

SQL100题

select * from emp;

select * from dept;

-- 1.1 列出至少有两个员工的部门

```
1 SELECT DEPTNO
2 FROM EMP
3 GROUP BY DEPTNO
4 HAVING COUNT(1)>=2;
```

--1.2 查询工资大于或者等于2000 的员工信息

```
1 SELECT *
2 FROM EMP
3 WHERE SAL>=2000;
```

--1.3 查询至少有两个员工的工资不低于2000的部门编号

```
1 SELECT DEPTNO
2 FROM EMP
3 WHERE SAL>=2000
4 GROUP BY DEPTNO
5 HAVING COUNT(1)>=2;
```

--2.3 查询工资 与 SCOTT 相同的员工信息,并且不返回SCOTT自己的信息

```
1 SELECT *
2 FROM EMP
3 WHERE SAL=(SELECT SAL FROM EMP WHERE ENAME='SCOTT');
```

--3. 列出所有员工的姓名及其直接上级的姓名。

```
1 SELECT a.ename AS 员工姓名,b.ename AS 上级姓名
2 FROM emp a
3 LEFT JOIN emp b
4 ON a.mgr=b.empno;
```

-- 4. 列出受雇日期早于其直接上级的所有员工。

```
1 SELECT a.ename AS 员工姓名
2 FROM emp a
3 LEFT JOIN emp b
4 ON a.mgr=b.empno
5 WHERE a.hiredate<b.hiredate;
```

--5. 列出部门名称和这些部门的员工信息，同时列出那些没有员工的部门。

```
1 SELECT *
2 FROM dept a
3 FULL JOIN emp b
4 ON a.deptno=b.deptno;
```

--6. 列出所有“CLERK”（办事员）的姓名及其部门名称。

```
1 SELECT A.ENAME AS 姓名,B.DNAME AS 部门名称
2 FROM EMP A
3 LEFT JOIN DEPT B
4 ON A.DEPTNO=B.DEPTNO
5 WHERE A.JOB='CLERK';
```

--7. 列出最低薪金大于1500的各种工作。

```
1 SELECT JOB
2 FROM EMP
3 GROUP BY JOB
4 HAVING MIN(SAL)>2000;
```

--8. 列出在部门“SALES”（销售部）工作的员工的姓名，假定不知道销售部的部门编号。

```
1 SELECT *
2 FROM EMP A
3 LEFT JOIN DEPT B
4 ON A.DEPTNO=B.DEPTNO
5 WHERE B.DNAME='SALES';
```

--9. 列出薪金高于公司平均薪金的所有员工。

```
1 SELECT *
2 FROM emp
3 WHERE sal>(SELECT AVG(sal) FROM emp);
```

--10. 列出与“SCOTT”从事相同工作的所有员工。

```
1 SELECT *
2 FROM emp
3 WHERE job=(SELECT job FROM emp WHERE ename='SCOTT');
```

--11. 列出薪金等于部门30中员工的薪金的非30号部门的员工的姓名和薪金。

```
1 SELECT *
2 FROM emp
3 WHERE sal IN (SELECT sal FROM emp WHERE deptno=30)
4 AND deptno<>30;
```

--13. 列出在每个部门工作的员工数量、平均工资和平均服务期限。

```
1 SELECT deptno AS 部门名称,COUNT(1) AS 员工数量,AVG(sal) AS 平均工资,AVG(Sysdate-  
hiredate) AS 平均入职时长  
2 FROM emp  
3 GROUP BY deptno;
```

--14. 列出所有员工的姓名、部门名称和工资。

```
1 SELECT deptno,COUNT(1) AS 部门人数  
2 FROM emp  
3 GROUP BY deptno;
```

--15. 列出所有部门的详细信息和部门人数。

```
1 SELECT job AS 岗位,MIN(sal) AS 最低工资  
2 FROM emp  
3 GROUP BY job;
```

--16. 列出各种工作的最低工资。

```
1 SELECT deptno AS 部门名称,MIN(sal) AS 最低工资  
2 FROM emp  
3 WHERE job='MANAGER'  
4 GROUP BY deptno;
```

--17. 列出各个部门的MANAGER（经理）的最低薪金。

```
1 SELECT deptno AS 部门名称,MIN(sal) AS 最低工资  
2 FROM emp  
3 WHERE job='MANAGER'  
4 GROUP BY deptno;
```

