

|  |
| --- |
| Hospital Management  System |
| Advanced Database Management System  Sec: A   |  |  | | --- | --- | | Name | ID | | 1. Shihab, Mehedi Hasan | 14-27923-3 | | 1. Sajal, Md. Min Hajul Alam | 14-25856-1 | | 1. Rifat, Mohiuddin Ahmed | 15-30438-3 | |
| Submitted To  **REZWAN AHMED** |

**Tables:**

CREATE TABLE ADMIN

(ADMIN\_ID NUMBER (7) PRIMARY KEY,

ADMIN\_NAME VARCHAR2(50) NOT NULL,

ADMIN\_ADDRESS VARCHAR2(100) NULL,

ADMIN\_CONTACT\_NO NUMBER(11) NULL,

ADMIN\_EMAIL VARCHAR2(200) NULL,

DATE\_OF\_BIRTH DATE NULL);

INSERT INTO ADMIN VALUES (1,'Shihab', 'Gazipur',01521213853,'sshihabb007@gmail.com',to\_date('03-NOV-96','DD-MM-YY'));

CREATE TABLE PATIENT

(P\_ID NUMBER(7) PRIMARY KEY,

P\_NAME VARCHAR2(50) NOT NULL,

P\_ADDRESS VARCHAR2(100) NOT NULL,

P\_GENDER VARCHAR2(10) NOT NULL,

P\_DOB DATE NOT NULL,

P\_CONTACT\_NO NUMBER(11) NOT NULL,

P\_STATUS VARCHAR2(30) NOT NULL,

P\_CONDITION VARCHAR2(30))

INSERT INTO PATIENT VALUES (101,'Tanjir', 'Basundhara','Male' ,to\_date('11-APR-95','DD-MM-YY'),01811112233,'In Door', 'Not Released');

INSERT INTO PATIENT VALUES (102,'Arafat', 'Kochukhet','Male' ,to\_date('10-JAN-90','DD-MM-YY'),01711804015,'Out Door', 'Not Released');

INSERT INTO PATIENT VALUES (103,'Wrivu', 'Banani','Male' ,to\_date('02-DEC-98','DD-MM-YY'),01744338165,'In Door', 'Not Released');

INSERT INTO PATIENT VALUES (104,'Dipto', 'Khilkhet','Male' ,to\_date('11-FEB-95','DD-MM-YY'),01933451987,'In Door', 'Not Released');

INSERT INTO PATIENT VALUES (105,'Bijoy', 'Basundhara','Male' ,to\_date('07-Feb-97','DD-MM-YY'),01557998877,'Out Door', 'Released');

INSERT INTO PATIENT VALUES (106,'Kawsur', 'Niketon','Male' ,to\_date('11-APR-95','DD-MM-YY'),01811112233,'In Door', 'Released');

CREATE TABLE ROOM

(ROOM\_NO NUMBER(7) PRIMARY KEY,

ROOM\_TYPE VARCHAR2(20) NOT NULL,

ROOM\_COST NUMBER(5) NOT NULL,

ROOM\_STATUS VARCHAR2(20) NOT NULL);

INSERT INTO ROOM VALUES(01,'NON AC',1000, 'empty');

INSERT INTO ROOM VALUES(02,'NON AC',1000, 'booked');

INSERT INTO ROOM VALUES(03,'NON AC',1000, 'empty');

INSERT INTO ROOM VALUES(04,'AC',1500, 'empty');

INSERT INTO ROOM VALUES(05,'AC',1500, 'booked');

INSERT INTO ROOM VALUES(06,'AC',1500, 'booked');

INSERT INTO ROOM VALUES(07,'DULEX',2500, 'booked');

INSERT INTO ROOM VALUES(08,'DULEX',2500, 'booked');

INSERT INTO ROOM VALUES(09,'DULEX',2500, 'empty');

CREATE TABLE DOCTOR

(D\_ID NUMBER(7) PRIMARY KEY,

D\_NAME VARCHAR2(50) NOT NULL,

D\_ADDRESS VARCHAR2(50) NOT NULL,

D\_CONTACT\_NO NUMBER(11) NOT NULL,

D\_JOINING\_DATE DATE NOT NULL,

D\_SALARY NUMBER(10) NOT NULL,

D\_DESIGNATION VARCHAR2(50) NOT NULL);

INSERT INTO DOCTOR VALUES (401,'Emer Khan','Dhaka',1811112153,to\_date('22-SEP-79','DD-MM-YY'),50000,'M.O');

INSERT INTO DOCTOR VALUES (402,'Helali','Board bazar',01682408748,to\_date('18-JAN-90','DD-MM-YY'),40000,'MBBS');

