数据库

# 一、知识点

数据库相关的概念

MySQL 安装与使用

Navicat使用（重点）

SQL语言-查询（重点）

高级用法（了解即可）

熟练编写查询语句！

数据存储

1.人工管理阶段

结绳记事：通过形状记录当天做过事情

甲骨：通过象形文字记录事情

纸：记录事情

2.文件系统阶段

磁盘：通过数字化文件的方式来记录事情

3.数据库系统阶段

通过数据库这个媒介，先将我们的实物对应成数据用数据库存储起来，然后通过代码的形式展现到网页或者手机应用软件中。

RDBMS

Relational Database Management System

关系 数据库 管理 系统

## 1.关系型数据库的类别

Oracle：银行、电信行业

MySQl：中小型企业（运用最广泛的一种数据库）

Microsoft SQL Server ：（微软的项目）

SQLite：轻量化数据库，移动平台

数据库核心元素

数据行（一条记录）

数据列（字段）

数据表（行和列的一个集合）

数据库（表格的集合）

SQL

Structured Query Language

结构 查询 语言

分类

重点：DQL­­：数据查询语言，用于对数据进行查询，如select

SQL 特点

1.特殊的语言，专门用来查询数据库的

2.不区分大小写

MySQL

商业版：收费，公司使用

社区版：免费

### 1.1MySQL特点：

开源：能看见源代码

免费：用户群体大

支持多平台

MySQL 组成部分

服务端

客户端

举例：我们去饭店吃饭

其中：饭店作为服务提供方，类似服务端

我们通过饭店获取吃饭的服务，类似客户端

Navicat操作

1.新建连接

2.新建数据库

3.新建表

4.添加数据

## 2.数据类型

整数：int 有符号范围（-2147483648 ~2147483647），无符号范围（0 ~ 4294967295）注意：手动设置长度无效

小数：decimal，decimal(5,2)第一位代表一共有几位数字，第二位代表有几位小数

字符串：varchar，范围（0~65533）。注意：需要指定范围：varchar(3)，代表最多三位字符，中文和英文都是一个字符占1位。

**日期时间**: datetime，范围（1000-01-01 00:00:00 ~ 9999-12-31 23:59:59），如'2020-01-01 12:29:59'

约束

主键（primary key）：确定当前数据唯一性的一组特殊字段

注意：

1.默认不能为空（不null）

2.习惯选用 int 类型，要求无符号（不能为负数）

3.为了方便添加数据，需要自增长

非空（not null）：此字段不允许不添加内容。

唯一（unique）：此字段的值不允许重复

外键（foreign key）：维护两个表之间的关联关系

## 数据库的备份与还原

步骤：

### 备份：

1.在要备份的数据库上鼠标右键，选择‘转储SQL文件’

2.下一级选择‘结构与数据’

3.选择备份文件存储位置

4.保存

### 还原：

1.新建一个数据库，字符类型和排列规则要和备份的数据库完全相同

2.新建数据库上鼠标右键，选择’运行SQL文件‘

3.选择备份文件存放位置

4.其他选项都不变，直接运行

# 二、数据表的操作：

## 1.创建表：

格式：

create table 表名(

字段名 类型 约束,

字段名 类型 约束

...

)

单个字段创建：

例：创建学生表，字段要求如下：姓名(长度为10)

create table students(

name varchar(10)

)

多个字段的创建：

例：创建学生表，字段要求如下：姓名(长度为10)， 年龄

create table students(

name varchar(10),

age tinyint unsigned

)

带有主键的创建：

例：创建学生表，字段要求如下：姓名(长度为10)， 年龄，身高(保留小数点2位)

create table students(

id int unsigned primary key auto\_increment,

name varchar(10),

age tinyint unsigned,

height decimal(5,2)

)

## 2.删除表

格式一：drop table 表名

示例：drop table students\_copy

格式二：drop table if exists 表名

示例：drop table if exists students

结合创表语句的用法：（注意：多条语句之间要用英文格式的;号隔开）

drop table if exists students;

create table students(

id int unsigned primary key auto\_increment,

name varchar(10),

age tinyint unsigned,

height decimal(5,2)

)

# 三、数据操作：

## 1.插入数据：

1.1插入单条语句：

格式一：

insert into 表名 values(...)

