www.codeboy.com/dz/

# 连接数据库

mysql.exe -h127.0.0.1 -P3306 -uroot -p

地址 端口 用户名

退出 quit；

简写

mysql -uroot

use； 后面写上数据库名称进入指定的数据库

show databases; 查看当前数据库服务器下的数据库(查看所有数据库)

show tables；显示当前数据库所有的数据表

desc； 描述表中有哪些列

row 行

coluwn 列

MySQL中常用SQL命令：

SQL：Structyred Query Language,结构化查询语言，用于操作关系型数据库服务器的数据----增删改查

SQL最早是由IBM提出，后提交给ISO最终成为了数据库标准语言。版本分为：SQL-87 SQL-92 SQL-99等。他被绝大多数的关系型数据库所支持。

（1） 交互模式：客户端输入一行，点击回车，服务器执行一行。适用于临时查看数据。

（2） 脚本模式：客户端把多行要执行的命令写在一个文本文件中，一次性提交给服务器执行，适用于批量的增删改查。

mysql -uroot < c:\xampp 路径

# SQL语法规范：

（1）SQL 命令可以使用单行注释：#...或者多行注释/\*...\*/

(2)若第N行语句出现语法错误，则此语句以及后续的所有语句都不在执行。

（3）SQL命令不区分大小写，习惯上数据库关键字都用大小写。非关键字都用小写

（4）每条SQL语句及英文的分号作为结尾，一条语句可以跨越多行，见到分号结束。use 命令除外。

# 日常开发使用的sql命令

（1）丢弃知道的数据库，如果存在的话

DROP DATABASE IF EXISTS jd;

（2）创建新的数据库

CREATE DATABASE jd;

（3）进入该数据库；

USE jd;

（4）创建数据表；

CREATE TABLE Student(

列1 数据类型 是否为空 是否为主键 是否自增长，

sid int not null primary key)

(5)更改表数据;

UPDATE 表名称 SET 列名='',列名2='',... where 列名='';

(6)删除表数据;

DELETE FROM 表名称 WHERE 列名='';

数据库中文乱码

计算机储存英文字符:

ASCII:一共128个,对所有的英文字符已经标点符号进行了编码

Latin-1:一共有256个,兼容ASCII,同时对欧洲的符号进行了编码.

计算机如何储存中文:

GB2312:对6000多常用简体中文进行编码,并同时兼容ASCII.

GBK: 对两万多个汉字进行了编码,涵盖了一些繁体中文,兼容GB2312.

BIG5: 台湾繁体字编码,兼容ASCII,不兼容GB(国标)系列.

Unicode:对世界上主流国家的语言进行了编码,兼容ASCII,不兼容GB,BIG5系列.

具体分为三种储存方案:

UTF-8,UTF-16,UTF-32等

editplus 默认编码设置

tool->profence->file->default coding

1).SQL脚本另存为的编码 UTF-8

2).客户端连接服务器使用的编码 SET NAMES UTF8

3).数据库储存数据使用的编码 CHARSET=UTF8;

# 列类型

1),数值类型(可以不加引号) 一个字节 2进制4位数(有正负符号,实际位数-1)

tinyint 微整型 占用一个字节 范围 -128~127

smallint 小整型 占用两个字节 范围 -32768~32767

int 整型 占用四个字节 范围 -2147483648~214748367

bigint 大整型 占用八个字节 范围太大,懒得算

TB->GB->MB->KB->BYTE->BIT 1BYTE->8BIT

2),浮点型

float 单精度浮点型,占4个字节 范围 3.4E38,比int大得多

double 双精度浮点型,占8个字节

decimal 定点小数(M,D)定点小数,M代表总的有效位数

D代表小数点后允许的有效位数,不存在误差

bool 布尔型,只能取值为 TRUE/1,FALSE/0

在正在存储的时候,会存储1和0,不能加引号

3),日期时间类型

DATE 日期类型 '2018-10-1'

项目中储存日期使用距离计算机元年(1970-1-1)的毫秒数,使用BIGINT来存储.