INSERT INTO DOCTOR VALUES (403,'Rezwan','BASUNDHARA',01833458769,to\_date('08-FEB-02','DD-MM-YY'),70000,'ASSISTANT PROFESSOR');

INSERT INTO DOCTOR VALUES (404,'Shakib khan','HATIRJHEEL',01833451951,to\_date('07-SEP-07','DD-MM-YY'),60000,'M.O');

INSERT INTO DOCTOR VALUES (405,'Sumon','KOCHUKHET',01987962967,to\_date('16-JUL-95','DD-MM-YY'),25000,'ASSISTANT SURGEN');

CREATE TABLE NURSE

(N\_ID NUMBER (7) PRIMARY KEY,

N\_NAME VARCHAR2(50) NOT NULL,

N\_ADDRESS VARCHAR2(50) NOT NULL,

N\_CONTACT\_NO NUMBER(11) NOT NULL,

N\_JOINING\_DATE DATE NOT NULL,

N\_SALARY NUMBER(10) NOT NULL,

N\_DESIGNATION VARCHAR2(50) NOT NULL);

INSERT INTO NURSE VALUES (501,'Rahima','UTTARA',1811112153,to\_date('22-SEP-80','DD-MM-YY'),6000,'JUNIOR NURSE');

INSERT INTO NURSE VALUES (502,'Tamanna','MIRPUR',1822985153,to\_date('23-JUN-80','DD-MM-YY'),9000,'SENIOR NURSE');

INSERT INTO NURSE VALUES (503,'PROGGA','BANANI',1787592153,to\_date('26-JUL-90','DD-MM-YY'),6000,'JUNIOR NURSE');

INSERT INTO NURSE VALUES (504,'Fatema','Tongi',1515115842,to\_date('12-SEP-95','DD-MM-YY'),6000,'JUNIOR NURSE');

INSERT INTO NURSE VALUES (505,'Morzina','BADDA',1887930053,to\_date('26-FEB-97','DD-MM-YY'),8000,'SENIOR NURSE');

CREATE TABLE RECEPTIONIST

(R\_ID NUMBER (7) PRIMARY KEY,

R\_NAME VARCHAR2(50) NOT NULL,

R\_EMAIL VARCHAR2(50) NOT NULL,

R\_ADDRESS VARCHAR2(50) NOT NULL,

R\_CONTACT\_NO NUMBER(11) NOT NULL,

R\_JOINING\_DATE DATE NOT NULL,

R\_SALARY NUMBER(10) NOT NULL,

R\_DESIGNATION VARCHAR2(50) NOT NULL);

INSERT INTO RECEPTIONIST VALUES (601,'Sakib','sakibb@gmail.com','UTTARA',1811872153,to\_date('21-SEP-80','DD-MM-YY'),5000,'JUNIOR RECEPTIONIST');

INSERT INTO RECEPTIONIST VALUES (602,'Mahmuda','mahmuda007@gmail.com','BADDA',1822089753,to\_date('21-JUN-80','DD-MM-YY'),8000,'SENIOR RECEPTIONIST');

INSERT INTO RECEPTIONIST VALUES (603,'TARIQUR','tariqur@gmail.com','BANANI',1701112153,to\_date('21-JUL-90','DD-MM-YY'),5000,'JUNIOR RECEPTIONIST');

INSERT INTO RECEPTIONIST VALUES (604,'TANVIR','tanvir@gmail.com','GULSHAN',1515112153,to\_date('01-SEP-95','DD-MM-YY'),5000,'JUNIOR RECEPTIONIST');

INSERT INTO RECEPTIONIST VALUES (605,'ANIKA','anika@gmail.com','BADDA',1811100053,to\_date('21-FEB-98','DD-MM-YY'),8000,'SENIOR RECEPTIONIST');

CREATE TABLE USER\_TYPE

(U\_ID NUMBER (7) PRIMARY KEY,

ID NUMBER (7) NOT NULL,

U\_TYPE VARCHAR2(50) NOT NULL,

U\_EMAIL VARCHAR2(50) NOT NULL,

U\_PASSWORD VARCHAR2(50) NOT NULL);

INSERT INTO USER\_TYPE VALUES (701, 601,'RECEPTIONIST','sakibb@gmail.com','123456');

INSERT INTO USER\_TYPE VALUES (702, 602,'RECEPTIONIST','mahmuda007@gmail.com','123456');

INSERT INTO USER\_TYPE VALUES (703, 603,'RECEPTIONIST','tariqur@gmail.com','123456');

