**/\*7.3.1\*/**

**CREATE OR REPLACE FUNCTION source(mflightno IN NUMERIC)**

**RETURN VARCHAR2**

**AS**

**msource  varchar2(12);**

**BEGIN**

**SELECT source1 INTO msource**

**FROM  FLIGHT**

**WHERE flight\_no = mflightno;**

**RETURN msource;**

**END  source;**

**/\*7.4.1\*/**

**DECLARE**

**msource varchar(12);**

**BEGIN**

**/\*-- calling the function source()\*/**

**msource := source(:flight\_number);**

**insert into res14 values(msource);**

**END;**

**/\*7.5.1\*/**

**CREATE FUNCTION FLT\_info( passno IN NUMBER, mfare IN NUMBER) RETURN NUMBER as**

**FLNO number;**

**BEGIN**

**SELECT FLIGHT\_NO INTO FLNO**

**FROM  PASSENG**

**WHERE PASS\_CODE=passno and FARE > mfare;**

**RETURN FLNO;**

**END;**

**/\*7.5.2\*/**

**DECLARE**

**code      number;**

**fare1     number;**

**FLTNO number;**

**BEGIN**

**code :=1115;**

**fare1 := 12000;**

**FLTNO:= FLT\_INFO(code,fare1);**

**-- positional notation**

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

**END;**

**/\*7.5.3\*/**

**DECLARE**

**code      INTEGER;**

**fare1     REAL;**

**FLTNO number;**

**BEGIN**

**fare1:=12000;**

**code :=1115;**

**FLTNO:=FLT\_INFO(mfare=>fare1,passno=>code);**

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

**END;**

/**\* 7.5.4\*/**

**DECLARE**

**code      INTEGER;**

**fare1     REAL;**

**FLTNO number;**

**BEGIN**

**fare1:=12000;**

**code :=1115;**

**FLTNO:=FLT\_info(code,mfare=>fare1);**

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

**END;**

**\*7.6.1\*/**

**Create FUNCTION info\_seat\_disc( column\_number in number)**

**Return number as**

**info       number(4,0);**

**Begin**

**IF (column\_number = 5)**

**Then**

**SELECT min(no\_of\_seats) into info**

**FROM TRAIN;**

**ElSIF (column\_number = 3)**

**THEN**

**SELECT max(per\_discount) into info**

**FROM JOURNEY;**

**END IF;**

**RETURN info;**

**END;**

**/\*7.6.2\*/**

**Declare**

**mxdiscount  number(6);**

**minseats            number(6);**

**BEGIN**

**/\* position number 3 is for discount column \*/**

**mxdiscount := info\_seat\_disc (3);**

**/\* position number 5 is for number of seats \*/**

**minseats     := info\_seat\_disc (5);**

**/\* the computes result is now inserted \*/**

**insert into total values(mxdiscount, minseats);**

**End;**

**/\*7.6.3\*/**

**CREATE FUNCTION per\_sal( ccode IN NUMBER, cname out varchar)**

**RETURN number**

**As**

**sal number(10,2);**

**BEGIN**

**SELECT  salary,cust\_name**

**INTO sal,cname**

**FROM CUST1,INFORM**

**where CUST1.cust\_code = ccode and CUST1.cust\_code=INFORM.cust\_code;**

**DBMS\_OUTPUT.PUT\_LINE(cname);**

**RETURN sal;**

**END per\_sal;**

**/\*7.6.4\*/**

**Declare**

**code      number(4);**

**sal1       number(8);**

**cnam  varchar(30);**

**Begin**

**/\* Function is called here \*/**

**sal1:= per\_sal(:code,cnam);**

**DBMS\_OUTPUT.PUT\_LINE(Sal1);**

**end;**

**/\* 7.6.5 \*/**

**Create function ret\_credit( mteacherno number)**

**Return number as**

**tot\_credit   number(3) := 0;**

**null\_credit   exception;**

**Begin**

**select sum(credits) into tot\_credit**

**from course**

**WHERE teacherno = mteacherno;**

**If tot\_credit is NULL**

**Then**

**RAISE null\_credit;**

**End IF;**

**return tot\_credit;**

**Exception**

**When null\_credit**

**Then**

**return 0;**

**End ret\_credit;**

**/\*7.6.6\*/**

**Declare**

**t\_credits  number(3);**

**cursor c1 is select \* from faculty**

**order by teacherno asc;**

**Begin**

**For faculty\_rec in c1**

**Loop**

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

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

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

**End loop;**

**End;**

**/\*7.6.7\*/**

**CREATE FUNCTION COMPUTE\_BILL( telnumber NUMBER, mcode 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 = mcode AND**

