# 前言 关于数据库

数据库系统解决的问题：持久化存储，优化读写，保证数据的有效性。

当前使用的数据库，主要分为两类

文档型：如sqlite，就是一个文件，通过对文件的复制完成数据库的复制；

服务型：如mysql，数据存储在物理文件中，使用终端以tcp/ip协议进行数据操作。

E-R模型：当前物理的数据库都是按照E-R模型进行设计的

E表示entry，实体

R表示relationship，关系

一个实体转换为数据库中的一个表

关系描述两个实体之间的对应规则，包括

一对一

一对多

多对多

关系转换为数据库表中的一个列；

在关系型数据库中一行就是一个对象。

三范式：设计数据库的一些规范被称为范式，后一个范式，都是在前一个范式的基础上建立的。

第一范式（1NF)：列不可拆分

第二范式（2NF)：唯一标识

第三范式（3NF)：引用主键

一个数据库就是一个完整的业务单元，可以包含多张表，数据被存储在表中

在表中为了更加准确的存储数据，保证数据的正确有效，可以在创建表的时候，为表添加一些强制性的验证，包括数据字段的类型、约束。

和大多数开源产品一样，mysql分为社区版和商业版，这两个版本又个字分四个版本发布：Alpha，Beta，RC，GA（正式版）。

5.7版本已经可以投入生产，性能更高。

# 一 MySQL入门

## 1 数据库操作

登录：mysql -uroot -p

注意：具体的sql语句中，可以使用 ; \g \G 三种形式表示语句结束，前两个表达的意思一样，只是结束标志，第三个表示格式化显示结果。

创建数据库 create database 数据库名 charset=utf8;

删除数据库 drop database 数据库名;

选择数据库 use 数据库名;

查看所有数据库 show databases;

查看当前数据库 select database();

## 2 存储引擎

常见的存储引擎有：

InnoDB 支持事务，所以占用空间大，适合频繁更新、删除的数据库

MyISAM 不支持事务、外键，所以访问速度快

MEMORY 使用内存存储数据，访问速度快，适合数据小、速度要求快的数据库

查看所有引擎 show engines;

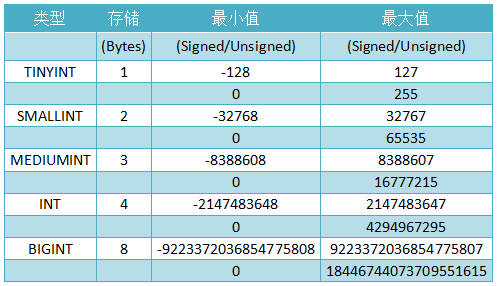
查看默认引擎 show variables like '%storage\_engine%';

修改默认引擎 在my.ini配置文件中修改：default-storage-engine

## 3 数据类型

查询数据类型

### 3.1 整数



### 3.2 小数



需要精确到小数点后10位以上，才会使用DOUBLE。精度要求非常高则可以使用位类型DEC和DECIMAL。FLOAT和DOUBLE存储的是近似值，DECIMAL存储的是字符串。

### 3.3 时间



表示年月日，一般使用DATE类型

表示年月日分秒，使用DATETIME

经常插入或者更新日期为当前系统时间，使用TIMESTAMP

表示分秒，使用TIME

表示年份，使用YEAR

### 3.4 字符串



存储小数据，可以使用CHAR（很少变化选择该类型）或者VARCHAR（经常变化），

## 4 约束

主键 primary key 性能最高，且唯一

非空 not null

惟一 unique

默认 default

外键 foreign key

自动递增auto\_increment

关于主键：  
 一列或者一组列，其值能位移区分表中的每一行。没有主键，更新或者删除表中特定的行很困难，因为没有安全的方法保证只涉及相关的行。虽然并不总是需要主键，但是大多数数据库管理人员都会为每个表创建一个主键，以便于以后的数据管理。

表中的任何列都可以作为主键，只要同时满足：任意两行都不具有相同主键值；每行都有一个主键值，且不为NULL。

使用多列多为主键时，上述条件必须应用到构成主键的所有列，所有列的组合必须唯一。

多字段主键创建案例：使用constraint

create table teacher(

name varchar(20),

fathername varchar(20),

sonname varchar(20),

constraint pk\_name\_sonname primary key(name,sonname)

);

## 5 表操作

表中的数据库对象包括：列、索引、触发器。

查看当前数据库中所有表

show tables;

创建表

create table 表名(列及类型);

create table students(

id int auto\_increment primary key not null,

name varchar(10) not null,

gender bit default 1,

birthday datetime

);

修改表

alter table 表名 add|change|drop 列名 类型;

alter table students add birthday datetime;

add 字段名 字段类型 在表的最后一个位置增加字段

add 字段名 字段类型 first 在表的第一个位置添加字段

add 字段名 字段类型 after 字段 在指定字段后添加字段

drop 字段名 删除字段

modify 字段名 字段类型 修改字段类型

modify 字段名1 字段类型 first|after 字段名2 修改字段顺序

change 字段名 新字段名 字段类型 修改字段名

change 字段名 新字段名 新字段类型 修改字段名、字段类型

删除表

drop table 表名;

更改表名称

rename table 原表名 to 新表名;

查看表的创建语句

show create table '表名';

查看当前表中所有列：

show columns from 表名;

describe 表名; //快速写法，也可以简写为：desc 表名;

## 6 数据操作

增加

全列插入：insert into 表名 values(值1,值2...)