示例：

insert into students values(null,'赵六',28,187.56)

**注意**：**主键字段需要占位（0，default，null任意一个都可以）**

格式二：部分字段设置值，值的顺序与给出的字段顺序对应

**insert into 表名(字段1,...) values(值1,...)**

示例：

insert into students(name) values('王五')

注意：要在表名后边指定插入的字段

1.2插入多条语句：

方式一：写多条insert语句，语句之间用英文分号隔开

示例：

insert into students values(null,'猪七',28,187.56);

insert into students(name) values('王五1');

insert into students(name) values('王五2')

方式二：一条insert语句

格式一：**insert into 表名 values(...),(...)...**

示例：insert into students values(0,'老夫子1',25,170),(0,'老夫子2',26,190)

格式二：**insert into 表名(列1,...) values(值1,...),(值1,...)...**

示例：insert into students(name) values('张飞1'),('张飞2')

## 2.修改数据：

格式：update 表名 set 列1=值1,列2=值2... where 条件

示例：update students set name='关羽' where id=10

## 3.删除数据：

1.直接删除数据：

格式：delete from 表名 where 条件

示例：delete from students where id=5

注意：如果不跟 where 条件，则会清空表格所有的数据

2.逻辑删除数据：

原理：额外添加一个标记字段，用来标识当前账号的可否使用状态，这种方式称之为逻辑删除

## 4.查询数据：

### 1.查询所有数据：

格式：select \* from 表名

示例：select \* from students

### 2.查询一列数据

格式：select 列1,列2,... from 表名

示例：select name,age,sex from students

### 3.字段起别名：

格式：关键字 as （注意：加中文不需要双引号）

示例：select name as 姓名,age as 年龄,sex as 性别 from students

### 4.消除重复行

格式：select distinct 列1,... from 表名;

示例：select distinct sex from students

### 5.条件where：

格式：select 字段1,字段2... from 表名 where 条件;

示例：select name 姓名,age 年龄,sex 性别 from students where studentNo=006

# 四、select：

## 1.比较运算 =/</>/<>/!=

等于: =

大于: >

大于等于: >=

小于: <

小于等于: <=

**查询不等于: != 或 <>**

示例：

查询小乔的年龄

select name,age from students where studentNo=008

查询20岁以下的学生

select \* from students where age<20

查询家乡不在北京的学生

select \* from students where hometown!= '北京'

select \* from students where hometown<> '北京'

select\* from students where not hometown='北京'

练习：

1、查询学号是'007'的学生的身份证号

2、查询'1班'以外的学生信息

3、查询年龄大于20的学生的姓名和性别

select \* from students where studentNo=007

select \* from students where class!='1班'

select name,sex from students where age>20

## 2.逻辑运算： and/or/not：

示例：

查询年龄小于20的女同学

select \* from students where age<20 **and** sex='女'

查询女学生或'1班'的学生

select \* from students where sex='女' o**r** class='1班'

查询非天津的学生

select \* from students where **not** hometown='天津'

练习：

1、查询河南或河北的学生

2、查询'1班'的'上海'的学生

3、查询非20岁的学生

select \* from students where hometown='河南' or hometown='河北'

select \* from students where class='1班' and hometown='上海'

select \* from students where not age=20

## 3.模糊查询：like

关键字：**like**，**%**代表匹配任意数量的字符，**\_**匹配一个字符

示例：

查询姓孙的学生

select \* from students where name like '孙%'

查询姓孙且名字是一个字的学生

select \* from students where name like '孙\_'

查询叫乔的学生

select \* from students where name like '%乔'

查询姓名含白的学生

select \* from students where name like '%白%'

练习：

1、查询姓名为两个字的学生

2、查询姓百且年龄大于20的学生

3、查询学号以1结尾的学生

select \* from students where name like '\_\_'

select \* from students where name like '百%' and age>20

select \* from students where studentNo like '%1'

——————————————————————————————————————

## 4.范围查询： in / between and

示例：

查询家乡是北京或上海或广东的学生

select \* from students where hometown **in** ('北京','上海','广东')

查询年龄为18至20的学生

select \* from students where age **between** 18 **and** 20

练习：

1、查询年龄在18或19或22的女生

2、查询年龄在20到25以外的学生

select \* from students where age in(18,19,22) and sex='女'

select \* from students where age not between 20 and 25

## 5.判断空：is null / is not null

示例：

查询没有填写身份证的学生

select \* from students where card **is null**

查询填写了身份证的学生

select \* from students where card **is not null**

## 6.排序： order by (asc desc)

注意：asc 升序，desc 降序，排序默认为升序

格式：select \* from 表名 order by 列1 asc|desc,列2 asc|desc,...

