数据库管理系统(Database Management System)是一种操纵和管理数据库的大型软件，用于建立、使用和维护数据库，简称DBMS

**LAMP**是指一组通常一起使用来运行动态网站或者服务器的[自由软件](https://baike.baidu.com/item/%E8%87%AA%E7%94%B1%E8%BD%AF%E4%BB%B6)名称首字母缩写。

1. 使用mysql

USE mysql;

DESC db; 显示表结构

SHOW COLUMNS FROM db;

2.表的创建

/\*创建数据库\*/

CREATE DATABASE jsp;

/\*使用数据库\*/

USE jsp;

/\*开始创建表test1\*/

AUTO\_INCREMENT 自增

CREATE TABLE test1(

id INT NOT NULL AUTO\_INCREMENT,

'name' CHAR(50) NOT NULL,

address CHAR(50) NOT NULL,

card\_id INT NOT NULL UNIQUE,

PRIMARY KEY(id)

)ENGINE =INNODB;

/\*删除表\*/

DROP TABLE test1;

/\*查看当前自动增长的值\*/

SELECT LAST\_INSERT\_ID();

MySQL

Sql分类：

数据定义语言：（DDL）：用来定义数据库对象：数据库（database），表（table），列（column）

关键字：创建（create），查看show，修改（alter），删除（drop）

数据库操作语言：（DML）用来对数据库表中的记录进行更新

关键字：插入insert，删除delete，更新update

数据库查询语言：（DQL）查询数据库中表的记录

关键字：select，from ，where

数据库的操作

1. 创建一个数据库

Create database yige

create database yige character set utf8;

1. 查询数据库

Show databases

1. 删除数据库

Drop databases yige

1. 使用数据库

use yige;

1. 查看当前数据库

select database();

表的操作

1. 创建表

create table user(

-> uid int(32) primary key auto\_increment,

-> uname varchar(32),

-> upassword varchar(32),

-> uage int(32),

-> usex varchar(32)

-> );

auto\_increment：如果插入没有设置该字段值，则从1开始自动增加值

1. 查看表

show tables;

1. 查看表结构

desc user;

1. 删除表

Drop table user;

1. 修改表名

rename talbe user to new\_user;

1. 修改表的编码

alter table new\_user character set utf8;

1. 查看表的编码

show create table new\_user;

列的操作

1.添加列

alter table user add uinfo varchar(32) not null;

1. 修改列的类型

alter table user modify uinfo varchar(100) null;

1. 修改列的名字

alter table user change 旧列名 新列名 varchar(32) not null;

alter table user change uinfo info varchar(32) not null;

1. 删除列

alter table user drop info;

数据的操作

1. 插入数据

insert into new\_user values(null,'yige',123456,23,'男');

insert into new\_user(uid,uname,upassword,uage,usex) values(null,'一个',123456,'32','男');

如果你可以插入英文的字符串，但是不可以插入中文

Set names gbk;

1. 修改数据

不带条件，会修改该列的所有的数据

update new\_user set usex='女';

带条件，则会修改指定的列的数据

1. 删除数据

delete from new\_user where uid=5;

数据查询

1. 查询所有

select \* from new\_user;

1. 查询用户名和年龄

select uname,uage from new\_user;

1. 使用别名

select \* from new\_user as（可以省略） u;

select uname 姓名 from new\_user;

1. 去重

select distinct(uage) from new\_user;

1. 条件查询

select \* from new\_user where uage=23;

select \* from new\_user where uage<32;

select \* from new\_user where uage<=32 and usex='女';

6.查询姓名第二个是个的用户

select \* from new\_user where uname like '%个';

7.查询年龄是23和32的用户

select \* from new\_user where uage in(23,32);

select \* from new\_user where uage=23 or uage=32;

排序

select \* from new\_user order by uage asc;

select \* from new\_user order by uage desc;

聚合函数

1. 求和

select sum(uage) from new\_user;

1. 求平均数

select avg(uage) from new\_user;

1. 统计

select count(\*) from new\_user;

分组

1.统计男和女共多少个

select usex,count(\*) from new\_user group by usex;

1. 带有条件

select usex,uage,count(\*) from new\_user group by usex having uage>20;

总结：

Show：数据库的操作

Group by：分组

Order by：排序

Having：分组后带有条件只能使用having

多表操作：

外键目的：保证数据的完整性，从表创建不能添加主表中不存在的记录，主表不能删除从表中已经引用的记录

表与表之间存在的关系

1. 一对多
2. 一对多创建表原则：从表（多）创建一个字段，字段作为外键执行主表（一）的主键；外键指向主表的主键
3. 外键：

Alter table 从表 add constraint 外键名称 foreign key(外键字段) references 主表（主表主键）;

从表外键的值是对主表主键的引用

从表外键类型必须与主表主键类型一致

外键是哪个分类的商品

1. 多对多

a).多对多的建表原则：需要创建三表，中间表至少需要两个字段，这两个字段分别作为外键指向各自一方的主键

3.一对一

A）外键唯一：主表的主键和从表的外键形成主外键关系，外键唯一 unique

b) 外键是主键：主表的主键和从表的主键，形成主外键关系

多表查询

1. 交叉多表查询（基本不会使用，得到两个表的乘积）
2. 语法：select \* from A,B;
3. 内连接查询（使用关键字inner join 【inner可以省略】）
4. 隐式内连接：select \* from A,B where 条件;
5. 显示内连接：select \* from A inner join B on 条件；
6. 外连接查询（使用关键字 outer join[outer 可以省略]）
7. 左外连接：select \* from A left outer join B on 条件；
8. 右外连接：select \* from A right outer join B on 条件；

内连接和外连接的区别

1. 左外连接;左表的全部及两个的交集（公共）
2. 右外连接：右表的全部及两个的交集（公共）
3. 内连接：查询两个表的交集（公共）

子查询

将一条select语句的结果作为另一条select语法的一部分

举例：查询服饰分类上架的商品信息

Select \* from product where category id=(select cid from category where cname=’服饰’)；

分页查询

使用limit关键字进行查询操作

Limit起始位置，每页显示的数目

每页显示两条数据

查询第一页：select \* from product limit0,2;

计算起始位置公式：要查询的页数-1\*每页要显示的数目

事务

手动事务：

1. 显示的开启一个事务：start transaction
2. 事务的提交：commit代表开启事务到事务提交中间的所有SQL都以为是有效的真正的更新数据库
3. 事物的回滚：rollback代表事务的回滚