1-编写基本的SQL

2017年1月21日 19:17

1.判断下面的 SELECT 语句是是否能执行成功

```
SELECT last_name,job,salary as Sal FROM employees;
desc employees;
```

2.判断下面的 SELECT 语句是否能执行成功

```
SELECT * FROM job_grades;
```

3.下面语句中，有几处错误

```
SELECT     employee_id,last_name,
sal x 12       ANNUAL SALARY
FROM           employees;
```

4.显示DEPARTMENTS的表结构，并从该表中选择所有的数据

```
desc departments;
select * from departments;
```

5.显示EMPLOYEES的表结构，并创建一个查询，显示每位员工的姓氏、职务代码、聘用日期和员工编号，并且首先显示员工编号，
为HIRE_DATE提供一个列别名STARTDATE，将你的SQL语句保存到lab1_5.sql中。

```
desc employees;
select employee_id,last_name,job_id,hire_date startdate from employees;
```

6.执行第5题中，保存的脚本

```
@/home/oracle/lab1_5.sql
```

7.创建一个查询，以显示EMPLOYEES表中唯一的职务代码。

```
select distinct job_id from employees;
```

8.第五题中的SQL语句，为需要查询的列标题设置别名为”Emp #”、”Employess”、”Job”和”Hire Date”，并再次运行你的查询

```
select employee_id "Emp #",last_name "Employees",job_id "Job",hire_date "Hire Date" from employees;
```

9.显示姓氏并连接该雇员的职务，它们之间同逗号和空格做分隔，然后为该输出的列设置别名”Employee and Title”

```
select last_name||', '||job_id "Employee and Title" from employees;
```

10.创建一个查询，显示EMPLOYEES表中所有的数据，用逗号分隔输出的所有列，并设置别名“THE_OUTPUT”

```
select employee_id||','||last_name||','||first_name||','||job_id||','||salary||','||department_id from employees;
```

11.把所有字段拼成一句有意义的话

```
select last_name||'的工资是'||salary from employees;
```

2-限制和排序

2017年1月21日 19:20

1.创建一个查询，显示工资超过12000的员工的姓氏和工资，将你的SQL语句保存到lab2_1.sql文件中，并使用该脚本进行查询。

```
select last_name,salary from employees where salary>12000;
```

2.创建一个查询，显示员工号176的姓氏和部门编号

```
select last_name,department_id from employees where employee_id=176;
```

3.修改lab2_1.sql，使其显示工资不在5000至12000范围内的所有员工的姓氏和工资，并将该修改后的脚本另存为lab2_3.sql，

使用脚本进行查询。

```
select last_name,salary from employees
where salary not between 5000 and 12000;
```

4.显示在2003年2月20日至2007年5月1日之间录用的员工姓氏、职务和录用日期，并按录用日期进行升序排序。

```
select last_name,job_id,hire_date
from employees
where hire_date between '20-2月-03' and '01-5月-07'
order by hire_date;
select last_name,job_id,to_char(hire_date,'yyyy-mm-dd')
from employees
where hire_date between to_date('2003-02-20','yyyy-mm-dd') and to_date('2007-05-01','yyyy-mm-dd')
order by hire_date;
```

5.按姓名的字母顺序显示部门20和部门90的所有员式的姓氏和部门编号。

```
select last_name,department_id from employees
where department_id in(20,90)
order by 1;
```

6.修改lab2_3.sql，使其列出工资在5000到12000之间，并且部门是20或者50的员工姓氏和工资，分别将列标记为“Employee”和

“Monthly Salary”。并将该修改过的脚本另存为lab2_6.sql。使用该脚本执行你的查询。

```
select last_name "Employee",salary "Monthly Salary" from employees
where salary between 5000 and 12000
and department_id in (20,50);
```

7.显示在2004年录用的每位员工的姓氏和录用日期

```
select last_name,hire_Date
from employees
where hire_date between '01-1月-04' and '31-12月-04' ;

select last_name,hire_Date
from employees
where to_char(hire_date,'yyyy')=2004;
```

8.显示没有经理的所有员工的姓氏和职称

```
select last_name,job_id from employees
where manager_id is null;
```

9.显示有奖金可拿的所有员工的姓氏、工资和奖金提成比率，并按工资和奖金提成比率进行降序排序。

```
select last_name,salary,commission_pct from employees
where commission_pct is not null
order by 2 desc,3 desc;
```

10.显示员工姓氏中第三个字母为“a”的所有员工的姓名和姓氏。

```
select last_name||first_name,last_name
from employees
where last_name like '__a%';
```

