RDBMS（关系型数据库管理系统）

**登录**

Mysql -u用户名 -p密码 -h主机地址 -p端口号 链接mysql

Sudo vim /etc/mysql/mysql.conf.d/mysql.cnf MYSQL配置文件

Bind-address 127.0.0.1 默认绑定的IP，注释掉，其它主机才能连接（不建议，ssh远比它安全）

如果设置了skip\_networking 只能本地登录

**SQL语句主要分为**

**DQL数据查询语言，用于i对数据进行查询 select**

**DML数据操作语言，对数据进行增加、修改、删除，insert、update、delete**

TPL事务处理语言，对事务进行处理，包括begin transaction、commit、rollback

DCL数据控制语言，进行授权与权限回收，如grant、revoke

DDL数据定义语言，进行数据库、表的管理等，如create、drop

CCL指针控制语言，通过指针完成表的操作，如declare cursor

**常用数据类型：**

整数：int，bit

小数：decimal

字符串：varchar，char

日期时间：date，time，datetime

枚举类型：enum

特别说明的 类型：

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

Char表示固定长度的字符串，如char(3)，如果填充’ab’时就会补一个空格位’ab ’

Varchar表示可变长度的字符串，如varchar(3)，填充’ab’时就会存储’ab’，不允许超过最大值

字符串text表示存储大文本，当字符大于4000时推荐使用

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：此字段不允许填写控制

唯一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主

外键foreign key：对关系字段进行约束，当为关系字段填写值时，会关联到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

说明：虽然外键约束可以保证数据的有效性，但是在进行数据的crud时，都会降低数据库的性能，所以不推荐使用。通常在逻辑层进行控制

Show databases 查看数据库

Select now() 显示当前时间

Selcet version() 查询版本

Create database python　创建数据库

Show create database python 显示怎么创建的

drop database python 删除数据库（有 – 或空格用 ``符号包住库名）

select database() 查看数据库

use python 使用数据库

show tables 显示所有数据表

desc xxx 查看表结构

create table students(

id int unsigned not null primary key,

name varchar(30),

age tinyint unsigned,

high decimal(5,2),

gender enum("男","女","中性","保密") default "保密",

cls\_id int unsigned

) 创建表

insert into students values(0,"老王",18,188.88,"男",0) 插入数据

修改表结构 alter table

alter table students add birthday datetime 添加字段

alter table students modify birthday date 修改字段

alter table students change birthday birth date default "1990-01-01" 改变字段

alter table students drop high 删除字段

drop table students 删除表

insert into students values(default,"成龙",20,"男",1,"1990-01-01") 可以填null ，default

枚举类型字段可以用填写数字

insert into students (name,gender) values("张三",1) 部分插入

insert into students (name,gender) values("大乔",2),("貂蝉",2) 多行插入

如果字段设置为默认值，不能填0、null只能用default

update students set gender=1 修改全部数据

update students set gender=1 where name="张三" where限定

alter table 表名 add/modify/change/drop 字段

insert into 表名 (字段1，字段2，字段3….) values(…..)

updte 表名 set 字段 where 条件

数据通常逻辑删除

Alter table students add is\_delete bit default 0

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

select \* from students

select id as 序号,name as 姓名 from classes

select students.name,students.age from students as s

数据表取了别名后不能用原名

select distinct gender from students distinct去重

identity(1,1) 从1开始递增，每次自增1

**条件查询**

比较运算符

>,<,<=,>=,=,!=或者<>

逻辑运算符

And,or,not 注意运算符优先级

Select \* from students where not (age >18 and gender=2)

**模糊查询**

like

%替换一个或者多个

\_替换一个

Select name from students where name = “\_小%”

rlike正则

select name from students where name rlike “^周.\*伦$”

**范围查询**

In,not in,between … and …, not between … and …

Select name,age from students where age not in (12,18,34)

Select name,age from students where not between 18 and 20

Not between and是写法，不是between and 的取反，加括号会报错

**空判断**

Is null，is not null

Select \* from student where height is null

**排序**

Order by 字段

Asc 升序（从小到大），desc 降序(默认升序)

Select \* from where (age between 18 and 20) and gender=1 order by age desc,id desc

**聚合函数**

Count 总数,max最大值,min最小值,sum求和,avg平均值,round(123.23,1)保留小数

Select count(\*) as 男性人数 from students where gender=1

Select max(age) from students where gender=2

Select round(sum(age)/count(\*),3) from students

**分组**

Group by

Select gender(唯一能标识组的字段),count(\*)(对分组的结果进行聚合) from students group by gender

分组的意义是和聚合函数一起用

Select gender,group\_concat(name) from students group by gender group\_concat()

Select gender,group\_concat(name,age) from students where gender=1 group by gender