--18. 列出所有员工的年工资,按年薪从低到高排序。

```
1 SELECT ename AS 员工姓名,sal+nvl(comm,0) AS 年薪  
2 FROM emp  
3 ORDER BY sal+NVL(comm,0);
```

--23.查询出所有奖金（comm）字段不为空的人员的所有信息。

```
1 SELECT *  
2 FROM emp  
3 WHERE comm IS NOT NULL;
```

--24.查询出薪水在800到2500之间（闭区间）所有员工的信息。

--25.查询出员工号为7521, 7900, 7782的所有员工的信息。

--26.查询出名字中有“A”字符, 并且薪水在1000以上（不包括1000）的所有员工信息。

--27.查询出名字第四个字母是“M”的所有员工信息。

--28.将所有员工按薪水升序排序, 薪水相同的按照入职时间降序排序。

--29.将所有员工按照名字首字母升序排序, 首字母相同的按照薪水降序排序。

--30.查询出最早工作的那个人的名字、入职时间和薪水。（聚合函数使用）

--31.显示所有员工的名字、薪水、奖金，如果没有奖金，暂时显示100.

--32.显示出薪水最高人的职位。

--33.查出emp表中所有部门的最高薪水和最低薪水，部门编号为10的部门不显示。

--34.删除10号部门薪水最高的员工。

--35.将薪水最高的员工的薪水降30%。

--36.查询员工姓名，工资和 工资级别(工资>=3000 为3级，工资>2000 为2级，工资<=2000 为1级)

--37、复制emp表格(只复制表结果不复制内容)

--38、复制emp表格，同时复制emp表内容

--39、对员工表的工资从低到高排序,求累计应发放薪资（分析函数）

--40、按照每个部门的工资从高到低排序,计算每个部门的累计工资（分析函数）

--41、将emp表按工资从高到低排序,并给出排名（排名不并列）（分析函数）

--41、将emp表按工资从高到低排序,并给出排名（排名并列）（分析函数）

--42、按照员工部门分组,给出工资从高到低的排名（排名不并列）（分析函数）

--43、员工表按入职先后排序,求出每两个员工的入职时间差（分析函数）

--44、创建一个emp的普通视图v_emp

--45、删除普通视图v_emp

--46、创建一个emp的物化视图mv_emp，并且指定自动刷新方式为on commit

--47、删除物化视图mv_emp

--48、创建B-树索引idx_emp

--49、查看B-树索引idx_emp

--50、删除索引idx_emp

--51、使用Hint调优方式中的parallel,使并行的数量为4

--52.01创建一张c_emp表，字段包括user_id,user_name,user_age

--52.02给c_emp表设置序列

--52.03创建一张cc_emp表，结构和c_emp表一致

--52.04为cc_emp表设置增量同步,同步的数据来源为c_emp

--52.05删除c_emp和cc_emp表

--53.01创建一张c_emp表，创建一张c_emp表，字段包括user_id(number),user_name(char),user_age(char)

--53.02将c_emp表中的user_name,user_age字段数据类型修改成varchar2(20)

```
1 ALTER TABLE c_emp MODIFY user_name VARCHAR2(20);
2 ALTER TABLE c_emp MODIFY user_age VARCHAR2(20);
```

--53.03删除c_emp表

```
1 DROP TABLE c_emp;
```

--54.01创建c_emp表，数据结构和emp表中的数据一致

```
1 CREATE TABLE c_emp AS SELECT * FROM emp;
```

--54.02删除c_emp表中的重复值（根据deptno字段去重）

```
1 DELETE FROM c_emp
2 WHERE ROWID NOT IN
3 (SELECT MIN(ROWID) FROM c_emp GROUP BY deptno);
```

--54.03清空c_emp表中的数据，之后再删除c_emp表

```
1 TRUNCATE TABLE c_emp;  
2 DROP TABLE c_emp;
```

--55.查询emp表的全部数据，并新增一个由empno和ename拼接而成的新字段

```
1 SELECT e.*,CONCAT(e.empno,e.ename)  
2 FROM emp e;
```

--56.查询emp表的全部数据，并新增一个由将job字段的内容改成小写，再改成大写的字段

```
1 SELECT e.*,LOWER(e.job),UPPER(LOWER(e.job))  
2 FROM emp e;
```

--57.查询emp表的全部数据，并新增一个由将ename字段中所有的“A”替换成“替”的新字段

```
1 SELECT e.*,REPLACE(e.ename,'A','替')  
2 FROM emp e;
```