11.显示员工姓氏中有“a”和“e”的所有员工的姓氏和姓名。

```
select last_name from employees
where last_name like '%a%'
and last_name like '%e%';
```

12.显示职务为sa_rep和st_clerk，且工资不等于2500，3500和7000的所有员工的姓氏，姓名、职务和工资。

```
select last_name,last_name||first_name,job_id,salary
from employees
```

```
where salary not in (2500,3500,7000)
and job_id in ('SA_REP','ST_CLERK');
```

13.修改lab2_6.sql，使其显示奖金提成为20%的所有员工的姓氏、姓名、工资和奖金。将该修改后的脚本另存为lab2_13.sql，
并使用该脚本执行该查询。

```
select last_name "Employee",salary "Monthly Salary" from employees
where salary between 5000 and 12000
and commission_pct =0.2;
```

3、4-单行函数、转换函数和条件表达式

2017年1月21日 19:21

1.编写一个查询，使其显示当前日期，将列名命名为Date。

```
select to_char(sysdate,'yyyy/mm/dd hh24:mi:ss') from dual;
```

2.显示每位员工的编号，姓氏，薪金和增加15%薪金之后薪金值（并将该列命名为New Salary）不允许输出的结果中有空值。

```
select employee_id,last_name,
salary*(1+nvl(commission_pct,0)),
salary*(1+nvl(commission_pct,0)+0.15) "New Salary"
from employees;
```

3.显示每位员工的编号，姓氏，薪金和增加15%薪金之后薪金值（并将该列命名为New Salary），添加一个列，命名为Increase，该列是从增加了15%工资以后的列，减去原有的工资，即工资实际涨了多少。同样要求，该查询结果中，不允许有空值。

```
select employee_id,last_name,salary*(1+nvl(commission_pct,0)) "Old Salary",
salary*(1+nvl(commission_pct,0)+0.15) "New Salary",
salary*(1+nvl(commission_pct,0))*0.15 "Increase"
from employees;
```

4.编写一个查询，显示姓氏以J、A或M开始的所有员工的姓氏，要求第一个字母大写，所有其它字母小写，并显示姓名的长度。

```
select initcap(last_name),length(last_name)
from employees
where upper(substr(last_name,1,1)) in ('J','A','M');
```

5.显示每位员工的姓氏，并计算今天和员工入职日期之间的月数。将该列命名为MONTHS_WORKED。按入职的月数进行排序。输出结果要求是进行四舍五入后的整数结果。

```
select last_name,round(months_between(sysdate,hire_date)) "MONTHS_WORKED" from employees;
```

6.编写一个查询，为每个员工产生如下内容

<员工姓氏> 现在的薪水是 <工资> 每月 他期望他能拿到每月<3倍工资>。 将该列命名为 Dream Salaries。

例如：

King 现在的薪水是 $24,000.00 每月 他期望他能拿到每月$72,000.00。

```
select last_name ||'现在的薪水是'
||to_char(salary,'$99,999.00')
||'每月 他期望他能拿到每月'
||to_char(salary*3,'$99,999.00')
||'。'
from employees;
```

7.创建一个查询，显示所有员工的姓氏和薪金。要求，姓氏要大写，将薪金格式规定为15个字符长，左边填充$，将该列命名为SALARY例如：

KING $$$$$$$$$$24000

```
select last_name,lpad(salary*(1+nvl(commission_pct,0)),15,'$')
from employees;
```

8.显示每位员工的姓氏、入职日期和薪金复核日期，薪金复核日期是入职六个月的第一个星期一进行。将该列命名为REVIEW。

规定这一日期格式，使其显示样式类似于”Monday,the Thirty-First of July,2000”。

```
alter session set nls_language=american ;
select last_name,hire_date,
to_char(next_day(add_months(hire_date,6),'MONDAY'),'fmDay,"the "Ddspth" of "Month,yyyy') "REVIEW"
from employees;
```

9.显示员工的姓氏、入职日期和该员工是在星期几开始工作的。将该列命名为DAY。按星期中各天的顺序（从星期一开始）将结果排序。

```
select last_name,hire_date,to_char(hire_date,'DAY') "DAY"
from employees
order by to_char(hire_date-1,'d');

select last_name,hire_date,to_char(hire_date,'DAY') "DAY"
from employees
order by
case to_char(hire_date,'fmDAY')
```

```
when 'MONDAY' then 1
when 'TUESDAY' then 2
when 'WEDNESDAY' then 3
when 'THURSDAY' then 4
when 'FRIDAY' then 5
when 'SATURDAY' then 6
when 'SUNDAY' then 7 end;
```

