/**\*9.2.1\*/**

**CREATE OR REPLACE PACKAGE update\_city AS**

**PROCEDURE u\_source( mflightno in numeric);**

**PROCEDURE u\_destin( mflightno in numeric);**

**END update\_city;**

**/\* 9.2.2\*/**

**CREATE OR REPLACE PACKAGE BODY update\_city AS**

**PROCEDURE u\_destin(mflightno in numeric)**

**AS**

**BEGIN**

**UPDATE  flight**

**SET destin = 'NEW DELHI'**

**WHERE  flight\_no = mflightno;**

**END u\_destin;**

**PROCEDURE u\_source(mflightno in numeric)**

**AS**

**BEGIN**

**UPDATE flight**

**SET source1 = 'NEW DELHI'**

**WHERE**flight\_no**=** mflightno**;**

**END u\_source;**

**END update\_city;**

/**\*9.2.3\*/**

**DECLARE**

**CURSOR c1 is SELECT FLIGHT\_NO, SOURCE1, DESTIN FROM  FLIGHT;**

**flight\_rec c1%rowtype;**

**BEGIN**

**OPEN c1;**

**LOOP**

**FETCH c1 INTO flight\_rec;**

**EXIT WHEN c1%NOTFOUND;**

**IF( (flight\_rec.source1 = 'DELHI') OR**

**(flight\_rec.source1 = 'Delhi'))**

**THEN**

**/\* calling the procedure of package update\_city \*/**

**update\_city.u\_source(flight\_rec.flight\_no);**

**ELSIF((flight\_rec.destin = 'DELHI') OR**

**(flight\_rec.destin = 'Delhi'))**

**THEN**

**/\* calling another procedure of package update\_city\*/**

**update\_city.u\_destin(flight\_rec.flight\_no);**

**END IF;**

**END LOOP;**

**CLOSE  c1;**

**END;**

**/\*9.4.1\*/**

**CREATE OR REPLACE PACKAGE CUST\_PACK**

**As**

**PROCEDURE BONUS( customer\_Code IN NUMERIC);**

**PROCEDURE add\_cust( code IN NUMBER,**

**name IN VARCHAR2,**

**age IN NUMBER,**

**address IN VARCHAR2,**

**phone IN VARCHAR2);**

**PROCEDURE remove\_cust( code IN NUMBER);**

**PROCEDURE remove\_inf( code IN NUMBER);**

**CURSOR c1 IS SELECT \* FROM cust1;**

**CURSOR c2 IS SELECT \* FROM inform;**

**CURSOR c3 IS SELECT \* FROM insure;**

**END CUST\_PACK;**

**/\*end of package specification \*/**

**/\*9.4.2\*/**

**CREATE OR REPLACE PACKAGE BODY  cust\_pack**

**IS**

**/\* global variable for this package \*/**

**/\* this can be accessed only inside the package \*/**

**pcust\_code number(5);**

**/\*\*\*\*\*\*\*\*\*\*\*\*\*bonus procedure\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/**

**/\* In this a procedure named as bonus() is created which takes**

**numeric parameter as input. \*/**

**PROCEDURE bonus(customer\_code IN NUMERIC )  IS**

**mbonus      FLOAT;**

**msalary     FLOAT;**

**over\_salary EXCEPTION;**

**mcode      NUMBER(6);**

**mdetail     varchar2(30);**

**BEGIN**

**SELECT salary INTO msalary FROM inform**

**WHERE cust\_code = customer\_code;**

**IF(msalary > 100000)**

**THEN**

**raise over\_salary;**

**ELSE**

**mbonus := msalary \* .10;**

**msalary := msalary + mbonus;**

**Dbms\_output.put\_line('Bonus Earned'||'    '||mbonus);**

**UPDATE INFORM SET salary = msalary**

**WHERE cust\_code = customer\_code;**

**END IF;**

**EXCEPTION**

**WHEN over\_salary**

**THEN**

**UPDATE INFORM SET salary = salary + 10000**

**WHERE cust\_code = customer\_code;**

**WHEN OTHERS**

**THEN**

**UPDATE INFORM SET salary = 0.0**

**WHERE cust\_code = customer\_Code;**

**END bonus;**

**/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*add\_cust procedure \*\*\*\*\*\*\*\*\*\*\*\*\*\***

**this procedure insert record in the cust1 table                      \*/**

**PROCEDURE  add\_cust( code IN NUMBER,**

**name IN VARCHAR2,**

**age IN NUMBER,**

**address IN VARCHAR2,**

**phone IN VARCHAR2)**

**IS**

**BEGIN**

**SELECT cust\_code INTO pcust\_code FROM cust1**

**WHERE cust\_code = code;**

**EXCEPTION**

**WHEN NO\_DATA\_FOUND**

**THEN**

**INSERT INTO cust1**

**VALUES( code, name, age, address, phone);**

**WHEN OTHERS THEN**

**raise\_application\_error( -20000, 'oracle error');**

**END add\_cust;**

**/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*remove\_cust procedure\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**this procedure removes a particular record from the cust1 table \*/**

