

安装mysql [直接到官网 https://www.mysql.com/](https://www.mysql.com/) 下载安装即可

当我们安装mysql以后在终端执行mysql并没有该指令，是因为还没有将mysql的路径加入环境变量

解决方法：在 ~/.bash_profile 中加入PATH=\$PATH:/usr/local/mysql/bin

苹果->系统偏好设置->最下面点mysql，关闭mysql服务

修改mysql密码（忘记密码的前提下）

cd /usr/local/mysql/bin/

输入 (sudo su) 回车以获取管理员权限
输入 (mysql_safe -skip-grant-tables &) 回车以禁止mysql验证功能，mysql会自动重启，偏好设置中的mysql状态会变成running

ALTER USER 'root'@'localhost' IDENTIFIED BY 'newpassword'; 修改密码

mysql 的连接

本地连接 mysql -u 用户名 -p

远程连接 mysql -h ip -u 用户名 -p

查看当前mysql版本 select version();

查看当前时间 select now();

查看当前所有的数据库 show databases;

修改使用的数据库 use 数据库名

创建一个数据库 create database 数据库名 charset=utf8, 其中charset用于指定数据库的编码格式

删除数据库 drop database 数据库名

查看当前使用的数据库 select database();

MySQL是一个常用的数据库，下面来介绍MySQL的各种命令：

一：

基本命令：

1:

启动MySQL服务命令：

net start 库名, 该命令应该是以管理员身份运行cmd时执行。（沐宇宇）

2:

停止服务命令：

net stop 库名 运行环境同启动服务

3:

连接数据库

mysql -u 用户名 -p

用户名和库名（服务名）是不同的，默认为root

4:

退出登录：

quit、exit

5:

查看当前版本号：

select version();

注意有同一个分号。

6:

查看当前时间

select now();

7:

远程连接：

mysql -h ip -u root -p 本机ip: 192.168.132.1

二：

数据库操作：

1:

创建数据库：

create database 名 charset=utf8;

charset=utf8:表示指定字符集格式为utf-8

2:

删除数据库：

drop database 名;

3:切换数据库：

use 库名

4:

查看当前使用的数据库：

select database();

三：表操作：

1:

查看当前数据库中拥有的表：

show tables;

2:

创建表：

create table 表名 (列及类型) ;

code:

create table man(id int auto_increment primary key,name varchar(20) not null,

age int not null,gender bit default 1);

id: 字段名

int: 该字段的数据类型

auto_increment: 表示该字段自增长，多用于id

primary: 表示该字段为该表的主键（唯一）

not null: 表示该字段不能为空

default: 指明默认值。

3:

删除表：

drop table 表名;

4:

查看表结构：

desc man;

应尽量避免修改表结构，如果不确定表的结构，可以事先定义几个占位表。

5:

查看建表语句：

show create table 表名;

6:

重命名表名：

rename table 原表名 to 新表名

7:

修改表：

alter table 表名 add | change | drop 列名 类型;

code:

alter table newname add isDELETE bit default 0;

四：

数据操作：

1:

插入数据：

a: 全列插入：

insert into 表名 values();

全列插入时数值应与表中的一一对应，对于自增长的数据，应该用0来占位，不能空。

数据的类型一定要对应，如字符串数据输入是应带''。

code:

insert into man values(0,'python',20,1,0);

b:缺省插入：

insert into 表名(列1, 列2, 列3……) values(值1, 值2, 值3……)

即列与数值一一对应，不需要插入值的列可以不列出。

c:同时插入多条数据：

insert into 表名 values(值1, 值2, ……),……

也可以和缺省插入结合插入数据。

2:

删除数据：

delete from 表名 where 条件;

为了保证删除的数据定位准确，应该使用主键作为条件，因为只有唯一一个，

注意：

如果没有条件，将把表中的所有数据都删除。

code:

delete from man where id=2;

3:修改数据：

update 表名 set 列1=值1, 列2=值2…… where 条件

注意：如果没有添加，则将全部相关列修改。

code:

update man set age=19 where id=3;

五：查数据

1:

查看表中的所有数据：

select * from 表名;

其中的 * 代表查询所有的列。

2:

查询指定的列：

既然 * 代表查询所有的列的数据，那么也可以指定需要查询的列。

select 列1, 列2…… from 表名;

code:

select name, age from man;

查询时也可以为相应的列取别名：

select name as a from man;

但是该别名只是显示上的区别，没有具体的意义。

消除重复行：

对于有些相同的数据，当不想全部显示出来时，可以用消除重复行：只需在列之前加distinct即可。

select distinct name, age from man;

3:

条件查询：

格式：

select 列条件 from 表名 where 行条件;

a: 比较运算：

当列当中存在数值时，可以利用数值之间的关系来进行删选。

code:

select * from man where id>4;