Where不能写在group by后面

Having 条件，对分组结果进行判断

**分页**

Limit start（开始的位置）,count（几条）

Select \* from students where gender=1 limit 2

Select \* from students limit 0,5

Limit (第N页-1)\*每个的个数,每页的页数

Select \* from students limit 2\*(6-1),2 order by age ×

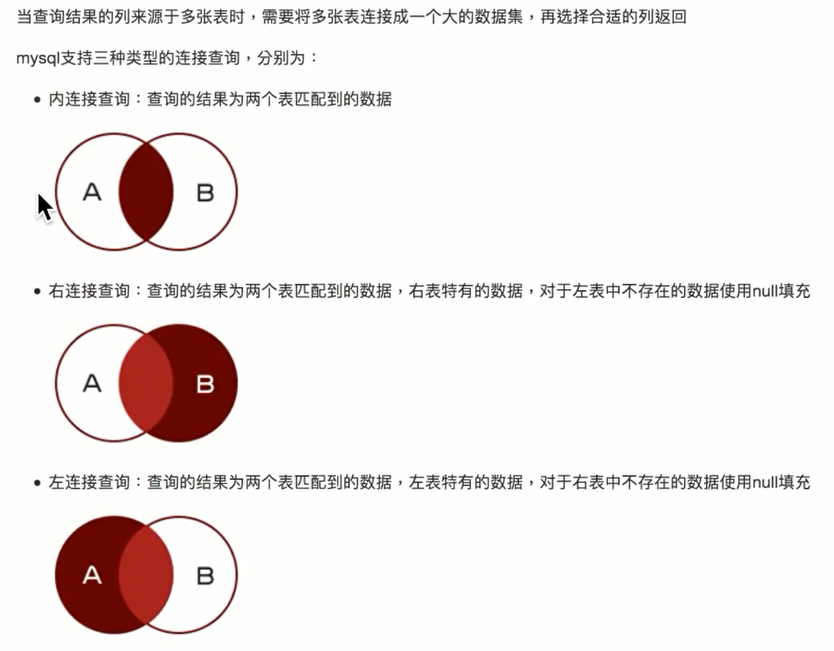
Select \* from students order by age 10,2

Select \* from students where gender=2 order by height desc limit 0,2

**连接查询**

Inner join … on 条件

Left join/right join



Select \* from students inner join classes(结果为笛卡尔积)

Select \* from student inner join classes on students.cls\_id=classes.id(交集)

Select c.name,s.\* from students as s innder join classes as c on s.cls\_id=c.id order by c.name,s.id

Select \* from students as s left join classes as c on s.cls\_id=c.id having c.id is null (用where也可)

从原表找就用where，从结果找就用having

**自联结**

一个表的字段关联另一个表的字段

Select province.atitle,city.atitle from areas as province inner join areas as city on city.pid=province.aid having province.atitle=”四川省”

**子查询**

Select里面嵌套Select

Select \* from students height = (select max(height) from students) 优先运行子查询

Select \* from areas where pid =(select aid from areas where atitle=”四川省”)

SQL语句没有固定格式，只有在效率上有区别，子查询效率比自联结低

**数据库设计**

（目前有迹可寻的范式有8个，一般到三范式即可）

三范式（Normal Form）

第一范式：不能再拆分

第二范式：必须有主键，非主键字段必须完全依赖与主键，不能存在部分依赖

第三范式：非主键的列必须直接依赖与主键，不能存在传递依赖。

create table if not exists goods\_cates(

id int unsigned primary key auto\_increment,

name varchar(40) not null

)

insert into goods\_cates (name) select cate\_name from goods group by cate\_name

select \* from goods\_cates

update goods as g inner join goods\_cates as c

on g.cate\_name=c.name set g.cate\_name=c.id

alter table goods change cate\_name cate\_id int unsigned not null

alter table goods add foreign key (cate\_id) references goods\_cates(id)

create table goods\_brands(

id int unsigned primary key auto\_increment,

name varchar(40) not null

)select brand\_name as name from goods group by brand\_name

show tables

select \* from goods\_brands

update goods as g inner join goods\_brands as b

on g.brand\_name=b.name set g.brand\_name=b.id

58到家30条军规：

<https://www.oschina.net/question/54100_2231325>

**SQL注入**

获取个人信息50条（累计）以上就违法

Self.cursor.execute(sql,[find\_name])