10.创建一个查询，使其显示员工的姓氏和奖金比率。如果某员工没有资金，则显示“No Commission”，将该列命名为COMM。

```
select last_name,nvl(to_char(commission_pct,'0.99'),'No Commission') "COMM"
from employees;
```

11.创建一个查询，使其显示员工的姓氏，并用星号指明他们的薪金。每个星号代表一千元，有几千就有几个星号，取整，不足一千的不参与统计。按薪金降序排序。将该列命名为EMPLOYEES_AND_THEIR_SALARIES。

类似：

```
King**********
select rpad(last_name,length(last_name)+trunc(salary*(1+nvl(commission_pct,0))/1000),'*')
from employees
order by salary*(1+nvl(commission_pct,0)) desc
```

12.使用DECODE函数编写一个查询，使其按照以下数据根据JOB_ID列的值显示所有员工的级别：

职务(Job)	级别(Grade)
AD_PRES	A
ST_MAN	B
IT_PROG	C
SA_REP	D
ST_CLERK	E
所有上述都不是	0

```
select last_name,job_id,
decode(job_id,'AD_PRES','A',
              'ST_MAN','B',
              'IT_PROG','C',
              'SA_REP','D',
              'ST_CLERK','E',
              '0') grade
from employees;
```

13.用CASE语法，实现上题的要求。

```
select last_name,job_id,
case job_id
when 'AD_PRES' then 'A'
when 'ST_MAN'  then 'B'
when 'IT_PROG' then'C'
when 'SA_REP'  then 'D'
when 'ST_CLERK' then'E'
else '0'  end grade
from employees;
```

5-分组函数

2017年1月21日 19:22

1.判断

分组函数通过处理多个行来为每个组生成一个结果。

2.判断

分组函数可以计算空值（count(*)）

3.判断

WHERE子句在包含到分组计算之前，可以对行进行限制

4.显示所有员工的最高、最低、总计和平均工资。分别将各列标记为Maximum、Minimum、Sum和Average。
将结果舍入到最接近的整数并按照平均工资进行升序排序。

```
select max(salary),min(salary),sum(salary),trunc(avg(salary))
from employees
order by 4;
```

5.显示每个职务类型的最低、最高、总计和平均工资，将结果舍入到最接近的整数，列的命名方式与上题相同
并按照职务类型进行升序排序。

```
select job_id,max(salary),min(salary),sum(salary),trunc(avg(salary))
from employees
group by job_id
order by 1;
```

6.显示职务相同的员工人数。

```
select job_id,count(*) from employees group by job_id;
```

7.确定经理的人数而不列出他们，将该列标记为Number of Managers。

```
select count(distinct manager_id) from employees;
```

8.编写一个查询，显示最高工资和最低工资之间的差额，将该列标记为DIFFERENCE。

```
select max(salary)-min(salary) from employees;
```

9.显示经理编号以及该经理所管员工的最低工资，不包括其经理未知的任何员工。排除最低工资不超过6000的所有组。
按工资降序进行排序。

```
select manager_id,min(salary)
from employees
where manager_id is not null
group by manager_id
having min(salary)>6000
order by 2 desc;
```

10.编写一个查询，显示每个部门编号，部门员工人数，和该部门的平均工资。
将各列命令为Deptno、Number of People、Avg of Salary。平均工资舍入到小数后两位，
排除掉未知部门，结果按部门编号进行排序。

```
select department_id,count(*),trunc(avg(salary),2)
from employees
where department_id is not null
group by department_id
order by 1;
```

11.创建一个查询，显示员工总数，以及其中在2003、2005、2006年入职的员工数。
为每列创建标题为Total、2003、2005、2006。例如输出如下：

```
TOTAL          2003        2005        2007
107               6          29          19
```

至少写出3种方法

```
select count(*) "TOTAL",
sum(
case to_char(hire_date,'yyyy')
when '2003' then 1
end) "2003",
sum(
case to_char(hire_date,'yyyy')
when '2005' then 1
end) "2005",
```

```
sum(
case to_char(hire_date,'yyyy')
when '2006' then 1
end) "2006"
from employees;
select
(select count(*) from employees) "TOTAL",
(select count(*) from employees where to_char(hire_Date,'yyyy')=2003) "2003",
(select count(*) from employees where to_char(hire_Date,'yyyy')=2005) "2005",
(select count(*) from employees where to_char(hire_Date,'yyyy')=2006) "2006"
from dual;
```