查询所有学生信息，按年龄从大到小排序，年龄相同时，再按学号从小到大排序

select \* from students order by age desc, studentNo asc

练习：

查询所有学生信息，按班级从小到大排序，班级相同时，再按学号再按学号从小到大排序

select \* from students order by class,studentNo

## 7.聚合函数：

### 1.统计个数 count()

查询学生总数

select count(\*) from students

### 2.最大值 max()

查询女生的最大年龄

select max(age) from students where sex='女'

### 3.最小值 min()

查询1班的最小年龄

select min(age) from students where class='1班'

### 4.求和 sum()

查询北京学生的年龄总和

select sum(age) from students where hometown='北京'

### 5.平均值 avg()

查询女生的平均年龄

select avg(age) from students where sex='女'

练习：

1、查询所有学生的最大年龄、最小年龄、平均年龄

2、一班共有多少个学生

3、查询3班年龄小于18岁的同学有几个

select max(age),min(age),avg(age) from students

select count(\*) from students where class='1班'

select count(\*) from students where class='3班' and age<18

## 8.分组及分组后再次筛选having：

格式：select 列1,列2,聚合... from 表名 group by 列1,列2...

示例：

查询各种性别的人数

select sex,count(\*) from students group by sex

查询各种年龄的人数

select age,count(\*) from students group by age

练习：

1.查询各个班级学生的平均年龄、最大年龄、最小年龄

select class,avg(age),max(age),min(age) from students group by class

分组后再次筛选：

**关键字：having**

示例：

格式：select 列1,列2,聚合... from 表名

group by 列1,列2,列3...

having 列1,...聚合...

示例：

查询男生总人数

select count(\*) from students where sex='男'

select sex,count(\*) from students group by sex having sex=“男”

**★对比where与having**

where是对from后面指定的表进行数据筛选，属于对原始数据的筛选

having是对group by的结果进行筛选

## 9.获取部分行：limit

格式：select \* from 表名 limit start,count

注意：start 代表开始位置，从0开始。count 代表要获取的行数。

示例：

查询前3行学生信息

select \* from students limit 0,3

查询第4到第6行学生信息

select \* from students limit 3,3

**分页实现：**

select \* from students limit 0,4

select \* from students limit 4,4

select \* from students limit 8,4

# 五、连接查询(多练习加深理解)

## 1.等值查询: inner join--- on

概念：等值连接查询：查询的结果为两个表匹配到的数据

格式：select \* from 表1,表2 where 表1.列=表2.列

内连接：

格式：select \* from 表1 inner join 表2 on 表1.列=表2.列

注意：一般情况下，等值连接=内连接。当等值连接或者内连接时，表的顺序对查询结果没有影响。

## 2.左连接：left join --- on

格式：select \* from 表1 left join 表2 on 表1.列=表2.列

注意：left join 关键字之前（左边）的表要求显示全部内容，关键字后边的表根据 on 后面的对应字段填充数据，没有的部分，用 null 占位。

例题：查询所有学生的成绩，包括没有成绩的学生

select \* from students stu

left join scores sc on stu.studentNo=sc.studentno

## 3.右连接：right join--- on

格式：select \* from 表1 right join 表2 on 表1.列=表2.列

注意：right join 关键字之后（右边）的表要求显示全部内容，关键字前边边的表根据 on 后面的对应字段填充数据，没有的部分，用 null 占位。

▲：左右连接在某些情况下，可以实现相同的查询结果！左右连接的选择需要根据查询需求灵活选用。

# 六、自关联：（上下级从属关系）

条件：1.只有一张表

2.表内所有对象的关系是上下级的层次关系。（例如：省市区）

# 七、子查询：

## 1.概念：

在一个 select 语句中,嵌入了另外一个 select 语句, 那么被嵌入的 select 语句称之为子查询语句

## 2.分类：

标量子查询：子查询返回的结果是一个数据(一行一列)

select \* from scores where studentno=(select studentNo from students where name='王昭君')