**视图**（数据库变了，程序不能变）

Create view 视图名称 as select 语句

通俗的讲，视图就是一条select语句执行后返回的结果集。创建视图的时候，主要的工作落在创建这条SQL查询语句上。

视图是对若干张基本表的引用，一张虚表，查询语句执行后的结果，不存储具体的数据（基本表数据发生了改变，视图也会跟着改变）

视图是为了方便查询的，不能进行增删改

提高重用性，像一个函数

对数据库重构，却不影响程序的运行

提高了安全性能，可以对不同的用户

让数据更加清晰

Drop table 视图名称

**事务**

它是一个操作序列，这些操作要么都执行，要么都不执行，它是一个不可分割的工作单位

Start transaction / begin

Select balance from checking where customer\_id=10233276

Update checking set balance=balance-200.00 where customer\_id=10233276

Update savings set balance=balance+200.00 where customer\_id=10233276

Commit

Rollback回滚

如果任何一个步骤失败都要回滚（回到上一条语句执行后的样子）

事务四大特性（ACID）

原子性（一个事务必须被视为一个不可分割的最小工作但愿，整个事务中的所有操作要么全部提交成功，要么全部失败回滚，对于一个事务来说，不可能只执行其中的一部分操作）

一致性（数据库总是一个一致性的状态转换到另一个一致性的状态）

隔离性（一个事务所做的修改在最终提交以前，对其它事务是不可见的。）

持久性（一旦事务提交，则其所做的修改会永久保存到数据库）

默认是开启了事务的（自动执行commit）

**索引**（数据量大，搜索数据慢时用到）

索引是一种特殊的文件（InnoDB数据表上的索引是表空间的一个组成部分），它们包含着对数据表里所有记录的引用指针

更通俗的说，数据库索引好比是一本书前面的目录，能加快数据库的查询速度。（但是会影响更新和插入的速度，因为它需要同样更新每个索引文件。）

Set profiling=1 开启运行时间检测

Select \* from test\_index where title=’ha-99999’ 查找第1万条数据ha-99999

Show profiles 查看执行的时间

Create index title\_index on test\_index(title(10)) 为表title列创建索引

数据量越大查询效率差距越大

一般的应用系统对比数据库的读写比例在10：1左右

B+树结构存储数据

Show index from goods 查看索引 数据表中的键都是主键

**账户管理**

show databases

use mysql

show tables

desc user

select user,host,authentication\_string from user

create user '用户名'@’访问主机' identified by '密码

grant权限列表on 数据库（具体到表名）to ‘用户名’@’访问主机’ identified by ‘密码’

create user 'hui'@'%' identified by '123456'

grant all privileges on jingdong.\* to 'hui'@'%'

all privileges所有权限

grant select ,insert,update on jingdong.\* to hui@% **with grant option**

flush privileges 刷新权限

drop user ‘用户名’@’主机’（推荐该方法）

delete from user where user=’用户名’

**Mysql主从**

读写分离

通过增加从服务器来提高数据库的性能，在主服务器上执行写入和更新，在从服务器上向外提供读功能，可以动态地调整从服务器的数量，从而调整整个数据库的性能

数据备份

从服务器可以终止**复制**进程，可以在从服务器上备份而不破坏主服务器相应数据

负载均衡

在主服务器上生成实时数据，在从服务器上分析这些数据，从而提高主服务器的性能

在主服务器上，必须开启二进制日志机制和配置一个独立的ID(cnf里的server\_id)

在每个从服务器上，配置一个唯一的ID，创建一个用来专门复制主服务器数据的账号

在复制进程前，在主服务器上记录二进制文件的位置信息

如果开始复制之前，数据库中已经有数据，就必须先创建一个数据快照（可以从mysqldump导出数据库，或直接复制数据文件）

配置从服务器要连接的主服务器的IP地址和登录授权，二进制日志文件名和位置

Mysqldump -uroot -p 数据库名 > poython.sql 备份

Mysql -uroot -p 新数据库名 < python.sql 创建新的数据库（<重定向输入）

Mysqldump -uroot -pmysql –all-databases –lock-all-tables > ~/master\_db.sql

Sudo scp ~/master\_db.sql hui@192.168.1.6:/home/mysql

Mysql -uroot -pmysql < master\_db.sql

Sudo vim /etc/mysql/mysql.conf.d/mysql.cnf

Sudo service mysql restart

Grant replication slave on \*.\* to ‘slave’@’%’ identified by ‘slave’

Flush privileges

Change mater to master\_host=’192.168.1.4’,master\_user=’slave’,master\_password=’slave’,master\_log\_file=’mysql-bin.000006’,master\_log-pos=590

Show slave status \G查看同步状态

注：复制是异步进行的，所以从服务器不需要一直连接着主服务器，从服务器甚至可以通过拨号断断续续地连接主服务器。通过配置文件，可以指定复制所有的数据库、

某个数据库甚至是某个数据库上的某个表。**复制，主服务器上的数据会被复制到从服务器上，从服务器不会复制给主服务器。**