocommit=0

4、事物开始

begin

5、执行事物的代码

代码块

6、执行提交或者回滚

commit work 提交

rollback work 回滚

注意：

1、当表存储引擎为**innoDB**时，未提交就回滚，数据不存在了

```
mysql> alter table user engine=innodb;
Query OK, 1 row affected (0.57 sec)
Records: 1  Duplicates: 0  Warnings: 0

mysql> select @@autocommit;
+--------------+
| @@autocommit |
+--------------+
|            1 |
+--------------+
1 row in set (0.00 sec)

mysql> set autocommit=0;
Query OK, 0 rows affected (0.00 sec)

mysql> begin;
Query OK, 0 rows affected (0.00 sec)

mysql> insert into user values(5,8);
Query OK, 1 row affected (0.04 sec)

mysql> select * from user;
+----+----------+
| id | username |
+----+----------+
|  1 | 2        |
|  5 | 8        |
+----+----------+
2 rows in set (0.00 sec)

mysql> rollback work;
Query OK, 0 rows affected (0.07 sec)

mysql> select * from user;
+----+----------+
| id | username |
+----+----------+
|  1 | 2        |
+----+----------+
```

2、当表存储引擎为**MyISAM**时，在未提交前回滚，数据还在

```
mysql> alter table user engine=myisam;
Query OK, 1 row affected (0.40 sec)
Records: 1  Duplicates: 0  Warnings: 0

mysql> select @@autocommit;
+--------------+
| @@autocommit |
+--------------+
|            0 |
+--------------+
1 row in set (0.00 sec)

mysql> begin;
Query OK, 0 rows affected (0.00 sec)

mysql> insert into user values(3,4);
Query OK, 1 row affected (0.00 sec)

mysql> select * from user;
+----+----------+
| id | username |
+----+----------+
|  1 | 2        |
|  3 | 4        |
+----+----------+
2 rows in set (0.00 sec)

mysql> rollback work;
Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql> select * from user;
+----+----------+
| id | username |
+----+----------+
|  1 | 2        |
|  3 | 4        |
+----+----------+
```

七、对于表结构的操作

1、给表添加新的字段（新添加字段默认放在最后一位）

关键字：add

alter table 表名 add 字段名 字段类型 约束条件

2、添加的字段排在某个位置

alter table 表名 add 字段名 字段类型 约束条件

first：排在第一位

after：排在某个字段后面

3、删除某个字段

关键字：drop

alter table 表名 drop 字段名

4、修改某个字段的类型以及约束条件

关键字：modify

alter table 表名 modify 字段名 约束条件

例如：alter table user modify sex tinyint not null default 1 after username;

5、修改字段名称

关键字：change

alter table 表名 change 旧字段 新字段 字段类型 约束条件

例如：alter table user change sex mysex tinyint;

6、添加索引

说明：创建常规索引，可以使用index和key来进行创建

（1）起索引名称：alter table 表名 add 索引类型 索引名称（索引字段）

例如：alter table user add index myusername(username);

（2）不起索引名称：alter table 表名 add 索引类型 （索引字段）

例如：alter table user add index(username);

7、删除索引

alter table 表名 drop key 索引名

例如：alter table user drop key username;

8、表的重命名

关键字：rename

alter table 表名 rename 新表名

9、创建一个和**a**一样的表结构表**b**

create table b like a

MySQL数据的操作

数据的增删改查的单词：

增：insert into

删：delete

改：update

查：select

一、INSERT数据的添加：

主体结构：

1、指定字段添加值：

insert into 表名（字段1[,字段2]） values（值1[,值2]）

2、不指定字段添加值：(全部字段都添加值)

insert into 表名 values()

3、添加多个值：

insert into 表名[(指定的字段)] values(值1),(值2)...

4、快速插入值

insert into 表名[(字段名)] select 字段[,*] from 表名

例如：insert into a(username) select username from a;

insert into a select * from a;（会报错，因为主键是唯一的）

注意：字段名和字段值一一对应

二、SELECT数据的查询

主体结构：

select 字段名/* from 表名

1、查询所有数据（不建议）

select * from 表名

2、指定字段进行查询（建议）

select 字段名 from 表名

3、给字段起别名：

select 字段名 别名 from 表名

关键字：as select 字段名 as 别名 from 表名

三、WHERE条件

1、比较运算符

（1）>

select * from 表名 where id>20

（2）<

select * from 表名 where id < 20

（3）>=

select * from 表名 where id >= 20

（4）<=

select * from 表名 where id <= 20

（5）=

select * from 表名 where id = 20

（6）!= 或 <>

select * from 表名 where id != 20

select * from 表名 where id<>20

2、逻辑运算符（a为表名）

（1）**and**：并且

select * from a where age>18 and age!=32；查询条件为18岁以上但不包含32岁的

（2）**or**：或

select * from a where age<=20 or id >20

（3）**between and**：在....之间

例如1：select * from a where age between 18 and 20

例如2：select * from a where age>=18 and age<=20

（4）**not between and**：不在....之间

例如1：select * from a where age not between 18 and 20

例如2：select * from a where age < 18 or age >20;

