复习

复杂查询

1.SQL语句中复杂查询分为 —— 聚合查询/分组查询

示例：查询出所有员工的数量

SELECT COUNT(\*) FROM emp; #推荐写法

SELECT COUNT(ename) FROM emp;

SELECT COUNT(eid) FROM emp;

SELECT COUNT(deptId) FROM emp;

|  |
| --- |
| 函数： FUNCTION，具有特定的功能，需要接收若干个数据，最终到处特定的计算结果。—— 饺子机 |

MySQL中提供了5个聚合函数

COUNT(...)/SUM(...)/AVG(..)/MAX(...)/MIN(...)

练习：查询出10号部门员工的数量

SELECT COUNT(ename) FROM emp WHERE deptId=10;

练习：查询出10号部门所有工资的总和

SELECT SUM(salary) FROM emp WHERE deptId=10;

练习：查询出所有男员工的平均工资

SELECT SUM(salary)/COUNT(\*) FROM emp WHERE sex=1;

SELECT AVG(salary) FROM emp WHERE sex=1;

练习：查询出1990年出生的员工工资最大值和最小值

SELECT MAX(salary),MIN(salary) FROM emp WHERE birthday>='1990-1-1' AND birthday<='1990-12-31';

**分组查询**

练习：查询出每个部门的员工数量

SELECT deptId,COUNT(\*) FROM emp GROUP BY deptId;

|  |
| --- |
| 注意：分组查询中SELECT后面只能跟分组条件，或者其它列的聚合函数； |

练习：查询出每个部门的平均工资，最大工资，最小工资

SELECT deptId,AVG(salary),MAX(salary),MIN(salary) FROM emp GROUP BY deptId;

练习：查询出男员工和女员工的人数是多少

SELECT sex,COUNT(\*) FROM emp GROUP BY sex;

2.复杂查询 —— 子查询

示例：查询出工资比tom高的员工的所有信息

步骤1：先查询出tom的工资是多少 —— 6000

SELECT salary FROM emp WHERE ename='tom';

步骤2：再查询出工资比6000高的员工信息

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“研发部”所有的员工信息

步骤1：查询出“研发部”的部门编号 ——10

SELECT did FROM dept WHERE dname='研发部';

步骤2：查询出10号部门所有的员工信息

SELECT \* FROM emp WHERE deptId=10;

总和：

SELECT \* FROM emp WHERE deptId=(SELECT did FROM dept WHERE dname='研发部');

3.复杂查询 —— 多表查询/跨表查询

示例：查询出所有的员工姓名及其所在部门的名称

SELECT ename,dname FROM emp,dept WHERE deptId=did ;

笛卡尔积

|  |
| --- |
| 注意：为了防止出现笛卡尔积，在多表查询的时候，需要添加查询条件 |

问题：无法查询出没有部门的员工，无法查询出没有员工的部门。

**SQL99中的多表查询**

**(1)内连接 INNER JOIN ... ON** —— 和之前的作用一样

SELECT ename,dname FROM emp **INNER JOIN** dept **ON** deptId=did;

**(2)左外连接 LEFT OUTER JOIN** ... **ON** —— 显示左侧所有的记录

SELECT ename,dname FROM emp **LEFT OUTER JOIN** dept ON deptId=did;

**(3)右外连接 RIGHT OUTER JOIN...ON** 显示右侧所有的记录

SELECT ename,dname FROM emp **RIGHT OUTER JOIN** dept **ON** deptId=did;

说明：outer关键字可以省略的。

**(4)全连接 FULL JOIN**

MySQL不支持全支持

**UNION / UNION ALL**

union all 两组数据出现了相同项**不合并。**

union 两组数据出现了相同项进行**合并**。

(SELECT ename,dname FROM emp LEFT JOIN dept ON deptId=did)

UNION ALL

(SELECT ename,dname FROM emp RIGHT JOIN dept ON deptId=did);

(SELECT ename,dname FROM emp LEFT JOIN dept ON deptId=did)

UNION

(SELECT ename,dname FROM emp RIGHT JOIN dept ON deptId=did);

学习一门编程语言路线图

(1)了解语言的背景，历史，特点，应用领域，现状

(2)搭建开发环境，编写 hello world

(3)声明变量和常量

(4)数据类型

(5)运算符

(6)逻辑结构

(7)通用小程序

(8)函数和对象

(9)第三方的类库，插件，组件，框架

(10)开发项目

1. JS的执行环境

(1)运行在客户端，使用浏览器所自带的JavaScript解释器。本身由于浏览器过多，存在兼容性问题。

(2)运行在服务器端，使用NodeJS解释器，基于谷歌的V8引擎。在服务器端不存在兼容性。

2.Nodejs下载

https://nodejs.org 下载 8.11.3 LTS

查看nodejs是否安装成功

**node -v**

显示所安装的nodeJS版本号

运行方式

node c:/xampp/..../01.js

练习：

创建02.js，打印个人的姓名，使用nodejs运行，查看结果。

3.语法规范

(1)区分大小写

(2)每一行结束，可以不加英文的分号结束，习惯上加上分号。

(3)分为单行注释( //... )，和多行注释( /\*...\*/ )

4.变量

变量是用于存储信息的容器

(1)变量的声明

var x = 1;

var是声明变量的关键字；

x是变量的名称

x=1表示把1赋值给变量x

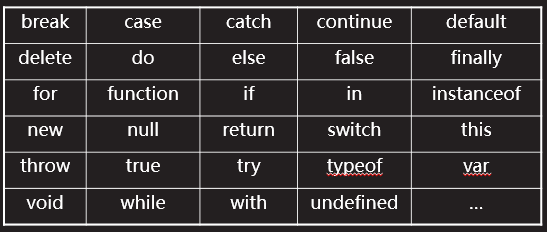
说明：如果声明了一个变量，没有赋值，默认是undefined。

(2)变量的命名规范

可以放字母、数字、下划线(\_)、美元符号($)

不能以数字开头

不能使用关键字和保留字作为变量名



保留字(class、int、float)

变量名称习惯上有明确的语义，例如用户名 userName，密码userPwd，命名分为驼峰命名法和下划线命名法。

练习：

使用变量来存储员工的信息，声明员工的编号，姓名，性别，生日，工资，并初始化具体的信息。

5.常量

常量就是不能改变的值。

使用 **const**关键字来声明

习惯上常量名称使用大写 const PI=3.14;

6.数据类型

MySQL：数值/日期时间/字符串

**JS分为原始类型和引用类型**

(1)原始类型

分为数值型、字符串型、布尔型、未定义型(undefined)、null

数值型又分为整数和浮点数

1)整数型

分为八进制(以0开头->010)，十进制，十六进制(以0X开头->0XF)

2)浮点型

使用小数点记录的数据，还可以使用指数记录的数据

例如：1.2 3.4 1.888e3

练习：

查看整数20,32,54的八进制，十六进制分别是多少

使用小数点和指数来表示圆周率 3.1415926

课后练习:

计算圆的周长和面积：

把半径值5使用变量存储，把圆周率使用常量存储。

最后打印周长和面积。