**area\_code = TEL\_REC. area\_code;**

**BILL\_AMOUNT := BILL\_AMOUNT + RP \* TEL\_REC. duration;**

**END LOOP;**

**RETURN BILL\_AMOUNT;**

**END;**

**/\* 7.6.8\*/**

**DECLARE**

**/\* list of customers, whose bill is to be prepared \*/**

**CURSOR C1 IS SELECT \* FROM CUST\_DETAILS;**

**/\* compute bill amount is stored in this variable\*/**

**BILL\_AMOUNT  NUMBER(7,2);**

**BEGIN**

**FOR CUST\_REC IN C1**

**LOOP**

**/\* function compute\_bill() is called here \*/**

**BILL\_AMOUNT := COMPUTE\_BILL(CUST\_REC.TEL\_NO, CUST\_REC.ACODE);**

**INSERT INTO BILL VALUES( CUST\_REC.TEL\_NO, CUST\_REC.CUST\_NAME, BILL\_AMOUNT);**

**END LOOP;**

**END;**

**/\* 7.6.9\*/**

**Create FUNCTION uprice2 (invno number )**

**Return number**

**As**

**twice number(10,2);**

**Begin**

**select unit\_price \* 2 into twice**

**FROM inventory**

**WHERE inv\_no = invno;**

**RETURN twice;**

**End;**

**/\*7.6.10\*/**

**DECLARE**

**minv     number(4);**

**mamount           number(8);**

**BEGIN**

**SELECT inv\_no, unit\_price into minv, mamount**

**FROM inventory**

**WHERE inv\_no = :inventory\_number;**

**DBMS\_OUTPUT.PUT\_LINE('Original Unit Price');**

**DBMS\_OUTPUT.PUT\_LINE(mamount);**

**/\* Function is called here \*/**

**mamount := uprice2( minv);**

**DBMS\_OUTPUT.PUT\_LINE('twice Unit Price');**

**DBMS\_OUTPUT.PUT\_LINE(mamount);**

**END;**

**/\*7.6.11\*/**

**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;**

**/\*7.6.12\*/**

**DECLARE**

**mplace varchar(12);**

**BEGIN**

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

**mplace := issuep(:lno);**

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

**END;**

**/\*7.6.13\*/**

**Declare**

**CURSOR c1 is**

**SELECT OWNER.owner\_code,LICENSE.lic\_no**

**from  OWNER, LICENSE where OWNER.Owner\_code=LICENSE.owner\_code;**

**Begin**

**For License\_rec in c1**

**Loop**

**if (issuep(License\_rec.lic\_no) = 'Delhi')**

**Then**

**insert into v3 values (License\_rec.owner\_code,License\_rec.lic\_no);**

**END IF;**

**end loop;**

**END;**

**/\*7.6.14\*/**

**CREATE FUNCTION  INVALID\_license(expiry\_date in date)**

**RETURN boolean as**

**cdate  date;**

**BEGIN**

**SELECT sysdate into cdate FROM sys.dual;**

**IF(expiry\_date  < cdate)**

**THEN**

**RETURN TRUE;**

**ELSE**

**RETURN FALSE;**

**END IF;**

**END INVALID\_license;**

**/\* 7.6.15\*/**

**DECLARE**

**CURSOR C1 IS SELECT LIC\_NO,DATE\_EXPIRY FROM LICENSE;**

**RES BOOLEAN;**

**BEGIN**

**For REC in  c1**

**LOOP**

**RES:= INVALID\_LICENSE(REC.DATE\_EXPIRY);**

**IF RES=FALSE**

**THEN**

**dbms\_output.put\_line('License NO'||REC.LIC\_NO||'   '||'License Invalid '||'   '||REC.DATE\_EXPIRY);**

**END IF;**

**END LOOP;**

**END;**