数据类型

整数：int，bit

小数：decimal

字符串：varchar,char

日期时间: date, time, datetime

枚举类型(enum) 下标从1开始

1.decimal表示浮点数，如decimal(5,2)表示共存5位数，小数占2位

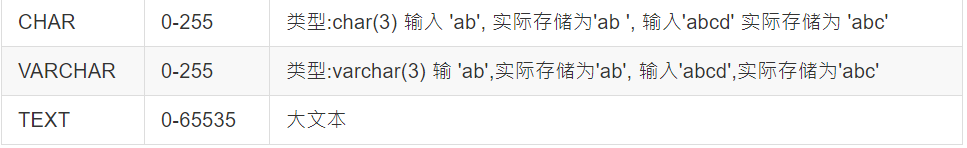
2.char表示固定长度的字符串，如char(3)，如果填充'ab'时会补一个空格为'ab '

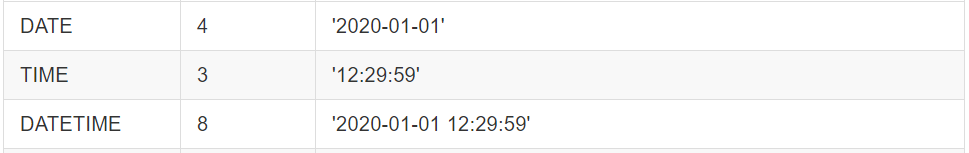
3.varchar表示可变长度的字符串，如varchar(3)，填充'ab'时就会存储'ab'

4.字符串text表示存储大文本，当字符大于4000时推荐使用

5.对于图片、音频、视频等文件，不存储在数据库中，而是上传到某个服务器上，然后在表中存储这个文件的保存路径







约束

主键primary key：物理上存储的顺序

非空not null：此字段不允许填写空值

惟一unique：此字段的值不允许重复

默认default：当不填写此值时会使用默认值，如果填写时以填写为准

外键foreign key：对关系字段进行约束，当为关系字段填写值时，会到关联的表中查询此值是否存在，如果存在则填写成功，如果不存在则填写失败并抛出异常

SQL命令

连接数据库

mysql -uroot -pmysql

退出

exit/ quit/ ctrl+d

查看所有数据库

show databases;

查看时间

select now();

查看版本

select version();

`…` ---->整体

创建数据库

create database 数据库名 charset=utf8;

删除数据库

drop database 数据库名;

查看当前使用的数据库

select database();

使用数据库

use 数据库名;

查看当前数据库中所有表

show tables;

创建表

auto\_increment表示自动增长

CREATE TABLE table\_name(

column1 datatype contrai,

column2 datatype,

column3 datatype,

......

columnN datatype,

PRIMARY KEY(one or more columns)

);

eg:

create table students(

id int unsigned primary key auto\_increment not null,

name varchar(20) default '',

age tinyint unsigned default 0,

height decimal(5,2),

gender enum('男','女','人妖','保密'),

cls\_id int unsigned default 0

);

查看表结构

desc 表名;

添加字段

alter table 表名 add 列名 类型;

例：

alter table students add birthday datetime;

修改字段：重命名版

alter table 表名 change 原名 新名 类型及约束;

例：

alter table students change birthday birth datetime not null;

修改字段：不重命名版

alter table 表名 modify 列名 类型及约束;

例：

alter table students modify birth date not null;

删除字段

alter table 表名 drop 列名;

例：

alter table students drop birthday;

删除表

drop table 表名;

例：

drop table students;

增删改查

增加(主键列是自动增长，通常使用0或者 default 或者 null 来占位)

全列插入:

insert into 表名 values(...)

例：

insert into students values(0,’郭靖‘,1,'蒙古','2016-1-2');

部分列插入：值的顺序与给出的列顺序对应

insert into 表名(列1,...) values(值1,...)

例：

insert into students(name,hometown,birthday) values('黄蓉','桃花岛','2016-3-2');

全列多行插入：值的顺序与给出的列顺序对应

insert into 表名 values(...),(...)...;

例：

insert into classes values(0,'python1'),(0,'python2');

修改

update 表名 set 列1=值1,列2=值2... where 条件

例：

update students set gender=0,hometown='北京' where id=5;

查询

查询所有列

select \* from 表名 where 条件;

select a as A,b as B from 表名 where 条件;

例：

select \* from classes where secore>60;

删除

物理删除(少用)

delete from 表名 where 条件

例：

delete from students where id=5;