1000毫秒 = 1秒

计算2018-10-8 距离计算机元年有多少毫秒

48\*365\*24\*60\*60\*1000 毫秒

time 时间类型 '14:16:23'

datetime日期时间类型 2018-10-1 14:16:23'

4),字符串类型

varchar(M) 变长字符串,M代表长度.不能确定长度时使用.

未填充完,用\0填充,一个就行.

char(M) 定长字符串,比如储存手机号(11位),身份证号(18位)等

如果未填充完 用\0填充(代表空格,缺几个填几个)

text(M) 大型变长字符串,M不超过2G.

char(M) 可能产生空间浪费,但是操作速度比varchar(M)快. M长度不超过255.

varchar(M) 不产生空间浪费,操作速度比CHAR(M)慢,M长度不超过65535.

# 列约束:

MSQ可以对插入表中的数据进行特定的验证,只有满足条件的数据才能插入成功,否则插入失败.

例如:员工编号应该是惟一的,不能重复;年龄大于0,小于100岁;一个人的政治面貌群众.党员.

团员.

主键 primary key声明为主键的列上不能插入重复的值,设置了主键,查找结果会按照

的由小到大排序---加快查找速度--一个表最多只能有一个主键.如果插入

的值为NULL,默认转为0.

唯一约束 unique 唯一约束,声明了唯一约束的列,不能插入重复的值,但允许插入一个或者

多个NULL.

null是一个比较特殊的值,可以多次重复使用.

非空约束 not null 声明了非空约束的上不能插入NULL,如果插入的是NULL,默认转成''.

对比primary key 和 unique not null(唯一且不能为空)

区别在于:前者会自动排序,后者不会;

前者在一个表中只能出现一次,后者在一个表中可出现多次.

默认值约束 DEFAULT 可以使用DEFAULT 关键字为列声明默认值,有两种方法可以应用到默认值

insert into 表名 values (列值,列值,DEFAULT)

insert into 表名(列名,列名) values(列值,列值,DEFAULT)

检查约束 CHECK

检查约束可以对要插入的数据进行检验

CREATE DATABASE Student(age TINYINT CHECK(age>=18 and age<=60)) –Mysql 不支持 会降低数据的插入速度.一般检查约束会在浏览器端进行

# MYSQL 标准SQL语句的分类

(1),DDL: DATA Define Language 定义语句语言

CREATE/DROP/ALTER

(2),DML: Data Munipulate Language 操作数据

INSERT/UPDATE/DELETE

(3),DQL: Data Query Language 查询数据

SELECT

(4),DCL: Data Control Language 控制用户权限

GRANT(授权)/REVOKE(收回权限)

### 简单查询 - 重点

1,简单查询 只查询特定的列

示例:查询所有员工的姓名,工资,生日

Select ename,salary,birthday from emp;

联系:查询所有员工的编号,姓名,性别,生日,工资

select eid,ename,sex,birthday,salary from emp;

2,简单查询—查询所有的列

Select \*from emp;

\*代表所有的列

3,给列取别名

示例:查询所有员工的姓名和工资,使用汉字的别名显示

Select ename 姓名,salary 工资 from emp;

练习:查询所有员工的编号,姓名,性别,所在部门,生日 给列起中文别名

Select **eid 员工编号,ename 姓名, sex 性别, deptid 所在部门, birthday 生日**

from emp;

4,简单查询 只显示不同的记录/合并相同的记录

示例:查询员工都在哪些部门

Select distinct deptid 部门名称 from emp;

练习:查看工资都有哪些档位

Select **distinct** salary 工资,from emp;

5简单查询 在查询时执行计算

示例:2\*3/4\*7/0.5

Select 2\*3/4\*7/0.5;

练习:查询出所有员工的姓名和年薪

Select ename 员工姓名 ,salary \* 12 年薪 from emp;

练习:假设每个员工的工资增加500块,年终奖是5000块,现在计算年薪是多少

Select ename 姓名,(salary+500) \*12+5000 年薪 from emp;