**PROCEDURE remove\_cust( code IN NUMBER)**

**IS**

**BEGIN**

**DELETE FROM cust1**

**WHERE cust\_code = code;**

**IF ( SQL%NOTFOUND)**

**THEN**

**raise\_application\_error( -20005, 'no data found');**

**END IF;**

**END remove\_cust;**

**/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*remove\_inf procedure\*\*\*\*\*\*\*\*\*\*\*\***

**this procedure removes a record from the inform table \*/**

**PROCEDURE remove\_inf( code IN NUMBER)**

**IS**

**BEGIN**

**DELETE FROM inform**

**WHERE cust\_code = code;**

**IF ( SQL%NOTFOUND)**

**THEN**

**raise\_application\_error( -20010, 'no data');**

**END IF;**

**END remove\_inf;**

**END cust\_pack;**

**/\*9.4.3(a)\*/**

**BEGIN**

**cust\_pack.remove\_inf(80);**

**END;**

**/\*9.4.3(b)\*/**

**BEGIN**

**cust\_pack.remove\_cust(90);**

**END;**

**/\* 9.4.3( c )\*/**

**BEGIN**

**cust\_pack.bonus(60);**

**END**

**/\* 9.4.3 (d)\*/**

**BEGIN**

**cust\_pack.add\_cust(80,’JOHN’,25,’8 ITI’,654113);**

**END;**

**\*\*\*\*\*\*\*\*\***

**/\*9.4.4\*/**

**CREATE PACKAGE VEHI\_MGMT**

**As**

**FUNCTION issuep( licenseno IN number)**

**RETURN varchar2;**

**PROCEDURE replacename( licenseno IN number, newname IN varchar2 );**

**END VEHI\_MGMT;**

**/\* End of creation of Package VEHI\_MGMT**

**/\*9.4.5\*/**

**CREATE PACKAGE BODY VEHI\_MGMT**

**AS**

**FUNCTION issuep( licenseno IN NUMBER)**

**RETURN VARCHAR2 AS**

**place VARCHAR2(15);**

**BEGIN**

**SELECT place\_issued INTO place**

**FROM license**

**WHERE lic\_no = licenseno;**

**RETURN place;**

**END issuep;**

**PROCEDURE replacename( licenseno IN NUMBER, newname IN VARCHAR2 )**

**AS**

**/\*this procedure replaces the name of place\_issued to new**

**name for the given license number \*/**

**BEGIN**

**UPDATE LICENSE**

**SET place\_issued = newname**

**WHERE lic\_no = licenseno;**

**END replacename;**

**--end of procedure definition**

**END VEHI\_MGMT;**

**/\* end of vehi\_mgmt package \*/**

**/\* 9.4.6(a)\*/**

**BEGIN**

**VEHI\_MGMT.replacename(15123,'Delhi');**

**END;**

**/\*9.4.6(b)\*/**

**DECLARE**

**mplace varchar(12);**

**BEGIN**

**/\* calling the function issuep()\*/**

**mplace :=VEHI\_MGMT.issuep(15123);**

**Dbms\_output.put\_line(mplace);**

**END;**

**/\*9.4.7\*/**

**Create package course\_mgmt**

**As**

**function ret\_credit( teacher number)**

**return number;**

**procedure faculty\_resign( teacher number);**

**CURSOR c1 is select \* from faculty;**

**End course\_mgmt;**

**/\*9.4.8\*/**

**Create package body course\_mgmt**

**As**

**FUNCTION ret\_credit( teacher number)**

**RETURN number as**

**tot\_credit   number(3) := 0;**

**null\_credit   exception;**

**BEGIN**

**SELECT sum(credits) INTO tot\_credit**

**from course where teacherno = teacher;**

**IF tot\_credit is NULL**

**THEN**

**raise null\_credit;**

**end if;**

**return tot\_credit;**

**EXCEPTION**

**WHEN null\_credit then**

**return 0;**

**end ret\_credit;**

**PROCEDURE faculty\_resign( teacher number )**

**as**

**BEGIN**

**/\* delete his record from the table faculty \*/**

**delete from faculty where teacherno = teacher;**

**/\* if that teacherno doesn't exist, then raise error \*/**

**IF sql%notfound THEN**

**raise\_application\_error( -20005, 