# MySQL

SQL分类

DDL:数据库的定义，与数据库/表结构：create,drop,alter

DML:数据操作语言，操作表数据inset update delete

DCL:数据控制语言，设置用户的访问权限，安全

DQL:数据查询语言 select from where

数据库：

创建数据库：create database 数据库名字

删除数据库：drop database 数据库名字

修改数据库：alter database character set 字符集

查询数据库：

show databases 查看所有数据库

show create database 数据库名字：查看数据库定义

select database()：查看当前正在使用的数据库

选中数据库：use 数据库名字

表结构的操作：

创建表：create table 表名（列名 列的类型（长度）约束，列名2，列的类型（长度）约束）；

列的约束：

主键约束：primary key

唯一约束：unique

非空约束：not null

删除表：drop table 表名

添加列：alter table 表名 add 列名 列的类型 列的约束

修改列：alter table 表名 modify 列名 列的类型 列的约束

修改列名：alter table 表名 change 旧列名 新列名 列的类型 列的约束

删除列：alter table 表名 drop 列名

修改表的字符集：alter table 表名 character set 字符集

修改表名：rename table 旧表名 to新表名

查看表：

show table：查看当前数据库中的所有表名

show create table 表名：查看表的定义结构/创建语句

desc 表名：查看表的结构

cmd中操作mysql初始语句： mysql –uroot -proot

列的类型：

int

char/varchar(char:固定长度，varchar：可变长度)

double

float

boolean

data

data: YYYY-MM-DD

time: hh:mm:ss

datatime: YYYY-MM-DD hh:mm:ss 默认值是null

timetamp: YYYY-MM-DD hh:mm:ss 默认使用当前时间

text：主要用来存放文本

blob：存放的是二进制

SQL对表中的CRUD的操作：

插入数据：insert into 表名（列名1，列名2，列名3）values（值1，值2，值3）；

insert into sudent(sid,sname,sex,age)values(1,”zhangsan”,1,23);

简单写法：insert into student values(2，”zhangsan”,1,23);

部分列插入：insert into student(sid,sname)values(3,”lisi”);(省略列名错误)

批量插入：

insert into student values(4,”zhangsan”,1,23),

insert into student values(4,”zhangsan”,1,23),

insert into student values(4,”zhangsan”,1,23);

批量插入效率高，设计关键字少。

查看表中数据：select \* from student；

命令行插入中文问题：

临时解决：set name gbk；命令窗口关闭后再输入就会有问题

解决：修改配置文件my.in

暂停MySQL服务

在安装路径找到my.in配置文件（c:\program files\mysql\mysql serve 5.5）

将文件编码改为GBK（57行）

保存文件退出

启动mysql服务

删除记录delete from 表名whrer[where 条件]

delete from student where sid=10;

delete from student;无条件，会将数据都删除

面试问题：delete 和 truncate 删除数据差别：

delete：DML 一条一条删除表中的数据

truncate：DDL 先删除表再重建表

执行效率问题：

如果数据比较少，delete比较高效率

如果数据比较多，truncate比较高效。

更新表记录 update 表名 set 列名=列的值，列名2=列的值2[where条件]

update student set sname=’王五’ where sid=5;(无条件都改) (字符日期用单引号)

查询记录：

select[distinct] [\*] [列名，列名2] from 表名[where 条件]

查看所有商品：select \* from product；

查询商品名称和价格：select pname,price from product

别名查询 as关键字（as可以省略）

表列名：select p.pname,p.price from product p

列别名：select pname as 商品名称，price as 商品价格 from product(as可略)

去掉重复的值：

select distinct price from product

select 运算查询：仅在查询结果上做+-\*/

select \*，price\*1.5 as 折后价 from product；

条件查询[where关键字]：

select \* from product where price > 60;

where 的条件写法：<>:不等于，标准SQL写法；！=:非标准SQL写法。

like：迷糊查询

-：代表一个字符

%：代表多个字符

select \* from product where pname like ‘\_熊%’;

in 在某个范围中获得值

select \* from product where cno (1,4,5);

排序查询:order by 关键字

asc:ascend升序（默认排序方式）

desc:descend 降序

select \* from product order by price desc;

聚合函数：

sum() avg() count() max() min()

select sum(price) from product;

注意：where条件后面不能接聚合函数

例：查出商品价格大于平均价格的所有商品

select \* from product where price > (select avg(price) from product);

分组：group by

1. 根据cno字段分组，分组后统计商品个数

select cno,count(\*) from product group by cno;

1. 根据cno分组，统计每组商品的平均价格，并且商品平均价格>60

select cno,avg(price) from product group by cno having avg(price) > 60;

having：可以接聚合函数 出现在分组之后；

where：不可接聚合函数，出现在分组之前

编写顺序：select from where group by having order by

执行顺序：from where group by having select order by