6.对查询结果进行排序

示例:查询所有员工信息,按照工资升序(降序)排列

Select \* from emp order by salary asc/desc;

练习:查询员工姓名,性别信息,按照工资高低进行排列

Select ename 姓名,sex 性别,salary 工资 from emp order by salary desc;

练习:查询所有的员工信息,把结果按照年龄由大到小排序

Select \* from emp order by birthday;

练习:查询所有的员工信息,结果按照姓名升序排序;

Select \* from emp order by ename;

练习:查询所有的员工信息,结果按照工资的降序排列,如果工资相同,按照姓名升序排列

Select \* from emp order by salary desc,ename asc;

练习:查询所有员工信息,女员工在男员工前边,如果性别相同,按照年龄降序排列

Select \* from emp order by sex asc,birthday asc;

7简单查询 条件查询

示例: 查询员工编号为5的员工所有信息

Select \* from emp where eid =5;

练习:查询出姓名为KING的员工的编号.工资,生日

Select eid 员工编号,salary 工资,birthday 生日 from emp where ename= ‘king’;

练习:查询出20号部门下所有员工的信息

Select \* from emp where deptid=20;

练习:查询出所有的男员工信息

Select \* from emp where sex = 1;

示例:查询出工资大于等于5000的员工所有信息

Select \* from emp where salary >= 5000;

比较运算符: > < >= <= !=

练习:查询出1991-1-1之后出生的员工所有信息

Select \* from emp where birthday > ‘1991-1-1’;

练习:查询出不在10号部门的所有员工信息

Select \* from emp where deptid != 10;

练习:查询暂时没有部门的员工信息

Select \* from emp where deptid is null;

练习:查询出有明确部门的员工信息

Select \* from emp where deptid is not null;

注意: NULL和任何值不能进行相等/不相等的判定,只能使用 is null/is not null.

练习:查询出工资大于6000的女员工所有信息

Select \* from emp where salary > 6000 and sex = 0;

练习:查询出工资在5000-5999之间的员工信息

Select \* from emp where salary>=5000 and salary<6000;

Select \* from emp where salary between 5000 and 5999;

练习:查询出工资小于4000和大于8000的员工信息

Select \* from emp where salary not between 4000 and 8000;

Select \* from emp where salary<4000 or salary>8000;

练习:查询出在1991年出生的员工信息

Select \* from emp where birthday like ‘1991-%-%’;

Select \* from emp where birthday between ‘1991-1-1’and ‘1991-12-31’;

Select \* from emp where birthday>'1991-1-1' and birthday<'1991-12-31';

练习:查询出在1990年之前和1993年之后出生的员工信息

Select \* from emp where birthday<'1990-1-1' or birthday>'1993-12-31';

Select \* from emp where birthday not between '1990-1-1' and '1993-12-31';

练习:查询出在10和30号部门的员工信息

Select \* from emp where deptid in (10,30);

Select \* from emp where deptid = 10 or deptid=30;

练习:查询出不在10和30号部门的员工信息;

Select \* from emp where deptid not in (10,30);

Select \* from emp where deptid!=10 and deptid !=30;

IS NULL / IS NOT NULL

AND / OR

BETWEEN..AND… / NOT BETWEEN… AND …

IN() / NOT IN()

简单查询 模糊查询

查询出所有姓名里边还有E的员工所有信息

Select \* from emp where ename like '%e%';

查询出所有员工中姓名以E结尾的员工信息

Select \* from emp where ename like '%e';

查询出姓名中倒数第二个字符是E的员工信息

Select \* from emp where ename like '%e\_';

查询出姓名中第二个字母是E的员工信息

Select \* from emp where ename like '\_e%';

注意:以上两个匹配符不能和=使用,必须使用LIKE关键字

分页查询

1,简单查询 分页查询

Select \* from emp limit start,count;

Start:开始的值

Count: 每页的大小

示例:查询第一页,每页显示5条记录

Select \* from emp limit 0,5;

练习:假设每页显示5条记录,查询第二页,第三页

Select \* from emp limit 5,5;

Select \* from emp limit 10,5;