列级子查询：返回的结果是一列(一列多行)

示例：select \* from scores where studentno in(select studentNo from students where age=18)

行级子查询：返回的结果是一行(一行多列)

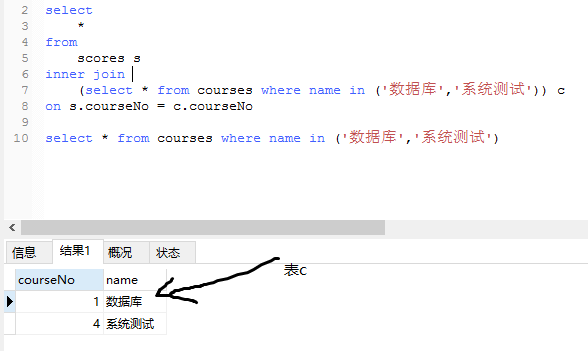
示例：

select \* from students where (sex,age)=(select sex,age from students where sex='男' and age=(select max(age) from students))

表级子查询: 返回的结果是多行多列

示例：

select \* from (select \* from courses where name in ('数据库','系统测试')) as c inner join scores sc on sc.courseNo = c.courseNo



注意：

1.标量子查询和行级子查询和列级子查询，一般都放到 where 后边，充当条件。

2.表级子查询：一般放到 from 后边，充当数据源（表）。

## 3.子查询中使用的特殊关键字：

in等同于=any等同于=some

<any或<some：小于其后的最大值

>any或>some: 大于其后的最小值

=all：没有什么实际意义

<>all：相当于不取其后的任意值

课堂笔记

# 八、数据库的相关操作

## 1.命令行连接数据库：

命令格式：mysql -u用户名 -p密码

示例:mysql -uroot -p123456

## 2.数据库操作：（注意：命令行操作，一定要用；结尾）

1.查看当前连接下的所有数据库：

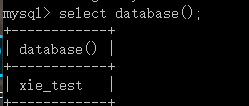
show databases；

2.选择要操作的数据库：

use 数据库的名字

3.查看当前数据库的名称：

select database();



常规操作：

1.创建数据库：

格式:create database 数据库名字 charset=utf8;

2.删除数据库：

格式:drop database 要删除的数据库名字;

## 3.数据表操作：

1.查看当前进入的数据库中的表

**show tables;**

2.查看某个表的结构

**desc 表名;**

3.查看创表语句

show create table 表名

4.其他正常的数据查询语句均能在命令行中执行

注意:

1.命令输入完需要以;结尾,否则命令会不生效.

2.如果中文内容显示乱码可以执行:**charset gbk**; set names gbk;

## 4.数据库备份：

注意：

1.要用管理员身份打开命令行

2.当前路径下能够执行mysqldump.exe

步骤：

备份：

1.mysqldump -uroot -p 要备份的数据库名字 > 备份后的文件名字.sql

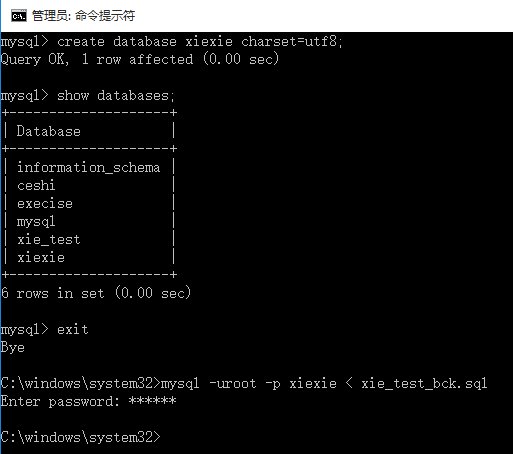
2.输入数据库密码执行备份

恢复：

1.新建一个新的数据库：create database 数据库名字 charset=utf8;

