用户root

密码Wang

DB：数据库

DBMS：数据库管理系统

SQL：结构化查询语言

数据库存储数据特点：

1. 数据存到表中，表存到库中
2. 一个库中可以有多张表，每张表具有唯一表名
3. 表中有一个或多个列，列又称为字段

**命令语句**

启动：net start mysql

关闭：net stop mysql

登录：mysql -u root -p密码

退出：quit exit

**常用SQL语句** #sql语句最后要加分号；结尾

show databases; #展示所有数据库

use 库名; #进入对应数据库

show tables; #展示库中所有表单

show tables from 库名; #展示对应库中的表单

select database(); #查看自己目前所在哪个数据库中

create table 表名(

列名 类型

列名 类型) ; #创建表，char（int）类型固定空间，varchar（int）类型可变空间

desc 表名; #查看表的结构

select \* from 表名; #查看表中数据

insert into 表名(列名 , 列名) value(数据，数据); #向表中插入数据

update 表名 set name=数据 where id=数据; #更新表中数据

delete from 表名 where id=数据; #删除表中数据

select version(); #查看数据库版本

**SQL语法规范**

1. 不区分大小写，但建议关键字大写，表名、列名小写
2. 每条命令最好用分号；结尾。用\g结尾也行
3. 每条命令根据需要可以进行缩进或换行
4. select from where group by having order by 关键字顺序
5. 注释

