1、创建一个emp1表，其结构和数据与emp 表完全一致。用游标完成操作:显示 工资低于1500 的职员信息,并显示如果给他们涨30%工资后的工资。

create table emp1 fromemp; set serveroutput declarecursor cur fromemp where sal<1500; begin curloop dbms\_output.put\_line(v\_counter.empno||' '||v\_counter.ename||' '||v\_counter.mgr||'' ||v\_counter.hiredate||' '||v\_counter.comm||' '||v\_counter.sal||' '||v\_counter.sal\*1.3); end loop; end;

2、编写一个pl/sql 块，输出所有员工的员工名、员工号、工资和部门号。 declare cursor c\_emp fromemp; begin c\_emploop dbms\_output.put\_line(v\_emp.ename||'' ||v\_emp.empno||' '||v\_emp.deptno||' '||v\_emp.sal); end loop; end;

3、查询名为“smith”的员工信息，并输出其员工号、工资、部门号。如果该员工不存 在，则插入一条新记录，员工号为2007，员工名为“smith”，工资为1500，部门号 为10。如果存在多个名为“smith”的员工，则输出所有名为“smith”的员工号、工资 和部门号。

Declare v\_emp emp%rowtype;

begin select v\_empfrom emp where ename='smith'; dbms\_output.put\_line(v\_emp.empno||'' ||v\_emp.sal||' '||v\_emp.deptno); exception when no\_data\_found emp(empno,ename,sal,deptno)values(2007,'smith',1500,10); when too\_many\_rows fromemp where ename='smith') loop dbms\_output.put\_line(v.empno||' '||v.sal||' '||v.deptno); end loop; end;

4、创建一个存储过程，以员工号为参数，输出该员工的工资。

create replaceprocedure showsal(p\_empno emp.empno%type) v\_salemp.sal%type; begin select sal v\_salfrom emp where empno=p\_empno; dbms\_output.put\_line(v\_sal); end; begin showsal(7844); end;

5、创建一个函数，以部门号为参数，返回该部门的平均工资;

create replacefunction fun\_avgsal(p\_deptno emp.deptno%type) return emp.sal%type v\_salemp.sal%type; begin select avg(sal) v\_salfrom emp where deptno=p\_deptno; return v\_sal; end; begin dbms\_output.put\_line (fun\_avgsal(10)); end;

6、创建一个函数，以员工号为参数，返回该员工所在部门的平均工资。

create replacefunction fun\_sal(p\_empno emp.empno%type) return emp.sal%type v\_salemp.sal%type; begin select avg(sal) v\_salfrom emp where deptno= (select deptno from emp where empno=p\_empno); return v\_sal; end; begin dbms\_output.put\_line (fun\_sal (7844)); end;

7、emp表上创建一个触发器，当插入、删除或修改员工信息时，统计各个部门 的人数及平均工资，并输出。

create replacetrigger trg\_emp after insert empdeclare v\_sal emp.sal%type; v\_count number; begin select avg(sal),count(\*) v\_sal,v\_countfrom emp; dbms\_output.put\_line(v\_sal||' '||v\_count); end; update emp set sal=500 where empno=7844;

8、创建一个包，包含一个过程和一个游标。游标返回所有员工的信息，过程实现每 次输出游标中的5 条记录。

create replacepackage pkg\_persistcursor cursorc\_emp fromemp; procedure displayemp; end; create replacepackage body pkg\_persistcursor proceduredisplayemp v\_empemp%rowtype; begin openc\_emp; end 1..5loop fetch c\_emp v\_emp;dbms\_output.put\_line(v\_emp.empno||' v\_emp.ename);end loop; end; end;// begin pkg\_persistcursor.displayemp; end;

9、创建一个存储过程，以2 个整数为参数，输出工资在两者间的员工信息。

create replaceprocedure salbetween(min binary\_integer,max binary\_integer) fromemp where sal between min max)loop dbms\_output.put\_line(v\_emp.empno||' v\_emp.deptno);end loop; end; begin salbetween(1000,3000); end;

10、在emp 表上创建一个触发器，保证修改员工工资时，改后的工资低于同部门的 最高工资，同时高于同部门的最低工资。

create replacepackage pkg\_deptno v\_deptnoemp.deptno%type; v\_sal emp.sal%type; end;// create replacetrigger trg\_updateemp before update eachrow begin pkg\_deptno.v\_sal:=:new.sal; pkg\_deptno.v\_deptno:=:new.deptno; end;// create replacetrigger trg\_statement after update empdeclare v\_highsal emp.sal%type; v\_lowsal emp.sal%type; begin select max(sal),min(sal) v\_highsal,v\_lowsalfrom emp where deptno= pkg\_deptno.v\_deptno; raise\_application\_error(-20001,'thesal beyond!');end end;update emp set sal=500 where empno=7844;

11、在emp 表上创建一个触发器，当插入、删除或修改员工信息时，统计各个部门 的人数及平均工资并输出。

12、创建一个包，包含一个过程和一个函数。过程以部门号为参数输出该部门中工 资最高的员工名和员工号，函数以部门号为参数返回该部门员工的最高工资。

create replacepackage pkg\_emp functionfunc\_highsal(p\_deptno emp.deptno%type) return emp.sal%type; procedure proc\_highsal(p\_deptno emp.deptno%type); end;// create replacepackage body pkg\_emp functionfunc\_highsal(p\_deptno emp.deptno%type) return emp.sal%type v\_highsalemp.sal%type; begin select max(sal) v\_highsalfrom emp where deptno=p\_deptno; return v\_highsal; end; procedure proc\_highsal(p\_deptno emp.deptno%type) fromemp where deptno=p\_deptno sal=func\_highsal(p\_deptno))loop dbms\_output.put\_line(v\_emp.empno||' '||v\_emp.ename); end loop; end; end;// begin pkg\_emp.proc\_highsal(10); end;