第几页从多少开始公式=(页面-1) \* 每页的大小数值(count)

Select \* from emp limit 10,10;

练习:假设每页显示7条记录,查询前三页

Select \* from emp limit 0,7;

Select \* from emp limit 7,7;

Select \* from emp limit 14,7;

注意:LIMIT后的两个数值不能添加引号

# 复杂查询

## 1),聚合查询

聚合查询 / 分组查询

示例:查询所有员工的数量

Select count(\*) from emp;

Select count(eid) from emp;

函数:功能体,接受若干个数据,最终导出特定的计算结果 比如:饺子机制造饺子的过程.

COUNT() 数量统计

SUM()总和

AVG()平均数

MAX()最大值

MIn()最小值

通过部门编号计算员工数量

Select count(deptid) from emp;

查询出所有员工的工资总和

Select sum(salary) from emp;

查询出男员工的平均工资是多少

Select avg(salary) from emp where sex =1;

Select sum(salary) /count(\*) from emp where sex =1;

查询出编号为10的部门的最高工资和最低工资的人员信息

Select max(salary) , min(salary) from emp where deptid =10;

##### 分组查询

查询出每个部门的员工数量

Select deptid 所属部门,count(\*) 员工数量 from emp group by deptid;

分组查询注意: 分组查询中 SELECT 后只能接分组的条件和聚合函数,不能直接显示其他的列

练习:查询出每个部门的平均工资,最大工资,最小工资

Select deptid 部门名称, avg(salary) 平均工资, max(salary) 最高工资, min(salary) 最低工资 from emp group by deptid;

练习:查询出男员工和女员工的人数分别是多少

Select sex 性别, count(\*) 人数统计 from emp group by sex;

## 2),子查询 多个SQL语句的综合

示例:查询出’研发部’所有员工的信息

步骤1,查询出研发部的部门编号 – 10

Select did from dept where dname = ‘研发部’;

步骤2查询出部门编号为10的员工信息

Select \* from emp where deptid = 10;

综合:

Select \* from emp where deptid = (Select did from dept where dname =’研发部’);

练习:比TOM工资高的员工信息(升降序排列)

Select \* from emp where salary>( Select salary from emp where ename = ‘tom’);

# order by salary asc/desc;

练习:查询低于平均工资的所有员工信息

Select \* from emp where salary<(select avg(salary) from emp);

练习:查询出和TOM同一年出生的员工所有信息

Select \* from emp where YEAR(birthday)=( Select YEAR( birthday) from emp where ename = ‘tom’);

YEAR() 提取日期中的年份

MONTH() 提取日期的月份

## 3),多表查询/跨表查询

示例:查询出所有的员工及其所在部门名称

Select ename 姓名, dname 部门名称 from emp,dept

Where emp.deptid = dept.did;

上述语法是SQL-92中的多表查询语法,存在天然缺陷, 这样查询不能查出没有部门的员工,没有员工的部门

在SQL-99多变查询中出现了新的查询语法

#### 内连接 INNER JOIN…ON—和SQL-92效果一样

Select ename, dname from emp INNER JOIN dept on deptid=did;

左外连接 LEFT JOIN …. ON –显示左侧所有记录, 即使在右侧没有对应的记录

Select ename ,dname from emp left join dept on deptid=did;

即使在右侧没有对应的记录,OUTER关键字可以省略

#### 右外连接 right outer join.. on –显示右侧所有记录,即使在左侧没有的记录

Select ename,dname from emp right join dept on deptid=did;

**全连接 – FULL JOIN 显示左侧表,右侧表所有记录**

**MYSQL不支持全连接**

**MYSQL中使用UNION/UNION ALL 来实现全连接**

**UNION 合并相同的记录,**

**UNION ALL 不合并相同的记录**

**Select ename from emp\_chn**

**UNION**

**Select ename from emp\_us**

**使用UNION 和UNION ALL 实现全连接**

**(Select ename,dname from emp left join dept on deptid=did)**

**union**

**(Select ename,dname from emp right join dept on deptid=did);**