create or replace PROCEDURE PATIENT\_ADMIT(pname in PATIENT.P\_NAME%type,paddress in PATIENT.P\_ADDRESS%type,pgender in PATIENT.P\_GENDER%type,pdob in PATIENT.P\_DOB%type,pcontact in PATIENT.P\_CONTACT\_NO%type,pstatus in PATIENT.P\_STATUS%type,loginname in VARCHAR2,roomno in ROOM.ROOM\_NO%type := null) AS

maxid PATIENT.P\_ID%type;

status PATIENT.P\_STATUS%type;

rec\_id NUMBER(5);

BEGIN

INSERT INTO PATIENT values(PATIENT\_SEQ.NEXTVAL,pname,paddress,pgender,pdob,pcontact,pstatus,'Not Released');

SELECT P\_STATUS INTO status FROM PATIENT WHERE P\_ID=(SELECT MAX(P\_ID) FROM PATIENT);

SELECT R\_ID INTO rec\_id FROM RECEPTIONIST WHERE R\_EMAIL=loginname;

SELECT P\_ID INTO maxid FROM PATIENT WHERE P\_ID=(SELECT MAX(P\_ID) FROM PATIENT);

INSERT INTO ASSIST\_PATIENT VALUES(ASSIST\_PATIENT\_SEQ.NEXTVAL,maxid,rec\_id);

IF(status='In Door') then

INSERT INTO ASSIGNED\_ROOM VALUES(ASSIGNED\_ROOM\_SEQ.NEXTVAL,maxid,roomno);

UPDATE ROOM SET ROOM\_STATUS='booked' WHERE ROOM\_NO=roomno;

END IF;

END PATIENT\_ADMIT;

create or replace PROCEDURE PATIENT\_CHECKOUT ( PATIENTID IN NUMBER ) AS

roomid ROOM.ROOM\_NO%TYPE;

BEGIN

UPDATE PATIENT SET P\_CONDITION='Released' WHERE P\_ID=PATIENTID;

SELECT R.ROOM\_NO INTO roomid FROM ROOM R,ASSIGNED\_ROOM A,PATIENT P WHERE R.ROOM\_NO=A.ROOM\_NO AND A.P\_ID=P.P\_ID AND A.P\_ID=PATIENTID;

UPDATE ROOM SET ROOM\_STATUS='empty' WHERE ROOM\_NO=roomid;

UPDATE LOG\_PATIENT SET RELEASE\_DATE=SYSDATE WHERE PATIENT\_ID=PATIENTID;

END PATIENT\_CHECKOUT;

create or replace PROCEDURE ADMIN\_CHANGE\_PASSWORD(UPASSWORD in USER\_TYPE.U\_PASSWORD%TYPE, UEMAIL in USER\_TYPE.U\_EMAIL%TYPE) AS

BEGIN

UPDATE USER\_TYPE SET U\_PASSWORD=UPASSWORD WHERE U\_EMAIL=UEMAIL;

END ADMIN\_CHANGE\_PASSWORD;

CREATE OR REPLACE PROCEDURE ADD\_DOCTOR(dname in DOCTOR.D\_NAME%type, daddress in DOCTOR.D\_ADDRESS%type, dcontact in DOCTOR.D\_CONTACT\_NO%type, djoindate in DOCTOR.D\_JOINING\_DATE%type, dsalary in DOCTOR.D\_SALARY%type, ddesignation in DOCTOR.D\_DESIGNATION%type) AS

BEGIN

INSERT INTO DOCTOR values(DOCTOR\_SEQ.NEXTVAL,dname,daddress,dcontact,djoindate,dsalary,ddesignation);

END ADD\_DOCTOR;

CREATE OR REPLACE PROCEDURE ADD\_NURSE(nname in NURSE.N\_NAME%type, naddress in NURSE.N\_ADDRESS%type, ncontact in NURSE.N\_CONTACT\_NO%type, njoindate in NURSE.N\_JOINING\_DATE%type, nsalary in NURSE.N\_SALARY%type, ndesignation in NURSE.N\_DESIGNATION%type) AS

