复习

练习

查询出工资在8000以上的女员工的姓名、性别、工资，结果集按照工资降序排列，获取前两个。

SELECT ename,sex,salary FROM emp

WHERE salary>=8000 AND sex=0

ORDER BY salary DESC

LIMIT 0,2;

性别 1/0

男/女 man/woman 男士/女士

日期时间

2019-7-4

7月4日 七月四日 7/4/2019

存储的是距离计算机元年的毫秒数

1970-1-1 0:0:0 1秒=1000毫秒

49\*365\*24\*60\*60\*1000

1.复杂查询

(1)聚合查询/分组查询

示例: 查询所有员工的数量

SELECT COUNT(ename) FROM emp;

练习: 查询女员工的数量，使用编号这一列

SELECT COUNT(eid) FROM emp WHERE sex=0;

练习: 查询男员工的数量，使用部门编号这一列

SELECT COUNT(deptId) FROM emp WHERE sex=1;

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

|  |
| --- |
| 聚合函数  函数:是一个功能体，提供若干个数据，产出特定的结果——饺子机  COUNT()/SUM()/AVG()/MAX()/MIN() |

练习: 查询所有员工的工资总和

SELECT SUM(salary) FROM emp;

练习: 查询出男员工的平均工资

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

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

练习: 查询出工资最高的员工

SELECT MAX(salary) FROM emp;

练习: 查询出工资最低的员工

SELECT MIN(salary) FROM emp;

练习: 查询出年龄最大的员工生日

SELECT MIN(birthday) FROM emp;

|  |
| --- |
| 分组查询：只能查询分组条件和聚合函数 |

示例：查询每个部门的员工最高工资

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

练习: 查询男女员工的数量，最高工资，平均工资各是多少

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

|  |
| --- |
| YEAR() 获取日期中的年份  MONTH() 获取日期中的月份 |

练习: 查询出1991年出生的员工有哪些

SELECT \* FROM emp WHERE YEAR(birthday)=1991;

练习: 查询出3月份生日的员工有哪些

SELECT \* FROM emp WHERE MONTH(birthday)=3;

(2)子查询

示例: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的员工有哪些

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的出生年份 1990

SELECT YEAR(birthday) FROM emp WHERE ename='tom';

步骤2: 查询出1990年出生的员工有哪些

SELECT \* FROM emp WHERE YEAR(birthday)=1990;

综合：

SELECT \* FROM emp WHERE YEAR(birthday)=(

SELECT YEAR(birthday) FROM emp WHERE ename='tom'

);

练习: 查询出研发部所有的员工

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

(3)多表查询

示例: 查询出所有的员工姓名及其部门名称

SELECT ename,dname FROM emp,dept;

产生笛卡尔积

解决：添加查询条件

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