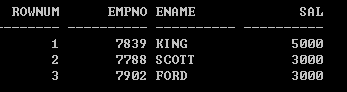
【课堂练习提示】

【第一题】： 找到员工表中工资最高的前三名， 要求按如下格式输出



——涉及Top-N分析问题。

8. 一般不在子查询中使用order by, 但在Top-N分析问题中，必须使用order by

补充知识：rownum 行号(伪列)

SQL> select rownum, empno, ename, sal from emp

借助行号将薪水降序排列。前三条即是我们想要的内容。

SQL> select \* from emp order by sal desc 但问题是如何取出前三行。

SQL> select \* from emp where rownum <= 3 order by sal 发现取出的结果不正确。

行号rownum需要注意的问题：

1. rownum永远按照默认的顺序生成。

SQL> select rownum, empno, ename, sal from emp order by sal desc

——发现行号是跟着行走的。查询结果顺序变了，行号依然固定在原来的行上。

行号始终使用默认顺序：select \* from emp所得到的顺序，没有排序，没有分组等。

只要能使行号随着重新排序，发生改变，那么取前三条记录，就是我们想要的结果。

1. rownum只能使用<, <=符号，不能使用>，>=符号。

想将现有的表进行分页。1-4第一页，5-8第二页……

SQL> select rownum, empno, ename, sal from emp where rownum >=1 and rownum<=4

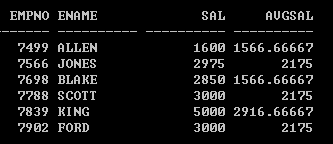
SQL> select rownum, empno, ename, sal from emp where rownum >=5 and rownum<=8

执行，发现结果：未选定行。原因是rownum不能使用>=符号。Where永远为假。

与行号生成的机制有关：Oracle中的行号永远从1开始——取了1才能取2，取了2才能取3，……

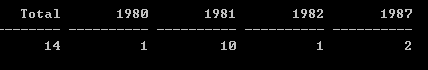
<=8可以是因为1234567挨着取到，而>=5不行，因为没有1234，不能直接取5。

【第二题提示】： 找到emp表中薪水大于本部门平均薪水的员工



几乎涵盖了所学的SQL所有内容（分组、函数、多表查询、子查询），除了order by和Hving外。

【第三题提示】： 统计每年入职的员工个数



——员工的入职年份是已知条件——1980、1981、1982、1987这4个。

要统计每年入职的人数，一定要知道每个员工的入职日期，可以通过查询hiredate列来得到。

SQL> select hiredate from emp;

结合查询结果，以1981年为例，如何统计出81年入职的有多少个人呢？可以从写C程序的角度入手。

思路：定义一个计数器count=0; 有一个81年的员工，就+1，不是81的就+0;最后查看count的值就可以了。

求和，使用sum函数，内部逻辑：sum(if 是81年 then +1 else +0)

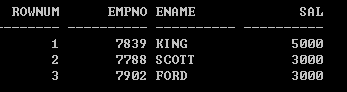
也就是取员工的hiredate列的“年”那一部分，与81比较，进行判断。

to\_char(hiredate, ‘yyyy’) 得到年，与‘1981’进行比较。

提示：该练习只考察函数的应用，注意调用关系。

【答案及相关知识学习分析】

【第一题】： 找到员工表中工资最高的前三名，按如下格式输出



解决该问题，select \* from emp order by sal desc 这条查询必须要有，但是排序后行号rownums没有改变，如果可以改变那么问题就可以解决。

select \* from emp order by sal desc这条查询结果本身是一个集合， 特点1：emp表所有数据均有，

特点2：已经按薪水降序排列好。

把这条语句的执行结果当成一个表来看待即可。——可以将它放置到“from”后面。

SQL> select rownum, empno, ename, sal

from (select \*

from emp

order by sal desc)

where rownum <=3

注：语句中所有的rownum所代表的行号是from后面子查询查出来的新行号，而不是emp表默认的行号。

—— rownum相关知识：—— Oracle中的【分页】—— 题设：1-4第一页、5-8第二页

只需要将上条语句改成“where rownum <=8”上限即可获得，

SQL> select rownum, empno, ename, sal

from (select \*

from emp

order by sal desc)

where rownum <=8

关键是 >=5 的获取问题，是不能直接写>=5的。但是这条SQL语句查询的结果是一个新的“集合”，

该集合中有一**列**是专门表示行号，**可以假想这列不再是伪列**，而是该集合中专门用来表示行号的列。

所以，可以给该列取一个别名r，并把该查询语句整体作为子查询，放到另外一条SQL语句的from后。

SQL> select \*

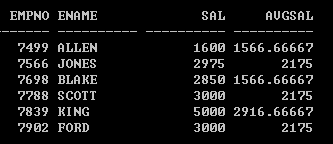
from ( select rownum r, empno, ename, sal

from (select \* from emp order by sal desc)

where rownum <=8 )

where r >=5;