逻辑删除，本质就是修改操作

alter table students add is\_delete bit default 0;

update students set is\_delete=1 where id=1;

SQL查询

查询所有字段

select \* from 表名;

查询指定字段

select 列1,列2,... from 表名;

使用 as 给字段起别名

select id as 序号, name as 名字, gender as 性别 from students;

可以通过 as 给表起别名

select students.id,students.name,students.gender from students;

select s.id,s.name,s.gender from students as s;

消除重复行

在select后面列前使用distinct可以消除重复的行

select distinct 列1,... from 表名;

where后面支持多种运算符，进行条件的处理

模糊查询

like

%表示任意多个任意字符

\_表示一个任意字符

rlike 用正则表达式查找

范围查询

in表示在一个非连续的范围内

between ... and ...表示在一个连续的范围内

空判断

判空is null

判非空is not null

排序

select \* from 表名 order by 列1 asc|desc [,列2 asc|desc,...]

asc从小到大排列，升序

desc从大到小排序，降序

总数

例：查询学生总数

select count(\*) from students;

最大值

max(列)表示求此列的最大值

例：查询女生的编号最大值

select max(id) from students where gender=2;

最小值

min(列)表示求此列的最小值

例：查询未删除的学生最小编号

select min(id) from students where is\_delete=0;

求和

sum(列)表示求此列的和

例：查询男生的总年龄

select sum(age) from students where gender=1;

平均值

avg(列)表示求此列的平均值

例：查询未删除女生的编号平均值

select avg(id) from students where is\_delete=0 and gender=2;

四舍五入 round(数据,保留指定位小数)

ed: round(12.45,1) 🡪12.5

分组

group by /group\_concat([“\_”,]字段名)可以作为一个输出字段来使用

select gender from students group by gender;

select gender,group\_concat(name) from students group by gender;

group by + having

having 条件表达式：用来分组查询后指定一些条件来输出查询结果

having作用和where一样，但having只能用于group by

select gender,count(\*) from students group by gender having count(\*)>2;

分页

select \* from 表名 limit start,count

求第n页的数据每页m条数据

select \* from students where is\_delete=0 limit (n-1)\*m,m

内连接 左连接 右连接

select \* from 表1 [as 1] inner或left或right join 表2 [as 1] on 表1**.列 = 表2.列**

自关联

根据本表字段间的关系,创建两个命名不同的表.

插入外键

给brand\_id 添加外键约束成功

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

删除外键

-- 需要先获取外键约束名称,该名称系统会自动生成,可以通过查看表创建语句来获取名称

show create table goods;

-- 获取名称之后就可以根据名称来删除外键约束

alter table goods drop foreign key 外键名称;

python与mysql交互

在py文件中引入pymysql模块

from pymysql import \*

创建对象：调用connect()方法

conn=connect(host='localhost',port=3306,database='jing\_dong',user='root',password='mysql',charset='utf8')

获取Cursor对象：调用Connection对象的cursor()方法

cs1=conn.cursor()

……

cs1.close()

conn.close()

cs1.execute(‘SQL语句’)

cs1. fetchone() 以元组形式返回一条数据,

cs1. fetchmany(num) 指定

cs1. fetchall() 所有

cs1.execute(‘SQL增加’)后

增加需要提交

conn.commit()

撤销

conn.rollback()

‘ or 1=1 or ‘1

视图

create view 视图名 as (查询的表)

drop view 视图名

事务特性

原子性 一致性 隔离性 持久性

事务开启

begin; 或者 start transaction;

提交事务

commit;

回滚事务

Rollback;

索引

show index from 表名;

创建索引

如果指定字段是字符串，需要指定长度，建议长度与定义字段时的长度一致

字段类型如果不是字符串，可以不填写长度部分

create index 索引名称 on 表名(字段名称(长度))

删除索引：

drop index 索引名称 on 表名;

创建新用户并设置权限

grant 权限列表 on 数据库 to '用户名'@'访问主机' identified by '密码' ;

grant select on jing\_dong.\* to 'laowang'@'localhost' identified by '123456';

all privileges 所有权限

grant 权限列表 on 数据库 to '用户名'@'访问主机' with grant option; 修改权限

刷新权限：flush privileges

删除用户

drop user '用户名'@'主机';

备份

mysqldump –uroot –p 数据库名 > python.sql;

恢复

mysql -uroot –p 新数据库名 < python.sql

聚合函数

avg(元素) 求平均值

min(元素) 求最小值

max(元素) 求最大值

sum(元素) 求和

sum(判断条件) 统计满足条件的数量

count(元素) 统计元素个数