MySQL

创建数据库

create database 名称；

创建数据库并且制定utf8字符集

create database 名称 character set utf8；

显示数据库

show databases；

显示创建数据库信息

show create database 名称；

删除数据库

drop database 名称；

查看服务器中的数据库，并把其中一个库的字符集修改为gb2312

alter database 名称 character set gb2312；

备份数据库

mysqldump -u 用户名 -p 数据库名 > 文件名.sql（window命令）

恢复数据库

Source 文件名.sql（sql语句）

方法一：

为了恢复库。要先创建库create database tt；

再source c:\tt.sql 可以打开一个sql脚本

方法二：

为了恢复库。要先创建库create database tt；

mysql -uroot -proot tt<c:\tt.sql (window命令)

创建表

create table 名称（

field列名 datatype类型，

field列名 datatype类型

）character set 字符集 collate 校队规则；

create table employee

(

Id int,

name varchar(40),

sex bit(1),

brithday date,

Entry\_date date,

job varchar(40),

salary decimal(8,2),

resume text

);

show tables；查看库的表

show create table 表名称；查看这个表的创建细节

desc 表名称；查看表结构

在上面员工表增加一个image列

alter table 表名称 add image blob;

修改job长度为60

alter table 表名称 modify job varchar(60);

删除sex列

alter table 表名称 drop sex；

修改表名称（库名不能改）

rename table 表名称 to 新名称；

修改表的字符集为utf8

alter table表名称 character set utf8;

列名name修改为username

alter table 表名称 change column 列名 新列名 varchar（20）；

删除表

drop table 表名称；

增加数据

insert into table

字符和日期包含在单引号

增加3个员工的信息

insert into 表名称（id，username，birthday）values（1，‘aaa’，‘1980-5-30’）；

select \* from 表名称；

插入数据注意1

如果不跟列名，比如：insert into 表名称 values（1，‘aaa’，‘1980-5-30’）；

则插入数据必须全部完整，不能有空值。

插入数据注意2

insert into 表名称 values（‘1’，‘aaa’，‘1980-5-30’）；

mysql自动把字符串类型转换成int

插入数据注意3

insert into 表名称 values（‘1’，‘张三’，‘1980-5-30’）；

要告诉mysql，采用gb2312

show variables like 'chara%';

set character\_set\_client=gb2312;

要想查看时不乱码

set character\_set\_results=gb2312;

窗口关了，又需要重新写才有效

更新数据

将所有员工的薪水变成5000

update 表名称 set 列名=5000；

将某个人的员工修改为3000

update 表名称 set 列名=3000 where 列名=‘具体名称’；

修改多列的数据

update 表名称 set 列名=4000，列名=‘具体名称’where 列名=‘具体名称；’

将某个人的薪水在原有的基础上增加1000元

update 表名称 set salary=salary+1000 where 列名=‘具体名称’；

更新数据注意问题

先写where条件。避免所有更新

删除数据

delete from

delete只删除记录，不删除表本身。如果要删除表，用drop table

删除所有记录

delete from 表名称；

truncate table 表名称；

delete是一行一行的删除，truncate是摧毁整个表结构

查找数据

查询所有信息

select \* from 表名称；

查询相应的列

select 列名，列名 from 表名称；

过滤表中的重复数据

select distinct 列名 from 表名称；

统计每个学生的总分

select name，（chinese+englis+math） from student；

使用别名

select name 姓名，（chinese+englis+math） 总分 from student；

select name as 姓名，（chinese+englis+math） as 总分 from student；

查询姓名为wu的成绩

select \* from student where name=‘王五’；

查询英语成绩大于90分的同学

select \* from student where english>90;

查询总分大于200分的所有同学

select name from student where （chinese+englis+math）>200；



查询英语分数在80-90之间的同学

select name from student where english>80 and english<90;

select name from student where english between 80 and 90; (包含80,90)

查询数学分数为89,90,91的同学

select \*from student where math in（89,90,91）；

查询所有李姓学生

select \* from student where name like ‘李%’；

select \* from student where name like ‘李\_’；(姓名只有2个字)

查询数学分>80,语文分>80的同学

select \* from student where math>80 and chinese>80;

排序查询order by

asc默认升序 desc 降序

对数学成绩排序后输出

select name，math from student order by math；

降序

select name，math from student order by math desc；

对总分排序后，降序

select name 姓名，（chinese+english+math） 总分 from student order by 总分 desc；

select name 姓名，（chinese+english+math） 总分 from student order by （chinese+english+math） desc；

对姓李的学生成绩排序

select \* from student where name like‘李%’ order by （chinese+english+math） desc；

合计函数count

统计一个班有多少学生？

select count（name） from student；

select count(\*) from student;

统计数学成绩大于90的学生有多少个？

select count（\*）from student where math>90；

统计总分大于250分的人数

select count(\*) from student where (chinese+math+english)>250;

关于count细节

count 不统计null的行

求和sum(仅对数值求和)

统计一个班级数学总成绩?

select sum(math) from student;

统计一个班级语文，英语，数学各科的总成绩

select sum(chinese),sum(english),sum(math) from student;

统计一个班级语文，英语，数学的成绩总和

select sum(chinese+english+math) from student;

统计一个班级语文成绩平均分

select sum(chinese)/count(chinese) as 平均分 from student;

平均函数avg

统计一个班级语文成绩平均分

select avg(chinese) from student;

max/min返回满足where条件的最大值/最小值

求班级最高分和最低分

select max(chinese+english+math),min(chinese+english+math) from student;

group by分组

对订单中商品归类后，显示每一类商品总价

select product，sum(price) from orders group by product;

having和where都可以实现过滤，但在having中可以使用合计函数，通常跟在group by后，作用于组

查询购买了几类商品，并且每类总价大于100的商品

select product from orders group by product having sum(price)>100;

时间日期函数



表

一个表必须有一个主键：不允许为空，不允许重复

删除主键

alter table 表名称 drop primary key;

定义主键自动增长

auto\_increment

id int primary key auto\_increment;

如果删除该id后，再添加，则id在删除的id+1（比如把id=2删除，再添加后，id为3）

假如设置名字不能重复，但是还是输入重复名字报错后，id也会被占用，从而造成id也会增加！！

定义唯一约束

unique

定义非空约束

not null

定义外键约束

constraint 名称\_FK foreign key(名称) references 主表(id)

表的设计

1对多或者多对1的设计，在多的一方，加入外键

constraint 多的一方名称\_FK foreign key(多的一方名称) references 表名称(id)

多对多的表设计，要加入中间表

设置主键 primary key

设置中间表

primarykey(名称1, 名称2),

constraint 名称1\_FK foreign key(名称1) references 表名称(id),

constraint 名称2\_FK foreign key(名称2) references 表名称(id)

1对1关系

主从关系，人可以没有身份证，身份证不能没有人。人是主，身份证是从

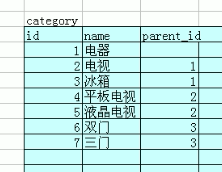
在从的表里面 加入3个约束：外键约束，唯一约束，非空约束

或者直接在从表里面，把从表的主键变成主表的外键

constraint 从表主键名称­\_FK foreign key(从表主键名称) references 主表名称(id)

自连接表

constraint 名称\_FK foreign key(名称) references 自己表名称(id);



create table data1 (

id integer primary key,

outkey vachar(20),

foreign key (outkey) references data2 (area)

)

create table data2(

id integer primary key auto\_increment,

location vachar(20),

area integer(10)

)