命令行打开MySQL：

Win+r ,输入cmd，再按CTRL+shift+enter，进入管理员模式

首先跳转到mysql的bin目录下：（我没跳转也能用？！）

E：

cd E:\MySQL\mysql-8.0.27-winx64\mysql-8.0.27-winx64\bin

以下命令可能需要：

net start mysql

mysql -uroot -p

数据库操作：(所有语句最后要有;)（操作正确显示Query OK）

创建数据库：create database database\_name;

查看数据库：show databases；

删除数据库：drop database database\_name；

选择指定的数据库：use database\_name;

查看数据库下的表列表：show tables;(tables这里是固定的！)

建表时comment用于加注释

创建表（eg）：create table cys（

id int primary key comment ‘编号’,

name varchar(9),

// sa char(10) default ‘123‘ //取默认值

age int nou null) //最后一个属性后无逗号

DEFAULT CHARSET `UTF8`;（注意这里用反引号）

表格

描述已自动生成

插入数据：insert into table\_name（属性序列） values(属性值序列);

查看表结构（不是看存的数据！）;desc table\_name;

图片包含 文本

描述已自动生成

**注： SQL语句结束可以使用‘;’形式，也可以使用‘\G’，该形式查看的数据更清晰。**

修改表:

alter table old\_table\_name rename new\_table\_name; 改表名

修改表名：rename table old\_table\_name to new\_table\_name

 修改字段属性名(同时也改了属性类型)

alter table table\_name change 旧属性名 新属性名 新数据类型;

修改属性类型

alter table table\_name modify 属性名 属性类型;

 增加字段属性

alter table table\_name add 属性名 类型 [完整性约束] [first|after 属性]

 删除字段属性

alter table table\_name drop 属性名;

修改字段排列顺序

alter table table\_name modify 属性名1 类型 first|after 属性名2;

使用方法：

alter table table\_name modify 属性名1 类型 first; //放在第一位置

alter table table\_name modify 属性名1 类型 after 属性名2; //指定属性1放在属性名2后面

查询表：

SQL的基础结构如下：

select \* from table\_name

[where 条件表达式]

[group by 属性名 [having 条件表达式]]

[order by 属性名 [ASC|DESC]]

带in的子查询

一般放在where条件后，批量查询

[not] in(元素1、元素2...元素n)

带between and的范围查询

[not] between 元素1 and 元素2

带like的模糊匹配

[not] like ‘字符串’

like后模糊匹配针对字符串可以携带通配符，‘%’表示0个或者多个字符串，‘-’只能表示单个字符串

空值查询

is [not ] null

带AND的多条件查询

添加表达式1 and 条件表达式2...and 条件表达式n

带OR的多条件查询

添加表达式1 or 条件表达式2 。。。or 条件表达式n

去重处理

select distinct 属性名 from 表

对结果排序

order by 属性名 [ASC|DESC]

order by属性后不添加默认是升序[asc] ASC:升序 DESC：降序

分组查询

group by 属性名 [having 条件表达式]

limit分页查询

limit len; //limit带一个参数，表示不指定初始位置的limit，表示从0好位置开始获取len长度数据返回

limit index len;//带两个参数，第一个参数表示起始位置，第二个参数表示每次读取的长度

数据库中数据处理的SQL

图片包含 文本

描述已自动生成文本

描述已自动生成图形用户界面, 文本

描述已自动生成

删除一个表：

drop table table\_name; //这样结构和数据全删

truncate table table\_name; //这样数据全删，结构保留（查表还在，但不能带where）

delete from table\_name; //数据全删，结构保留（查表还在，能带where）eg:delete from mytable where id = 1;

别名：

select t.id ‘我的编号’ from mytable t;

显示时间：

curdate(),curtime(),now(),sysdate() (now() 在执行开始时值就得到了， sysdate() 在函数执行时动态得到值)