--58.查询emp表的全部数据，并新增一个由截取ename最后二个字母的新字段

```
1 SELECT e.*,SUBSTR(e.ename,-2,2)  
2 FROM emp e;
```

--59.查询emp表的全部数据，并新增一个由将comm中的空值转换成100的新字段

```
1 SELECT e.*,NVL(comm,100)  
2 FROM emp e;
```

--60.查询emp表的全部数据，并新增一个由将hiredate字段转换成“YYYY”格式

```
1 SELECT e.*,to_char(e.hiredate,'YYYY')  
2 FROM emp e;
```

--61.查看emp表的数据结构

```
1 SELECT * FROM all_tab_columns WHERE table_name='EMP';
```

--62.列出emp表中sal（薪水）与deptno=30部门的员工的薪金相同的所有人信息（使用exists方法）。

```
1 SELECT *  
2 FROM emp a  
3 WHERE EXISTS(SELECT b.sal FROM emp b WHERE a.sal=b.sal AND deptno=30);
```

--63使用伪列的方法查看taobao_user表的前50行数据

```
1 SELECT *
2 FROM taobao_user
3 WHERE ROWNUM<=50;
```

--64使用case when将chengji表行转列（将course课程名转成字段，数据为scores分数）

```
1 SELECT ID,NAME,
2 SUM(CASE WHEN course='语文' THEN score END ) AS 语文,
3 SUM(CASE WHEN course='数学' THEN score END ) AS 数学,
4 SUM(CASE WHEN course='英语' THEN score END ) AS 英语,
5 SUM(CASE WHEN course='历史' THEN score END ) AS 历史,
6 SUM(CASE WHEN course='化学' THEN score END ) AS 化学,
7 SUM(score) AS 总成绩
8 FROM chengji
9 GROUP BY ID,NAME
10 ORDER BY ID;
```

--65创建表e_emp,字段 (id NUMBER(2),name VARCHAR2(20)) , 使用insert all多行插入(1,'xiongyuan'),(2,'xiongbao')两条数据

```
1 INSERT ALL
2 INTO e_emp
3 VALUES
4 (1,'xiongyuan')
5 INTO e_emp
6 VALUES
7 (2,'xiongbao')
8 SELECT * FROM dual;
```

--66创建表c_emp和cc_emp，表结构和emp相同。当job='CLERK'时插入c_emp表，当job='MANAGER'时插入cc_emp表中

```
1 INSERT ALL
2 WHEN job='CLERK' THEN
3 INTO c_emp
4 WHEN job='MANAGER' THEN
5 INTO cc_emp
6 SELECT *
7 FROM emp;
```

--67用plsql格式定义a=10，b=20计算并打印a+b

```
DECLARE
  a NUMBER :=10;
  b NUMBER :=20;
BEGIN
  dbms_output.put_line(a+b);
END;
```

```

1 DECLARE
2     a NUMBER :=10;
3     b NUMBER :=20;
4 BEGIN
5     dbms_output.put_line(a+b);
6 END;

```

--68用plsql格式使用隐式游标打印出emp表中empno=7369的员工姓名和工资
(数据类型同emp表)

```

1  --PLSQL写法
2  DECLARE
3      v_ename emp.ename%TYPE;
4      v_sal emp.sal%TYPE;
5  BEGIN
6      SELECT ename,sal
7      INTO v_ename,v_sal
8      FROM emp
9      WHERE empno=7369;
10     dbms_output.put_line('姓名: ' || v_ename || '工资: ' || v_sal);
11     END;
12
13  --存储过程
14  CREATE OR REPLACE PROCEDURE sp_emp(p_empno IN emp.empno%TYPE,p_ename OUT
15  emp.ename%TYPE,p_sal OUT emp.sal%TYPE)
16  IS
17  BEGIN
18      SELECT ename,sal
19      INTO p_ename,p_sal
20      FROM emp
21      WHERE empno=p_empno;
22     dbms_output.put_line('name:' || p_ename || 'sal:' || p_sal);
23     END;

```

--69用plsql格式分别使用case when 和if判断a=10和b=20两个数大小，并输出较大值

```

1  DECLARE
2      a NUMBER :=10;
3      b NUMBER :=20;
4  BEGIN
5      CASE
6          WHEN a>b THEN
7              dbms_output.put_line('你说得对');
8          WHEN a<b THEN
9              dbms_output.put_line('你是不是傻');
10         END CASE;
11     END;
12
13  DECLARE
14      a NUMBER :=10;
15      b NUMBER :=20;
16  BEGIN