BEGIN

INSERT INTO NURSE values(NURSE\_SEQ.NEXTVAL,nname,naddress,ncontact,njoindate,nsalary,ndesignation);

END ADD\_NURSE;

create or replace PROCEDURE ADD\_RECEPTIONIST(rname in RECEPTIONIST.R\_NAME%type, remail in RECEPTIONIST.R\_EMAIL%type, raddress in RECEPTIONIST.R\_ADDRESS%type, rcontact in RECEPTIONIST.R\_CONTACT\_NO%type, rjoindate in RECEPTIONIST.R\_JOINING\_DATE%type, rsalary in RECEPTIONIST.R\_SALARY%type, rdesignation in RECEPTIONIST.R\_DESIGNATION%type) AS

maxid USER\_TYPE.ID%TYPE;

email USER\_TYPE.U\_EMAIL%TYPE;

BEGIN

INSERT INTO RECEPTIONIST values(RECEPTIONIST\_SEQ.NEXTVAL,rname,remail,raddress,rcontact,rjoindate,rsalary,rdesignation);

SELECT R\_ID,R\_EMAIL INTO maxid,email FROM RECEPTIONIST WHERE R\_ID=(SELECT MAX(R\_ID) FROM RECEPTIONIST);

INSERT INTO USER\_TYPE values(USER\_TYPE\_SEQ.NEXTVAL,'RECEPTIONIST',email,123456,maxid);

END ADD\_RECEPTIONIST;

CREATE OR REPLACE PROCEDURE ADD\_ROOM(roomtype IN ROOM.ROOM\_TYPE%TYPE, roomcost IN ROOM.ROOM\_COST%TYPE) AS

BEGIN

INSERT INTO ROOM values(ROOM\_SEQ.NEXTVAL, roomtype, roomcost, 'empty');

END ADD\_ROOM;

create or replace PROCEDURE ADMIN\_PROFILE\_UPDATE(UEMAIL IN ADMIN.ADMIN\_EMAIL%TYPE, ANAME IN ADMIN.ADMIN\_NAME%TYPE, AADDRESS IN ADMIN.ADMIN\_ADDRESS%TYPE, ACONTACT IN ADMIN.ADMIN\_CONTACT\_NO%TYPE, ADOB IN ADMIN.DATE\_OF\_BIRTH%TYPE) AS

BEGIN

UPDATE ADMIN SET ADMIN\_NAME=ANAME, ADMIN\_ADDRESS=AADDRESS, ADMIN\_CONTACT\_NO=ACONTACT, DATE\_OF\_BIRTH=ADOB WHERE ADMIN\_EMAIL=UEMAIL;

END ADMIN\_PROFILE\_UPDATE;

create or replace PROCEDURE LOGIN\_INFO\_PROCEDURE(email in USER\_TYPE.U\_EMAIL%TYPE) AS

id USER\_TYPE.ID%TYPE;

old USER\_TYPE.U\_EMAIL%TYPE;

CURSOR REC IS

SELECT LOGGED\_PERSON FROM LOGIN\_INFO;

FLAG NUMBER(2):=0;

BEGIN

SELECT ID INTO id FROM USER\_TYPE WHERE U\_EMAIL=email;

FOR I IN REC LOOP

IF I.LOGGED\_PERSON=id THEN

FLAG:=1;

END IF;

END LOOP;

IF FLAG=1 THEN

UPDATE LOGIN\_INFO SET LOGIN\_DATE=SYSDATE WHERE LOGGED\_PERSON=id;

ELSIF FLAG=0 THEN

INSERT INTO LOGIN\_INFO VALUES(LOGIN\_INFO\_SEQ.NEXTVAL,id,sysdate);

END IF;

END LOGIN\_INFO\_PROCEDURE;