12.创建一个矩阵查询，使其显示所有的职务类型、部门20、50、80和90部门基于部门编号的职务工资总和，以及所有职务的工资总和。

使其输出为

```
select job_id job,
sum(
case department_id
when 20 then salary
end) "Detp20",
sum(
case department_id
when 50 then salary
end) "Detp50",
sum(
case department_id
when 80 then salary
end) "Detp80",
sum(
case department_id
when 90 then salary
end) "Detp90",
sum(salary) total
from employees
group by job_id
order by 1;
```

6-多表查询

2017年1月21日 19:23

1.编写一个查询，以显示所有员工的姓氏、部门编号和部门名称，要求使用自然联结和传统语法两种方式实现。

```
select last_name,department_id,department_name
from employees join departments using(department_id);
select a.last_name,a.department_id,b.department_name
from employees a,departments b
where a.department_id=b.department_id;
```

2.创建部门80中所有职务的唯一列表，并列出部门的地点。要求至少写出两种分别使用自然联结和传统语法的语句。

```
select distinct job_id,city
from employees join departments using(department_id)
join locations using(location_id)
where department_id=80;
select distinct a.job_id,c.city
from employees a,departments b,locations c
where a.department_id=b.department_id
and b.location_id=c.location_id
and a.department_id=80;
```

3.编写一个查询，以显示赚取奖金的所有员工的姓氏、部门名称、城市代码和城市名称。要求同上

```
select last_name,department_name,location_id,city
from employees join departments using(department_id)
join locations using(location_id)
where commission_pct is not null;
```

4.显示姓氏中含有小写字母a的所有员工姓氏和部门名称。要求同上

```
select last_name,department_name
from employees join departments using(department_id)
where last_name like '%a%';
```

5.编写一个查询，以显示在Toronto工作的所有员工的姓氏、职务、部门编写和部门名称。要求同上

```
select last_name,job_id,department_id,department_name
from employees join departments using(department_id)
join locations using(location_id)
where city='Toronto';
```

6.显示员工姓氏、员工编号以及他们经理的姓氏和经理编号。将这些列分别标记为Employee、Emp#、Manager和Mgr#。要求同上

```
select a.last_name "Employee",a.employee_id "Emp#",
b.last_name "Manager",b.employee_id "Mgr#"
from employees a join employees b
on (a.manager_id=b.employee_id);
```

7.在上题基础上，使其显示中包含没有上级经理的员工King。要求同上。

```
select a.last_name "Employee",a.employee_id "Emp#",
b.last_name "Manager",b.employee_id "Mgr#"
from employees a left outer join employees b
on (a.manager_id=b.employee_id);
```

8.创建一个查询，以显示员工的姓氏、部门编号以及与该员工在同一部门一起工作的有多少人？要求同上

```
select a.last_name,a.department_id,count(b.employee_id)-1
from employees a join employees b
on(a.department_id=b.department_id)
group by a.last_name,a.department_id
order by 2,1
```

9.使用数据库用户hr，执行下发的脚本create_job_grades.sql，显示JOB_GRADE表的结构，创建一个查询，显示所有员工姓名、职务、部门名称、工作地点以及工资级别。要求同上

```
select a.last_name||a.first_name,b.department_name,c.city,d.grade
from employees a join departments b on(a.department_id=b.department_id)
join locations c on(b.location_id=c.location_id)
```

```
join job_grade d on(a.salary between d.low_sal and high_sal);
```
10.创建一个查询，显示在全名叫KimberelyGrant的员工之后入职的所有员工的姓名和入职日期。要求同上
```
select hire_date from employees where first_name||last_name='KimberelyGrant';
select last_name,hire_Date from employees where hire_Date>'24-MAY-07';
select b.last_name,b.hire_Date
from employees a join employees b
on (b.hire_Date>a.hire_date)
where a.first_name||a.last_name='KimberelyGrant';
```
11.显示在其经理之前入职的所有员工的全名和入职日期，以及其经理的全名和入职日期。列名分别为Employee、Emp Hired、Manager和Mgr hired
要求同上。
```
select a.last_name||a.first_name "Employee",a.hire_Date "Emp Hired",b.last_name
||b.first_name "Manager",b.hire_date "Mgr hired"
from employees a join employees b
on (a.manager_id=b.employee_id)
and a.hire_date<b.hire_date;
```