2.执行 mysql -uroot -p 新数据库的名字 < 备份后的文件名字.sql

3.输入数据库密码执行

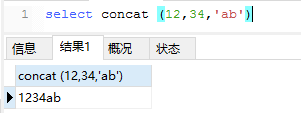


# 九、函数

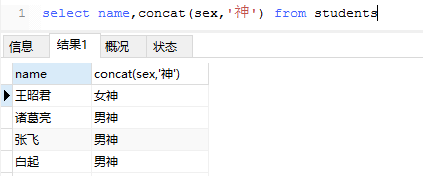
## 1.字符串函数：

### 1.拼接字符串:concat(str1,str2...)

1.select concat(12,34,'ab')



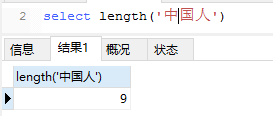
2.select name,concat(sex,'神') from students



### 2.包含字符个数:length(str)

1.select length('abc')

2.select length('中国人') 在navicat 中文字符为3 在命令行里为2



### 3.截取字符串

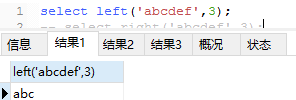
1.left(str,len)返回字符串str的左端len个字符

2.right(str,len)返回字符串str的右端len个字符

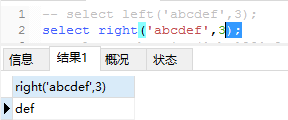
3.substring(str,pos,len)返回字符串str的位置pos起len个字符

示例：

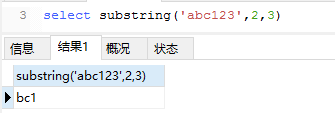
1. select left('abcdef',3) 从左边开始截取3个字符



2.select right('abcdef',3) 从右边开始截取3个字符



3.select substring('abc123',2,3); 从第二个数字开始截取，截取3个字符



### 4.去除空格 trim [trɪm]

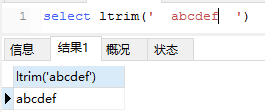
1.ltrim(str)返回删除了左空格的字符串str

2.rtrim(str)返回删除了右空格的字符串str

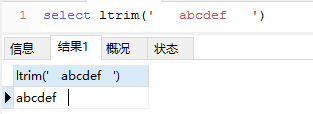
3.trim(str）同时去除两边空格 ★trim 对制表符不起作用

示例：

1. select ltrim(' abcdef ')



1. select rtrim(' abcdef ')

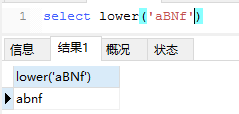


3.select trim(' abcdef ')

### 5.大小写转换，①lower(str) ②upper(str)

示例：

1. select lower('aBCd') 转换成小写



2.select upper('aBCd') 转换成大写

## 2.数字函数:

### 1.求四舍五入值round(n,d)

n表示原数，d表示小数位置，默认为0

示例:

1. select round(1.6) 默认为0



1. select round(1.6,1)



### 2.求x的y次幂pow(x,y)

示例: select pow(2,3)

### 3.获取圆周率PI()

示例: select PI();

### 4.随机数rand()，值为0-1.0的浮点数

示例: select rand()

**★随机取一条数据的语句:select \* from students order by rand() limit 1**

## 3.日期时间函数:

1.当前日期current\_date()

select current\_date();

2.当前时间current\_time()

select current\_time();

3.当前日期时间now()

select now();

日期时间格式化:

示例:

1.select date\_format('2016-12-21','%Y %m %d')

2.select date\_format(now(),'%Y %m %d')

# 十、流程控制:

示例1:

select

case 1

when 1 then 'one'

when 2 then 'two'

else 'zero'

end as result;

示例2:

select

name,sex,

case sex

when '女' then concat(left(name,1),'美铝')

when '男' then concat(left(name,1),'帅锅')

else concat(left(name,1),'某某')

end as result

from students

# 十一、自定义函数:

## 语法如下:

delimiter $$

create function 函数名称(参数列表) returns 返回类型

begin

SQL语句

end

$$

delimiter ;

注意:delimiter的作用是将;的作用转义掉.

示例:

## 创建:

1：设置分割符

delimiter $$

2：创建函数

create function my\_trim(str varchar(100)) returns varchar(100)

begin

return ltrim(rtrim(str));

end

$$

3：还原分割符

delimiter ;

调用:

select ' abc ',my\_trim(' abc ')

## 存储过程:

语法如下:

delimiter //

create procedure 存储过程名称(参数列表)

begin

sql语句

end

//

delimiter ;

注意:delimiter的作用是将;的作用转义掉.