## 获取当前时间戳函数：current\_timestamp, current\_timestamp()

## 日期/时间转换为字符串 函数：date\_format(date,format), time\_format(time,format)

## MySQL 日期、时间转换函数：date\_format(date,format), time\_format(time,format) 能够把一个日期/时间转换成各种各样的字符串格式。

## 它是 str\_to\_date(str,format) 函数的 一个逆转换。

## mysql> select date\_format('2020-07-24 11:32:24', '%Y%m%d%H%i%s');

## +----------------------------------------------------+

## | date\_format('2020-07-24 11:32:24', '%Y%m%d%H%i%s') |

## +----------------------------------------------------+

## | 20200724113224 |

## +----------------------------------------------------+

## 1

## 2

## 3

## 4

## 5

## 6

## 2.2 字符串转换为日期 函数：str\_to\_date(str, format)

## 可以看到，str\_to\_date(str,format) 转换函数，可以把一些杂乱无章的字符串转换为日期格式。

## 另外，它也可以转换为时间。“format” 可以参看 MySQL 手册。

## select str\_to\_date('07/24/2020', '%m/%d/%Y'); -- 2020-07-24

## select str\_to\_date('07/24/2020' , '%m/%d/%y'); -- 2020-07-24

## select str\_to\_date('07.24.2020', '%m.%d.%Y'); -- 2020-07-24

## select str\_to\_date('11:32:24', '%h:%i:%s'); -- 11:32:24

## select str\_to\_date('07.24.2020 11:32:24', '%m.%d.%Y %h:%i:%s'); -- 2020-07-24 11:32:24

## 1

## 2

## 3

## 4

## 5

## 2.3 日期/天数转换 函数：to\_days(date), from\_days(days)

## select to\_days('0000-00-00'); -- 0(日期转换为天数)

## select to\_days('2020-07-24'); -- 737995(日期转换为天数)

## select from\_days(0); -- 0000-00-00(天数转换为日期)

## select from\_days(737995); -- 2020-07-24(天数转换为日期)

## 1

## 2

## 3

## 4

## 2.4 时间/秒转换 函数：time\_to\_sec(time), sec\_to\_time(seconds)

## select time\_to\_sec('01:00:05'); -- 3605(时间转换为秒数)

## select sec\_to\_time(3605); -- 01:00:05(秒数转换为时间)

## 1

## 2

## 2.5 拼凑日期/时间 函数：makdedate(year,dayofyear), maketime(hour,minute,second)

## select makedate(2020,31); -- 2020-01-3(拼凑日期)

## select makedate(2020,32); -- 2020-02-01(拼凑日期)

## select maketime(12,15,30); -- 12:15:30(拼凑时间)

## 1

## 2

## 3

## 2.6 Unix时间戳/日期 转换函数

## unix\_timestamp(),

## unix\_timestamp(date),

## from\_unixtime(unix\_timestamp),

## from\_unixtime(unix\_timestamp,format)

## 下面是示例：

## select unix\_timestamp(); -- 1595563223

## select unix\_timestamp('2020-07-24'); -- 1595520000

## select unix\_timestamp('2020-07-24 12:02:05'); -- 1595563325

## select from\_unixtime(1595563223); -- 2020-07-24 12:00:23

## select from\_unixtime(1595520000); -- 2020-07-24 00:00:00

## select from\_unixtime(1595563325); -- 2020-07-24 12:02:05

## select from\_unixtime(1595563325, '%Y %D %M %h:%i:%s %x'); -- 2020 24th July 12:02:05 2020

## 1

## 2

## 3

## 4

## 5

## 6

## 7

## 3.MySQL 日期时间计算函数

## 3.1 为日期增加一个时间间隔：date\_add()

## set @dt = now();

## select date\_add(@dt, interval 1 day); -- add 1 day

## select date\_add(@dt, interval 1 hour); -- add 1 hour

## select date\_add(@dt, interval 1 minute); -- ...

## select date\_add(@dt, interval 1 second);

## select date\_add(@dt, interval 1 microsecond);

## select date\_add(@dt, interval 1 week);

## select date\_add(@dt, interval 1 month);

## select date\_add(@dt, interval 1 quarter);

## select date\_add(@dt, interval 1 year);

## select date\_add(@dt, interval -1 day); -- sub 1 day

## 1

## 2

## 3

## 4

## 5

## 6

## 7

## 8

## 9

## 10

## 11

## 12

## 13

## MySQL adddate(), addtime()函数，可以用 date\_add() 来替代。