【第二题】： 找到员工表中薪水大于本部门平均薪水的员工



首先，对于每个员工，应该得到自己部门的平均薪水：

SQL> select deptno, avg(sal) from emp group by deptno;

其次，要从员工表中找员工。两张表 ↓

from emp e, (select deptno, avg(sal) from emp group by deptno) d

筛选出 e.deptno=d.deptno 得到本部门的员工，且薪水大于平均薪水。 ↓

from emp e, (select deptno, avg(sal) avgsal from emp group by deptno) d

where e.deptno=d.deptno and e.sal > d.avgsal

最终，显示要查询的列empno，ename，sal，avgsal。 ↓

SQL> select e.empno, e.ename, e.sal, d.avgsal

from emp e, (select deptno, avg(sal) avgsal from emp group by deptno) d

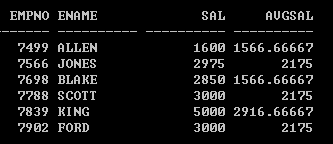
where e.deptno=d.deptno and e.sal > d.avgsal;

这个练习中包含：组函数、分组、多表查询、子查询。

——相关知识：——【相关子查询】——

7. 一般先执行子查询(内查询)，再执行主查询(外查询)；但是相关子查询除外。

相关子查询概念：**将主查询的某个值，作为参数传递给子查询**。



以第二题为例：

找到员工表中薪水大于本部门平均薪水的员工信息

首先，根据题意要求查员工号、姓名、薪水，其他的先不理会。

select empno, ename, sal from emp;

其次，员工的薪水应该大于自己所在部门的平均薪水。这时在子查询中要使用 ↓

到主查询的部门号。主查询通过别名，将部门号传递给子查询。 ↓

select empno, ename, sal from emp e

where sal > ( select avg(sal)

from emp

where deptno=e.deptno);

这种查询就叫做“相关子查询”，因为在子查询中引用主查询的值，主查询的值作为参数传递给子查询。

↓

第三，同理可以将平均薪水，用相同的方法得到。 ↓

SQL> select empno, ename, sal, (select avg(sal) from emp where deptno=e.deptno) avgsal

from emp e

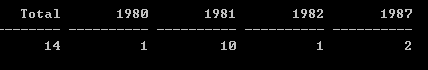
where sal > (select avg(sal )

from emp

where deptno=e.deptno);

总结：在这条SQL语句中有两个子查询，一个在select后，一个在where后。这两个子查询都是相关子查询。主查询通过员工表的别名e，将员工号分别传递给两个子查询。

【第三题】： 统计每年入职的员工个数



根据提示，主要使用sum来完成对1和0的求和操作。sum函数内部，是一个if、else判断。实现是使用decode或者case。由于sum是函数，我们同样用函数decode来实现。

——decode()：第一个参数：待判断的值，条件，结果，条件，结果……最后是else的情况。

to\_char(hiredate, ‘yyyy’) 得到年，与‘1981’进行比较。

——sum(decode(to\_char(hiredate, ‘yyyy’), ‘1981’, 1, 0)) 可求出1981年入职的员工总数。其他各年类似

SQL> select sum(decode(to\_char(hiredate, 'yyyy'), '1981', 1, 0)) "1981", 注意：字符串是数字，必须加“”

sum(decode(to\_char(hiredate, 'yyyy'), '1982', 1, 0)) "1982",

sum(decode(to\_char(hiredate, 'yyyy'), '1980', 1, 0)) "1980",

sum(decode(to\_char(hiredate, 'yyyy'), '1987', 1, 0)) "1987",

from emp; ↓

↓

SQL> select count(\*) Total,

sum(decode(to\_char(hiredate, 'yyyy'), '1981', 1, 0)) "1981",

sum(decode(to\_char(hiredate, 'yyyy'), '1980', 1, 0)) "1980",

sum(decode(to\_char(hiredate, 'yyyy'), '1982', 1, 0)) "1982",

sum(decode(to\_char(hiredate, 'yyyy'), '1987', 1, 0)) "1987"

from emp;

如忽略输出格式，可用分组：SQL> select to\_char(hiredate ,'yyyy'), count(\*) from emp group by to\_char(hiredate ,'yyyy')