单行注释：#注释文字

单行注释：-- 注释文字

多行注释：/\*注释文字\*/

**SQL语句**

DQL（data query language）数据查询语言 主要是select

基础查询语法

select 查询列表 from 表名；

查询列表可以是：表中字段，常量值，表达式，函数

查询的结果是一个虚拟的表

查询表中单个字段：select 字段 from 表名

1.查询表中单个字段

select 字段 from 表名；

2.查询表中多个字段

select 字段，字段，字段 from 表名；英文逗号，隔开

3.查询表中所有字段

select \* from 表名；\*代表所有

4.查询常量值

select 100；

select ‘wang’；

5.查询表达式

select 100\*98；

6.查询函数

select version（）；

7.起别名：

#便于理解

#如果要查询的字段有重名的情况，使用别名可以区分开来

方式一：用as

select 100\*98 as 别名；

方式二：用空格

select 100\*98 别名；

案例：查询salary，显示结果为out put

select salary as ‘out put’ from 表名；

#as可能会和后面的词组成关键字，需用引号区分开

8.去重：distinct关键字

select distinct 列名 from 表名；

9.+号的唯一作用：运算符

#两个操作数都为数值型，则做加法运算

#若其中一方为字符型，mysql试图将字符型转换成数值型，成功则做加法运算，若转换失败，则将字符型数值转换成0做加法运算

#只要其中一方为null，则结果必为null

10.拼接函数concat（）

select concat（last\_name,first\_name）as 姓名 from 表名；

11.判断是否为空函数ifnull（）

select ifnull（待判断列名，返回参数） as 别名 from 表名；

条件查询：where关键字

select 查询列表 from 表名 where 筛选条件；

1.按条件表达式筛选

条件运算符：> < = >= <= 不等：!=和<>

2.按逻辑表达式筛选

逻辑运算符：与：&& 或：｜｜ 非：！ 也支持and or not

3.模糊查询

like

#一般和通配符搭配使用

通配符：%（任意多个字符，包含0字符）

\_（任意单个字符）

\ （转义符）

escape关键字：自定义一个转义符

between and：

between 临界值 and 临界值；

#可以提高语句简洁度

#包含左右临界值，是闭区间

#两个临界值顺序不能错

in：

in（列表，列表，列表）

#判断某字段的值是否属于in列表的某一项

#使用in提高语句简洁度

#in列表的值类型必须一致或兼容

is null 或 is not null：

#判断是否是null值，=和！=（<>）不能判断mull值

安全等于<=>：

#既可判断null值又可判断数值

排序查询：关键字order by 排序列表 asc升序/desc降序，不写默认升序

select \* from 表名 order by 排序列表 asc/desc；

#order by子句中支持单个字段、多个字段、表达式、函数、别名

#order by子句一般是放在查询语句的最后面，limit子句除外

常见函数

调用：select 函数名（实参列表）

分类：1.单行函数

2.分组函数：做统计使用，又称统计函数、聚合函数、组函数

单行函数：

字符函数：

length（）获取参数值的字节个数

concat（）拼接字符串

upper（）小写变大写 lower（）大写变小写

substr/substring（‘str’，int）截取字符，从int开始往后

substr/substring（‘str’，int1，int2）截取字符，从int1开始截int2个

#mysql索引从1开始

instr（‘str1’，‘str2’）返回子串第一次出现的索引，找不到返回0

trim（‘str’）去除字符串内的空格

trim（‘str1’，‘str2’）去除str2前后的str1

lpad（‘str’，int，‘字符’）用指定字符实现左填充指定长度

rpad（‘str’，int，‘字符’）用指定字符实现右填充指定长度

replace（‘str1’，‘str2’，‘str3’）用str3替换str1里的str2

数学函数：

round（float）四舍五入

round（float，int）四舍五入，保留小数点后int位

ceil（float）向上取整，负数取大值

floor（float）向下取整，负数取小值

truncate（float，int）截断，小数点后截取int位

mod（int，int）取模/取余，原理：a-a/b\*b

日期函数：返回全部/部分时间的函数

now（）返回当前系统日期+时间

curdate（）返回当前日期不包含时间

curtime（）返回当前时间不包含日期

year（）返回日期中的年份

month（）返回日期中的月份

monthname（）返回日期中的月份的英文

day（）返回日期中的日子

hour（）返回日期中的小时

minute（）返回日期中的分钟

second（）返回日期中的秒

datediff（日期1，日期2）返回日期1与日期2的差值

str\_to\_date（日期，格式）将日期格式的字符转换成指定格式的日期

#mysql只认识xxxx-mm-dd标准格式的日期，该函数将非标准格式日期解释给mysql，主要用于将输入的日期转换成计算机语言

date\_format（日期，格式）将日期转换成字符

#将输出的日期转换成用户想要的格式

其他函数

version（）返回当前mysql版本号

database（）返回当前所在库名

databases（）返回所有库名

user（）返回当前用户名

流程控制函数

if（被判断语句，是：返回的值，不是：返回的值）

case函数使用情况一

case（要判断的字段或表达式）

when 常量1 then 要显示的值1或语句1；#值不加分号，语句加分号

when 常量2 then 要显示的值2或语句2；

...

else 要显示的值n或语句n；

end

case函数使用情况二

case

when 条件1 then 要显示的值1或语句1；

when 常量2 then 要显示的值2或语句2；

...

else 要显示的值n或语句n；

end

分组函数：

sum（）求和

avg（）平均值

max（）最大，支持英文顺序

min（）最小，支持英文顺序

count（）计算个数

count函数详细介绍：

count（字段）统计表内字段下所有不为null的行数

count（\*）统计表内所有字段下不为null的行数，经常用来统计行数

count（常量）相当于在表内添加一列常量并统计所添加的行数

效率：

MYISAM存储引擎下，count（\*）效率高

INNODB存储引擎下，count（\*）和count（1）效率差不多，比count（字段）高

#sum、avg一般用于处理数值型，max、min、count可以处理任何类型

#以上分组函数都忽略null值

#可以和distinct搭配实现去重的运算

#一般使用count（\*）统计行数

#和分组函数一同查询的字段有限制，要求是group by的字段

分组查询：group by关键字（每个）

select 分组函数，列（出现在group by后面）from 表名 where group by 列

#搭配分组函数使用

分组查询后筛选：having关键字

#分组查询中的筛选条件分为两类

数据源 位置 关键字

分组前筛选 原始表 group by子句前 where

分组后筛选 分组后的结果集 group by子句后 having

#分组函数做条件肯定用having

DML（data manipulation language）数据操作语言 主要是增删改数据

DDL（data define language）数据定义语言 创建与删除库和表

TCL（transaction control language）树控制语言