'Teacher no: '**

**||to\_char(teacher) || ' not existing');**

**END IF;**

**End faculty\_resign;**

**End course\_mgmt;**

**/\*9.4.8\*/**

**Declare**

**t\_credits  number(3);**

**Begin**

**/\* cursor used from the package \*/**

**for faculty\_rec in course\_mgmt.c1**

**loop**

**/\* package member function is called here \*/**

**t\_credits := course\_mgmt.ret\_credit( faculty\_rec.teacherno);**

**insert into ct values( faculty\_rec.teacherno,**

**faculty\_rec.teachername, t\_credits);**

**end loop;**

**end;**

**/\*9.4.9\*/**

**BEGIN**

**COURSE\_MGMT.faculty\_resign(5);**

**END;**

**;**

**/\*9.4.10\*/**

**Declare**

**t\_credits  number(3);**

**Begin**

**/\* cursor used from the package \*/**

**for faculty\_rec in course\_mgmt.c1**

**loop**

**t\_credits := course\_mgmt.ret\_credit( faculty\_rec.teacherno);**

**insert into ct values( faculty\_rec.teacherno,**

**faculty\_rec.teachername, t\_credits);**

**end loop;**

**END;**

**/\*9.4.11 \*/**

**Create package phone**

**as**

**/\* function to compute bill \*/**

**function compute\_bill( telnumber number, code number)**

**return number;**

**/\* procedure to delete customer records \*/**

              procedure delete\_cust( telnumber number );

**CURSOR c1 is select \* from cust\_details;**

**End phone;**

**/\*9.4.12\*/**

**create or replace package body phone**

**as**

**/\* Following function computes the bill amount. The parameters supplied to this are the telephone number and his area code.\*/**

**FUNCTION COMPUTE\_BILL(telnumber number, code number)**

**RETURN NUMBER**

**AS**

**CURSOR C1 IS SELECT duration, area\_code**

**FROM CALLS\_INFO**

**WHERE TEL\_NO = telnumber ;**

**Bill\_Amount  number(7,2) := 0;**

**RP      number(3,2);**

**BEGIN**

**FOR TEL\_REC IN C1**

**LOOP**

**SELECT rateperminute INTO RP**

**FROM RATE\_INFO**

**WHERE ACODE = code AND**

**AREA\_CODE = TEL\_REC.AREA\_CODE;**

**Bill\_Amount:= Bill\_Amount + RP \* TEL\_REC.DURATION;**

**END LOOP;**

**RETURN Bill\_Amount;**

**END;**

/\*-procedure delete\_cust() has telephone number as the parameter and

deletes the records from cust\_details table cust\_info table for the given telephone number and stores  the number of records deleted . This also has an Exception section\*/

**PROCEDURE DELETE\_CUST( telnumber Number)**

**AS**

**cdeleted   Number;**

**ideleted    Number;**

**BEGIN**

**DELETE FROM CUST\_DETAILS**

**WHERE TEL\_NO = telnumber;**

**cdeleted  := SQL%rowcount;**

**INSERT INTO RD VALUES( cdeleted );**

**DELETE FROM CALLS\_INFO**

**WHERE TEL\_NO = telnumber;**

**ideleted := SQL%ROWCOUNT;**

**INSERT INTO RD VALUES( ideleted);**

**EXCEPTION**

**WHEN NO\_DATA\_FOUND THEN**

**INSERT INTO RD VALUES(0);**

**END DELETE\_CUST;**

**end phone;**

**/\*9.4.13\*/**

**DECLARE**

**BILL\_AMOUNT number;**

**BEGIN**

**FOR CUST\_REC IN phone.C1**

**LOOP**

**/\* function compute\_bill() of package phone is called here \*/**

**BILL\_AMOUNT := phone.COMPUTE\_BILL(CUST\_REC.Tel\_no, CUST\_REC.Acode);**

**/\* the bill amount along with the details of customer are stored here \*/**

**INSERT INTO BILL**

**VALUES( CUST\_REC.Tel\_no, CUST\_REC.Cust\_Name, Bill\_Amount);**

**END LOOP;**

**END;**