```

```

17 IF a>b THEN
18     dbms_output.put_line('你说得对');
19 ELSE
20     dbms_output.put_line('你是不是傻');
21 END IF;
22 END;

```

--70用plsql格式，使用for loop格式计算1至100的连续加法的和

```

1 DECLARE
2     n NUMBER :=0;
3 BEGIN
4     FOR i IN 1..100 LOOP
5         n:=n+i;
6     END LOOP;
7     dbms_output.put_line(n);
8 END;

```

--71用显式游标查询10号部门所有员工的姓名并打印

```

1 DECLARE
2     CURSOR c_emp IS
3     SELECT ename
4     FROM emp
5     WHERE deptno=10;
6 BEGIN
7     FOR v_emp IN c_emp LOOP
8         dbms_output.put_line('姓名:' || v_emp.ename);
9     END LOOP;
10 END;

```

--72利用动态sql创建表e_emp

```

1 DECLARE
2     v_sql VARCHAR2(100):='create table e_emp as select * from emp';
3 BEGIN
4     EXECUTE IMMEDIATE v_sql;
5 END;

```

--73创建存储过程，能向dept表中添加一个新记录。（in参数）


```

1 CREATE OR REPLACE PROCEDURE p_dept(p_deptno IN c_dept.deptno%TYPE,p_dname IN
  c_dept.dname%TYPE,p_loc IN c_dept.loc%TYPE)
2 IS
3 BEGIN
4     INSERT INTO c_dept
5     (deptno,dname,loc)
6     VALUES
7     (p_deptno,p_dname,p_loc);
8     END;
9
10 --调用
11 begin
12     p_dept(12306,'DA','yongding');
13 end;

```

--74创建存储过程给定deptno,从emp表中查询给定职工的职工姓名和工资。

```

1 CREATE OR REPLACE PROCEDURE sp_emp(p_empno IN emp.empno%TYPE,
2     p_ename OUT emp.ename%TYPE,
3     p_sal OUT emp.sal%TYPE )
4 IS
5 BEGIN
6     SELECT ename,sal
7     INTO p_ename,p_sal
8     FROM emp
9     WHERE empno=p_empno;
10    dbms_output.put_line('姓名' || p_ename || '工资' || p_sal);
11    EXCEPTION
12        WHEN no_data_found THEN
13            dbms_output.put_line('未找到记录');
14    END;
15
16 --调用
17 DECLARE
18     v_empno emp.empno%TYPE:=&输入工号;
19     v_ename emp.ename%TYPE;
20     v_sal emp.sal%TYPE;
21 BEGIN
22     sp_emp(v_empno,v_ename,v_sal);
23 END;

```

--75创建存储过程，在执行调用过程时，输入emp表中某个雇员的姓名

```

1 CREATE OR REPLACE PROCEDURE sp_emp(p_ename IN emp.ename%TYPE,p_sal OUT
  emp.sal%TYPE)
2 IS
3 BEGIN
4     SELECT sal
5     INTO p_sal
6     FROM emp
7     WHERE ename=p_ename;
8     dbms_output.put_line('薪水' || p_sal);
9     EXCEPTION
10        WHEN no_data_found THEN
11            dbms_output.put_line('查无此人');
12    END;

```

```

13
14  --调用
15  DECLARE
16      v_ename emp.ename%TYPE:=&输入姓名;
17      v_sal emp.sal%TYPE;
18  BEGIN
19      sp_emp(v_ename,v_sal);
20  END;

```

--76创建存储过程，根据员工编号删除emp表中的相关记录

```

1  CREATE OR REPLACE PROCEDURE sp_emp(p_empno IN emp.empno%TYPE)
2  IS
3  BEGIN
4      DELETE FROM c_emp
5      WHERE empno=p_empno;
6      dbms_output.put_line('工号为:'||p_empno||'员工信息删除成功!');
7  EXCEPTION
8      WHEN no_data_found THEN
9          dbms_output.put_line('未找到改员工工号，请核实工号');
10 END;
11
12 --调用
13 DECLARE
14     v_empno emp.empno%TYPE:=&请输入工号;
15 BEGIN
16     sp_emp(v_empno);
17 END;

```