示例:

创建:

1：设置分割符

delimiter //

2：创建存储过程

create procedure proc\_stu()

begin

select \* from students;

end

//

3：还原分割符

delimiter ;

调用:

call proc\_stu();

自定义函数及存储过程的**作用:**

1.存储过程和函数都是为了可重复的执行操作数据库的 SQL 语句的集合

2.存储过程和函数都是一次编译,就会被缓存起来,下次使用就直接命中缓存中已经编译好的 SQL, 不需要重复编译

3.减少网络交互,减少网络访问流量

# 十二视图

**创建:**

格式:create view 视图名称 as select语句

示例:

create view v\_stu\_score\_course as

select

stu.\*,cs.courseNo,cs.name courseName,sc.score

from

students stu

inner join scores sc on stu.studentNo = sc.studentNo

inner join courses cs on cs.courseNo = sc.courseNo

注意:在图形化页面查看视图,有专门的位置存放视图,如果在命令行里查看,则跟数据表一起显示.

**删除:**

格式:drop view 视图名称

示例:drop view v\_stu\_score\_course

调用:视图的用途就是查询

示例:

select \* from v\_stu\_score\_course;

# 十三、事务:

作用:通过事务的特性来确保数据操作的正确性.

命令:

开启事务: begin 或 start transaction;

提交修改: commit

回滚修改: rollback

# 十四、索引:

是一种优化数据库查找数据操作速度的一种方案.

查看索引:

show index from 表名

创建索引:

方式一：建表时创建索引

示例:

create table create\_index(

id int primary key,

name varchar(10) unique,

age int,

key (age)

);

方式二：对于已经存在的表，添加索引

格式:create index 索引名称 on 表名(字段名称(长度))

示例:

create index age\_index on create\_index(age);

create index name\_index on create\_index(name(10));

删除索引：

格式:drop index 索引名称 on 表名;

# 十五、外键:

查看外键:show create table 表名

创建外键:

方式一：创建数据表的时候设置外键约束

注意:两张表的对应字段的类型也要对应起来,否则创建外键会失败!!!

格式:

foreign key(自己的字段) references 主表(主表字段)

示例:

create table goods\_fk(

id int unsigned primary key auto\_increment,

name varchar(150),

cate\_id int unsigned,

brand\_id int unsigned,

price decimal(10,3) default 0,

is\_show bit default 1,

is\_saleoff bit default 0,

foreign key(cate\_id) references goods\_cates(cate\_id),

foreign key(brand\_id) references goods\_brands(brand\_id)

);

方式二：对于已经存在的数据表设置外键约束

注意:两张表的对应字段的类型也要对应起来,否则创建外键会失败!!!

格式:

alter table 从表名 add foreign key (从表字段) references 主表(主表字段);

示例:

alter table goods add foreign key (cate\_id) references goods\_cates(cate\_id);

alter table goods add foreign key (brand\_id) references goods\_brands(brand\_id);

删除外键:

1.需要先获取外键约束名称

格式:show create table 表名

2.获取名称之后就可以根据名称来删除外键约束

格式:alter table goods drop foreign key 外键名称

示例:

alter table goods drop foreign key goods\_ibfk\_1;

alter table goods drop foreign key goods\_ibfk\_2;

# 十六、修改密码:

## 1.修改已知密码:

1.连接数据库

2.使用 mysql 数据库

use mysql;

3.使用更新数据语句修改密码

update user set password=password('新密码') where user='用户名';

示例：

update user set password=password('123') where user='root';

4.刷新权限(使新密码生效)：

flush privileges;

注意: password()为数据库提供的密码明文加密函数.

## 2.修改遗忘的密码:

1.查找配置文件:

Centos中：配置文件位置为/data/server/mysql/my.cnf

Windows中：配置文件位置为C:\Program Files (x86)\MySQL\MySQL Server 5.1\my.ini

2.搜索找到mysqld，在它的下一行，添加skip-grant-tables

[mysqld]

skip-grant-tables

3.重启 mysql 服务使配置文件生效

4.重启服务后,实现免密码登录

5.用已知密码的修改方式修改密码,记得刷新权限

6.重新修改配置文件,重启服务,使免密登录设置失效

7.修改完成!