练习：查询出工资在8000以上的男员工，结果按照工资的降序排列

select \* from emp where salary>8000 and sex=1 order by salary desc;

一.简单查询 —— 分页查询

查询的结果中有太多数据，一次显示不完可以做成分页

需要两个已知的条件：当前的页码、每页显示的数据量

|  |
| --- |
| 每页开始查询的值 = (当前的页码-1) \* 每页显示的数据量 |

select \* from emp limit 开始查询的值, 每页显示的数据量;

练习：假设每页显示5条数据，查询出前4页的数据

第1页：select \* from emp limit 0,5;

第2页：select \* from emp limit 5,5;

第3页：select \* from emp limit 10,5;

第4页：select \* from emp limit 15,5;

|  |
| --- |
| 注意事项：limit后的两个值必须写数字，不能加引号。 |

二.复杂查询

1.聚合查询/分组查询

示例：查询出所有员工的数量

select count(\*) from emp;

练习：使用员工的编号列查询员工的数量

select count(eid) from emp; #推荐使用主键列

练习：使用员工部门编号列查询员工的数量

select count(deptid) from emp;

|  |
| --- |
| 函数：是一个功能体，需要提供若干个数据返回结果  聚合函数  count()/sum()/avg()/max()/min()  数量 总和 平均 最大 最小 |

练习：查询出所有男员工的工资总和

select sum(salary) from emp where sex=1;

练习：查询出10号部门的平均工资

select avg(salary) from emp where deptid=10;

练习：查询出30号部门的最高工资

select max(salary) from emp where deptid=30;

练习：查询出年龄最大的员工的生日

select min(birthday) from emp;

|  |
| --- |
| 分组查询建立在聚合查询基础之上的，通常只查询聚合函数和分组条件。 |

示例：查询出男女员工的数量，工资总和

select count(eid),sum(salary),sex from emp group by sex;

练习：查询出各部门的平均工资，最高工资，最低工资

select avg(salary),max(salary),min(salary),deptid from emp group by deptid;

|  |
| --- |
| year() 获取日期中的年份部分 |

示例：查询出‘1973-7-15’的年份

select year('1973-7-15');

示例：查询出1993年出生员工有哪些

select \* from emp where year(birthday)=1993;

2.子查询

|  |
| --- |
| 子查询是多个查询命令的组合，把一个的结果作为另一个的条件使用 |

示例：查询出工资最高的员工

步骤1：查询出工资的最高值 —— 50000

select max(salary) from emp;

步骤2：查询最高值对应的员工

select \* from emp where salary=50000;

综合：

select \* from emp where salary=(select max(salary) from emp);

练习：查询出比tom工资高的员工有哪些

步骤1：查询出tom的工资是多少 —— 6000

select salary from emp where ename='tom';

步骤2：查询出工资高于6000的员工

select \* from emp where salary>6000;

综合：

select \* from emp where salary>(select salary from emp where ename='tom');

练习：查询出和tom同一年出生的员工有哪些

步骤1：查询出tom出生的年份 —— 1990

select year(birthday) from emp where ename='tom';

步骤2：查询出1990年出生的员工

select \* from emp where year(birthday)=1990 and ename!='tom';

综合：

select \* from emp where year(birthday)=(select year(birthday) from emp where ename='tom') and ename!='tom';

3.多表查询

查询的数据分布在多个表中，前提是多个表之间建立了关联

示例：查询出所有员工的姓名及其部门名称

select ename,dname from emp,dept where deptid=did;

select emp.ename,dept.dname from emp,dept where emp.deptid=dept.did;

(1)内连接

select ename,dname from emp inner join dept on deptid=did;

和之前的查询结果一致

(2)左外连接

select ename,dname from emp left outer join dept on deptid=did;

显示左侧表中所有的记录，先写哪个表哪个就是左

(3)右外连接

select ename,dname from emp right outer join dept on deptid=did;

显示右侧表中所有的记录，后写哪个表哪个就是右

左外连接和右外连接中outer关键字可以省略的

(4)全连接

同时显示左侧和右侧所有的记录

full join on

mysql不支持全连接

左外和右外结果进行联合

union 联合后，合并相同的记录

union all 联合后，不合并相同的记录

(select ename,dname from emp left outer join dept on deptid=did)

union

(select ename,dname from emp right outer join dept on deptid=did);

学习一门编程语言步骤

1.了解的背景知识：历史、现状、特点、应用场景

2.搭建开发环境，编写hello world

3.变量和常量

4.数据类型

5.运算符

6.逻辑结构

7.通用的小程序

8.函数和对象

9.第三方的库和框架

10.个人项目

程序员必做50题（有基础的）

https://wenku.baidu.com/view/af66e2f14afe04a1b071de42.html

电子书：JavaScript高级程序设计

http://www.codece.com/archives/267

一.JS概述

1.历史

(1)1995年，JS最早出现在Netscape公司的浏览器中

(2)1996年，在IE浏览器中也开始出现了JS

(3)1997年，ECMA组织制定了JS的标准规范ECMAScript

(4)2009年，JS开始向后端发展

2.现状

JS既可以运行在客户端浏览器中，也可以运行在服务器端

3.特点

是解释型语言，编译一行执行一行

基于对象

弱类型语言

跨平台

4.应用场景

开发浏览器端的交互效果，服务器端开发

二.开发环境

1.浏览器

每个浏览器中都有自带的JS解释器

谷歌/edge/火狐/Safari/欧朋

360/2345/搜狗/猎豹....

2.服务器端

Node.js

下载地址：www.nodejs.org

打开命令行cmd

node -v 查看当前Node.js版本号

3.运行JS

**浏览器**

新建01.js和01.html两个文件，把js嵌入到html

|  |
| --- |
| <script src='01.js'></script> |

**Node.js**

node 拖拽文件 回车

三.JS的语法规范

区分大小写

每行结束的分号可以省略

分为单行注释(//..)和多行注释(/\*..\*/)

四.变量

存储数据的容器

(1)声明变量

var x=1;

使用var关键字声明变量，变量名称为x，然后把1保存进去

(2)命名规则

变量名称可以由字母、数字、下划线、美元符合组成，不能以数字开头，不能使用关键字，不要使用汉字

有语义

email phone user\_name userName yongHuMing

(3)变量的赋值

变量声明后未赋值为undefined(未定义)，变量可以多次赋值，并且赋不同类型的值，这是弱类型语言的特点

(4)一次声明多个变量

var n1=1, n2=2, n3;

多个变量之间用逗号隔开

课后任务

(1)复习今天内容，整理思维导图

(2)练习：声明多个变量，分别保存商品的单价和数量，计算出商品的总价并输出

练习：交换两个变量的值

a=1 b=2 交换后 a=2 b=1

(3)预习js第2天

http://www.codece.com/archives/190