缺省插入：insert into 表名(列1,...) values(值1,...)

同时插入多条数据：

insert into 表名

values(值1,值2,值3...),

(值1,值2,值3...),

(值1,值2,值3...),

或者： insert into 表名(列1,...) values(值1,...),(值1,...)...;

注意：自动增加约束、默认值约束的字段可以不用插入数值。

插入查询语句：可以实现从一个表复制数据到另一个表

insert into 表名(字段1,字段2,字段3...) select ......

删除

delete from 表名 where 条件

逻辑删除，本质就=是修改update：alter table students add isdelete bit default 0;

如果需要删除则：update students isdelete=1 where ...;



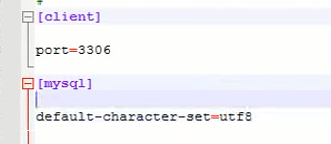
修改

update 表名 set 列1=值1,... where 条件

查询

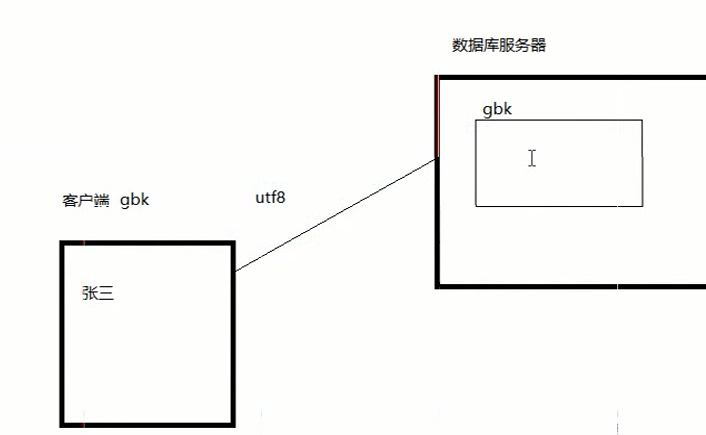
select \* from 表名

## 7 命令行插入中文乱码



注意：需要先关闭服务，修改后重启服务

即使客户端、服务端都是gbk的编码，存储的时候仍然是乱码，因为数据的传输中被编译成了utf8：



# 二 查询入门

## 1 select

select \* from user; 查询全部列

select username from user; 查询指定列

select username,password from user; 查询指定多个列

select username as name from user; 为列起个别名，列以别名会出现在结果集中

select distinct gender from user; distinct关键字用来消除重复行

select age+1 from user; 查询的列的值可以进行数学运算

## 2 筛选：where与运算符

select \* from 表名 where 条件;

条件：使用where子句对表中的数据筛选，结果为true的行会出现在结果集中

比较运算符：= > >= < <= !=或<>

select \* from students where id>3;

select \* from students where sname!='黄蓉';

select \* from students where isdelete=0;

逻辑运算符：and or not

select \* from students where id>3 and gender=0;

其他条件：like，between and，is null，in。

## 3 模糊查询：like与通配符

使用like进行模糊查询：

% 表示任意多个任意字符， \_ 表示一个任意字符

select \* from students where name like '黄%'; 查询姓黄的学生

select \* from students where name like '黄\_'; 查询姓黄且名字是一个字的学生

select \* from students where name like '黄%' or name like '%靖%'; 查询姓黄或叫靖的

注意：不要过度使用通配符，尽量使用别的方式达到相同目的（为了提升性能）；

尽量不要把通配符置于搜索模式的开始处，因为这样搜索最慢；

## 5 范围查询：in以及not in与between and

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

select \* from students where id in(1,3,8); 查询编号是1或3或8的学生

select \* from students where id between 3 and 8;查询学生是3至8的学生

select \* from students where id between 3 and 8 and gender=1;查询学生是3至8的男生

## 4 空判断：is null

判空is null，判非空is not null，注意：null与''是不同的

select \* from students where hometown is null; 没有填写地址的学生

select \* from students where hometown is not null; 填写了地址的学生

select \* from students where hometown is not null and gender=0; 填写了地址的女生

## 5 优先级

小括号，not，比较运算符，逻辑运算符

and比or先运算，如果同时出现并希望先算or，需要结合()使用