（5）**in**：在.....里面

例如1：select * from a where age in(18,20)

例如2：select * from a where age=18 or age=20

3、子查询（条件还是一条sql语句）

select * from a where id in(select id from a where age>18)

4、order by排序（默认是升序）

order by 字段名 asc/desc 升序/降序

例如1：select * from a order by age desc；按照年纪降序排列

例如2：select * from a where id>40 order by id desc；在id>40里面，按照id降序排列

5、is/not is：来查询某个字段是否为null

（1）查询username为null的数据

select * from a where username is null

（2）查询username不为null的数据

select * from a where username is not null

6、limit取值

(1)limit 5：从开头位置取出**5**条数据，等同于：**limit 0,5**

例如：select * from a limit 5；　　select * from a limit 0,5;

(2)条件组合式的查询：

select * from a where age>=30 order by id desc limit 5;查询年龄大于30、id按降序排列，取出id最大的5条数据

7、group by分组

(1) 统计男和女分别有多少人？

select sex,count(*) as total from a group by sex;

(2) 统计每个班级的男和女分别有多少人？

select classid,sex,count(*) as total from a group by classid,sex;

select sex,classid count(*) from a group by sex,classid;

8、having条件：相当于你的where条件

（**1**）按照班级、性别来分组，查询每班人数大于**2**的数据：

select classid,sex,count(*) as total from a group by classid,sex having total>20;

（**2**）按照班级、性别来分组，查询为**python1708**的数据

select classid,sex,count(*) as total from a group by classid,sex having classid='python1708';

（**3**）按照班级、性别分组，查询班级为**python1707**和**python1708**、人数大于**2**的数据

select classid,sex,count(*) as total from a group by classid,sex having classid in ('python1707','python1708') and total>2;

9、模糊查询like

主体结构：

（**1**）**like '%**字符**%'**：包含

例如：select * from a where username like '%三%'

（**2**）**like '%**字符**'**　：以该字符结尾的数据

例如：select * from a where username like '%四'

（**3**）**like '字符%'** ：以该字符开头的数据

例如：select * from a where username like '四%'

10、去除重复值

select distinct age from a；查询去除重复数据后的age字段的值

四、**DELETE**删除

（一）主体结构：

delete from 表名 [where条件]

例如：delete from a where id=2;

（二）清空表的方式：

1、truncate 表名，自增回归原位，从1开始

2、delete from 表名

alter table 表名 auto_increment=1；把自增设置为1

注意：

1、删除时，如果没有where条件，删除所有数据

2、删除后的数据，自增依然记录当前的数据的位置

五、**UPDATE**修改

（一）主体结构：

update 表名 set 字段名1=值1[,字段名2=值2 where条件]

如：update a set username='三四张',sex='m' where id=1;

（二）例子

在所有的年龄的基础上，加2

update a set age=age+2;

注意：

当修改时，没有where条件时，会将所有数据都进行修改

六、聚合函数

1、count()：统计个数

2、max()：最大值

3、min()：最小值

4、sum()：求和

5、avg()：求平均数

七、多表联查

1、隐式内连接查询

select * from goods,user where user.id=goods.uid and uid=1

select user.id,user.username,goods.goodsname from goods,user where user.id=goods.uid and uid=1

2、显式内连接查询：**INNER JOIN**

select * from user INNER JOIN goods on user.id=goods.uid and user.id=2

A表 inner join B表 on 条件

3、左关联**LEFT JOIN**

select * from user LEFT JOIN goods on user.id=goods.uid and user.id=2

注意：

左关联以左表为主表，右表为辅表，会将主表所有数据查询出来，辅表没有关联匹配上的数据用null来占位

4、右关联**RIGHT JOIN**

select * from user RIGHT JOIN goods on user.id=goods.uid and user.id=2

注意：

右关联以右表为主表，左表为辅表，会将主表所有数据查询出来，辅表没有关联匹配上的数据用null来占位

八、以下代码作为了解

1、修改密码：

set password for 用户名@localhost=password('新密码')

2、创建用户

（**1**）选择**mysql**库：

use mysql

（**2**）查看所有用户：

select user from user

（**3**）创建用户

create user zm identified by '123456'；zm代表用户名称

（**4**）授予权限：

grant all on python.* to zm；all代表所有权限，python是数据库名

select,update等权限

（5）回收权限：

revoke all on python.* from zm;

（6）删除用户：

drop user zm;

（7）刷新权限：

flush privileges

python操作MySQL数据库

pip install pymysql ：去cmd界面导入pymsql模块

import pymysql：导入pymysql

1、链接MySQL数据库

conn=pymysql.connect(主机名,用户名,密码,数据库名)

2、设置字符集

conn.set_charset('utf8')

3、创建游标对象，对数据进行增删改查

cursor=conn.cursor()

4、准备SQL语句

5、执行SQL语句

cursor.execute(sql语句)

6、获取查询的结果集

（1）获取所有结果集

cursor.fetchall()

（2）获取一条结果集

cursor.fetchone()

7、获取受影响的行数

cursor.rowcount

8、关闭链接：

```
conn.close()
```