7-子查询

2017年1月21日 19:23

1.编写一个查询，显示和Zlotkey在同一个部门的所有员工的姓氏和入职日期，但不包括Zlotkey

```
select last_name,hire_date
from employees
where department_id=(
                    select department_id
                    from employees
                    where last_name='Zlotkey')
and last_name <> 'Zlotkey';
```

2.创建一个查询，显示年薪（工资*12加奖金）超过平均年薪的所有员工的员工编号、姓氏、部门名称和部门所在地。

```
select employee_id,last_name,department_name,city
from employees join departments using(department_id)
join locations using(location_id)
where salary*12*(1+nvl(commission_pct,0))>(
                                        select avg(salary*12*(1+nvl(commission_pct,0)))
                                        from employees);
```

3.编写一个查询，显示所有员工的员工编号，姓氏，部门ID，条件是他们所工作的部门里有员工姓氏包含且只包含一个u，输出结果里，排除这些姓氏里包含且只包含一个u的结果。

```
select employee_id,last_name,department_id
from employees
where department_id in(
                      select department_id
                      from employees
                      where instr(last_name,'u',1,1)>0
                      and instr(last_name,'u',1,2)=0)
and (last_name not like '%u%'
or instr(last_name,'u',1,2)>0);
```

4.显示部门的location ID为1700的所有员工姓氏、部门编号和职务。

```
select last_name,department_id,job_id
from employees join departments using(department_id)
where location_id=1700;
select last_name,department_id,job_id
from employees
where department_id in(
                      select department_id
                      from departments
                      where location_id=1700);
```

5.显示King的每个下属员工的姓氏、工资、经理ID和经理姓氏。

```
select a.last_name,a.salary,a.manager_id,b.last_name
from employees a join employees b
on(a.manager_id=b.employee_id)
where b.last_name='King';
select a.last_name,a.salary,a.manager_id,b.last_name
from employees a join employees b
on (a.manager_id=b.employee_id)
where a.manager_id in(
                    select employee_id
                    from employees
                    where last_name='King')
```

6.查询部门工作地点在Seattle，且职务与111号员工相同的所有员工的员工编号、姓氏、部门编号和职务。

```
select employee_id,last_name,department_id,job_id
from employees
where department_id in (
                        select department_id
                        from departments join locations
                        using(location_id)
```

```
                        where city='Seattle')
and job_id=(
            select job_id
            from employees
            where employee_id=111)
```

7.查询工资超过平均工资，且部门里有员工姓氏中包含字母u的所有员工的员工编号，姓氏和工资。

```
select employee_id,last_name,salary
from employees
where salary>(
              select avg(salary)
              from employees)
and department_id in(
                    select department_id
                    from employees
                    where last_name like '%u%')
```

8-集合

2017年1月21日 19:23

1.列出不包含职务ST_CLERK的所有部门ID

```
select department_id from employees
minus
select department_id from employees where job_id='ST_CLERK'
```

2.显示没有设立部门的国家代码和国家名称

```
select country_id,country_name
from countries join locations using(country_id)
where location_id in(
select location_id from locations
minus
select location_id from departments);

select country_id,country_name
from countries join locations using(country_id)
where location_id in(
select location_id from locations
minus
select location_id from departments)
minus
select country_id,country_name
from countries join locations using(country_id)
where location_id in(
                    select location_id
                    from departments);
```

3.不使用DISTINCT，输出设立有部门的国家代码

```
select country_id,country_name
from countries join locations using(country_id)
where location_id in(
                    select location_id
                    from departments)
intersect
select country_id,country_name
from countries;
```

4.创建一个查询，显示部门10，50，20部门所包含的职务和部门ID，要求部门显示的顺序以10，50，20显示。

```
select job_id,department_id
from employees
where department_id=10
union all
select job_id,department_id
from employees
where department_id=50
union all
select job_id,department_id
from employees
where department_id=20
```

5.列出哪些员工，从事当前职务之前，还在公司任职过其它职务，输出该员工的ID，以及以前任职的其它职务的ID

```
select employee_id,job_id
from job_history
minus
select employee_id,job_id
```

```
from employees;
```
6.整合以下两个查询，合并成一个输出。
1）显示所有员工的姓氏和部门ID，不管该员工是否有部门。
2）显示所有部门的ID和部门名称，不管该部门是否有员工。
```
select last_name,department_id,to_char(null) department_name from employees
union
select to_char(null),department_id,department_name from departments;
```