下面是 date\_add() 实现 addtime() 功能示例：

## mysql> set @dt = '2020-07-24 12:02:05';

## mysql> select date\_add(@dt, interval '01:15:30' hour\_second);

## +------------------------------------------------+

## | date\_add(@dt, interval '01:15:30' hour\_second) |

## +------------------------------------------------+

## | 2020-07-24 13:17:35 |

## +------------------------------------------------+

## mysql> select date\_add(@dt, interval '1 01:15:30' day\_second);

## +-------------------------------------------------+

## | date\_add(@dt, interval '1 01:15:30' day\_second) |

## +-------------------------------------------------+

## | 2020-07-25 13:17:35 |

## +-------------------------------------------------+

## 1

## 2

## 3

## 4

## 5

## 6

## 7

## 8

## 9

## 10

## 11

## 12

## 13

## 3.2 为日期减去一个时间间隔：date\_sub()

## MySQL date\_sub() 日期时间函数 和 date\_add() 用法一致，不再赘述。

## mysql> select date\_sub('2020-07-24 12:02:05', interval '1 1:1:1' day\_second);

## +----------------------------------------------------------------+

## | date\_sub('2020-07-24 12:02:05', interval '1 1:1:1' day\_second) |

## +----------------------------------------------------------------+

## | 2020-07-23 11:01:04 |

## +----------------------------------------------------------------+

## 1

## 2

## 3

## 4

## 5

## 6

## 3.3 日期、时间相减函数：datediff(date1,date2), timediff(time1,time2)

## MySQL datediff(date1,date2)：两个日期相减 date1 - date2，返回天数。

## mysql> select datediff('2020-07-08', '2020-07-01'); -- 7

## mysql> select datediff('2020-07-01', '2020-07-08'); -- -7

## 1

## 2

## MySQL timediff(time1,time2)：两个日期相减 time1 - time2，返回 time 差值。

## mysql> select timediff('2020-07-24 13:17:35', '2020-07-24 00:00:00'); -- 13:17:35

## mysql> select timediff('13:17:35', '00:00:00'); -- 13:17:35

## 1

## 2

## 注意：timediff(time1,time2) 函数的两个参数类型必须相同。

## 3.4 时间戳（timestamp）转换、增、减函数：

## timestamp(date) – date to timestamp

## timestamp(dt,time) – dt + time

## timestampadd(unit,interval,datetime\_expr)

## timestampdiff(unit,datetime\_expr1,datetime\_expr2)

## 请看示例部分：

## -- 转换

## select timestamp('2020-07-24'); -- 2020-07-24 00:00:00

## select timestamp('2020-07-24 13:17:35', '01:01:01'); -- 2020-07-24 14:18:36

## select timestamp('2020-07-24 13:17:35', '10 01:01:01'); -- 2020-08-03 14:18:36

## -- 增

## select timestampadd(day, 1, '2020-07-24 13:17:35'); -- 2020-07-25 13:17:35

## select date\_add('2020-07-24 13:17:35', interval 1 day); -- 2020-07-25 13:17:35

## MySQL timestampadd() 函数类似于 date\_add()。

## -- 减

## select timestampdiff(year,'2020-07-24','2018-01-01'); -- -2

## select timestampdiff(day ,'2020-07-24','2018-01-01'); -- -935

## select timestampdiff(hour,'2020-07-24 12:00:00','2020-07-24 00:00:00'); -- -12

## select datediff('2020-07-24 12:00:00', '2020-07-20 00:00:00'); -- 4

## MySQL timestampdiff() 函数就比 datediff() 功能强多了，datediff() 只能计算两个日期（date）之间相差的天数。

## 4.MySQL 时区（timezone）转换函数：convert\_tz(dt,from\_tz,to\_tz)

## select convert\_tz('2020-07-24 12:00:00', '+08:00', '+00:00'); -- 2020-07-24 04:00:00

## 时区转换也可以通过 date\_add, date\_sub, timestampadd 来实现。

## select date\_add('2020-07-24 12:00:00', interval -8 hour); -- 2020-07-24 04:00:00

## select date\_sub('2020-07-24 12:00:00', interval 8 hour); -- 2020-07-24 04:00:00

## select timestampadd(hour, -8, '2020-07-24 12:00:00'); -- 2020-07-24 04:00:00

## 日程表 中度可信度描述已自动生成

## 获取当前日期及[格式化输出](https://so.csdn.net/so/search?q=%E6%A0%BC%E5%BC%8F%E5%8C%96%E8%BE%93%E5%87%BA&spm=1001.2101.3001.7020) 获取系统日期：now() 格式化日期：date\_format(date,format)