表示显示所有列中id>4的行。即显示id>4的行的所有数据。

还可以进行的比较运算：

=、<、<=、>=、!=

b:逻辑运算：

当在进行条件运算的时候还可以加入逻辑运算来进一步筛选。

code:

select * from man where id>3 and gender=1;

c: 模糊查询：

关键字：

like

相当于正则表达式来筛选：

其中：% 表示任意多个任意字符。

_ 表示一个任意字符。

code:

select * from man where name like '黄%';

表示的是查询 man 中以黄开头的所有元素的所有数据。

d: 范围查询：

多用于列中存在数值的表的数据查询，

又分为：

in 不连续范围查询

between……and…… 连续范围查询

code:

select * from man where id in (5,8);

查询id=5和id=8的行的所有列数据。

select * from man where id between 5 and 8;

查询id从5到8的所有数据

e: 空判断查询：

对于允许为空的数据，当没有输入数据时，该成员为空，那么我们就可以通过该标志来查询。

空判断： is null

非空判断： is not null

code:

select * from student where name is null

select * from student where name is not null

f: 查询逻辑的优先级：

小括号，not，比较运算符，逻辑运算符，and的优先级高于or

4: 聚合：

所谓的聚合就是将数据进行聚集，然后进行一系列的运算，从而对数据进行分析。

MySQL提供了五个常用的聚合方法：

a: count(*):

计算总行数，括号中可以是*也可以是列名。

作用：计算总的行数。

b:max(列):

求此列的最大值

c:min (列) :

求此列的最小值。

d: avg (列) :

求此列的平均值

e:sum (列) :

求此列的和

code:

select count(*) from man;

select max(id) from man where gender=0;

附：where的作用：是对数据库中的数据进行筛选。

5: 分组：

所谓的分组，就是根据你指定的分组的依据，将该依据相同的元素放到同一集合中，

分组的主要目的是对相同的数据进行统计。

所以一般结合聚合函数一起使用。

关键词：group by (分组的依据)

语法：

select 列1, 列2……,聚合 from 表名 group by 列1, 列2……;

前一个列1, 列2……表示分组后需要显示的内容。

后一个列1, 列2……表示分组的依据。

code:

select * from man group by age;

说明：根据age为分组依据，即age相同的元素放到同一集合中，

而该集合显示出来的内容为*，即全部。

code:

select gender,count(*) from man group by gender;

说明：根据gender来分组，并且统计每个组拥有的总行数。

分组后再筛选：

关键词：having 筛选条件。

code:

select gender count (*) from man group by gender having gender=0;

说明：先根据gender进行分组，分组后的结果放在一个结果集中，然后再在这个结果集中

筛选出gender=0的数据。

附：

where和having的区别：

筛选的对象不同：where是对数据库进行筛选，having是对结果集进行筛选。

6: 排序：

语法：select * from 表名 order by 列1 asc|desc, 列2 asc|desc ……

说明：order by 后面的列表表示的是排序内容的来源列，asc|desc 表示排序的方式，

*代表的也是结果集中需要显示的列。

code:

select * from man order by age asc,id desc;

select id,name,age from man order by age asc,id desc;

7: 分页查看：

当数据较多时，为了方便查看，通常采用分页的方式来查看。

语法：

select * from 表名 limit start, count;

其中start表示开始的索引数（以0开始），count表示跨距。

code:

select * from man limit 0,4;

六：

表与表之间的关联：

当不同的表与表之间有某种关联关系时，我们希望建立起他们之间的关系，即将他们

关联起来。

比如：表class存放着所有班级的id, name, 人数

表students存放着所有学生的id, name, gender, 所属班级。

那么表students中的学生所属班级是表class中的某个数据。

而我们在创建表students这个表时，就不用具体的班级去表示所属班级，

而可以用class表中的某个与班级name——对应的数据（如id）来替代，即用class中

的id作为students中的外键。

code:

创建class 表：
create table class(id int auto_increment primary key,name varchar(20) not null,stunum int not null);

插入数据：

insert into class values(0,'python',20),(0,'python2',25),(0,'python3',30);

创建students表：

create table students(id int auto_increment primary key,name varchar(20) not null,gender bit default 1, classid int not null,foreign key(classid) references class(id));

插入数据：

insert into students values(0,'lilei',1,1), (0,'tom',1,2), (0,'jack',0,3);

关联查询：

当我们需要查询时，由于学生所处的班级信息在students这个表中是以外键的形式给出的，所以不满足查询的结果，

我们要的结果是显示班级的具体信息，这时候可以通过关联class这个表来将外键替换成关联的内容：

方法：

select students.name,class.name from class inner join students on class.id=students.classid;

其中students.name,class.name 表示结果中要显示的内容，from后的表表示查询的表，inner join后的表表示外键所在的表，

on后面表示外键的对应关系。

MySQL

mysql基本使用