INSERT INTO USER\_TYPE VALUES (704, 604,'RECEPTIONIST','tanvir@gmail.com','123456');

INSERT INTO USER\_TYPE VALUES (705, 605,'RECEPTIONIST','anika@gmail.com','123456');

INSERT INTO USER\_TYPE VALUES (706, 1,'ADMIN','sshihabb007@gmail.com','123456');

CREATE TABLE ASSIGNED\_ROOM

(AR\_ID NUMBER (7) PRIMARY KEY,

P\_ID NUMBER (7) NOT NULL,

ROOM\_NO NUMBER(7) NOT NULL);

INSERT INTO ASSIGNED\_ROOM VALUES (801,101,5);

INSERT INTO ASSIGNED\_ROOM VALUES (802,103,6);

INSERT INTO ASSIGNED\_ROOM VALUES (803,104,7);

INSERT INTO ASSIGNED\_ROOM VALUES (804,106,8);

INSERT INTO ASSIGNED\_ROOM VALUES (805,107,2);

CREATE TABLE ASSIST\_PATIENT

(AP\_ID NUMBER (7) PRIMARY KEY,

P\_ID NUMBER (7) NOT NULL,

R\_ID NUMBER (7) NOT NULL);

INSERT INTO ASSIST\_PATIENT VALUES (901,106,601);

INSERT INTO ASSIST\_PATIENT VALUES (902,101,601);

INSERT INTO ASSIST\_PATIENT VALUES (903,107,602);

INSERT INTO ASSIST\_PATIENT VALUES (904,103,603);

INSERT INTO ASSIST\_PATIENT VALUES (905,105,604);

INSERT INTO ASSIST\_PATIENT VALUES (906,102,605);

INSERT INTO ASSIST\_PATIENT VALUES (907,104,604);

CREATE TABLE GOVERNS\_ROOM

(GR\_ID NUMBER (7) PRIMARY KEY,

N\_ID NUMBER (7) NOT NULL,

ROOM\_NO NUMBER(7) NOT NULL);

INSERT INTO GOVERNS\_ROOM VALUES (201,501,1);

INSERT INTO GOVERNS\_ROOM VALUES (202,502,2);

INSERT INTO GOVERNS\_ROOM VALUES (203,503,3);

INSERT INTO GOVERNS\_ROOM VALUES (204,502,4);

INSERT INTO GOVERNS\_ROOM VALUES (205,503,5);

INSERT INTO GOVERNS\_ROOM VALUES (206,504,6);

INSERT INTO GOVERNS\_ROOM VALUES (207,505,7);

INSERT INTO GOVERNS\_ROOM VALUES (208,505,8);

INSERT INTO GOVERNS\_ROOM VALUES (209,503,9);

CREATE TABLE ATTEND\_PATIENT

(ATP\_ID NUMBER (7) PRIMARY KEY,

P\_ID NUMBER (7) NOT NULL,

D\_ID NUMBER (7) NOT NULL);

INSERT INTO ATTEND\_PATIENT VALUES (1001,106,301);

INSERT INTO ATTEND\_PATIENT VALUES (1002,101,301);

INSERT INTO ATTEND\_PATIENT VALUES (1003,107,302);

INSERT INTO ATTEND\_PATIENT VALUES (1004,103,303);

INSERT INTO ATTEND\_PATIENT VALUES (1005,105,304);

INSERT INTO ATTEND\_PATIENT VALUES (1006,102,305);

CREATE TABLE LOG\_PATIENT

( "PATIENT\_ID" NUMBER NOT NULL ENABLE,

"RELEASE\_DATE" DATE NULL,

"ADMIT\_DATE" DATE NULL,

CONSTRAINT "LOG\_PATIENT\_PK" PRIMARY KEY ("PATIENT\_ID") ENABLE

)

CREATE TABLE LOGIN\_INFO

("LID" NUMBER(20,0) NOT NULL ENABLE,

"LOGGED\_PERSON" NUMBER(20,0),

"LOGIN\_DATE" DATE,

CONSTRAINT "LOGIN\_INFO\_PK" PRIMARY KEY ("LID") )

**VIEWS**

CREATE OR REPLACE FORCE VIEW "PATIENT\_VIEW" ("P\_ID", "P\_NAME", "P\_ADDRESS", "P\_GENDER", "P\_DOB", "P\_CONTACT\_NO", "P\_STATUS", "P\_CONDITION", "ADMIT\_DATE", "RELEASE\_DATE") AS

SELECT P."P\_ID",P."P\_NAME",P."P\_ADDRESS",P."P\_GENDER",P."P\_DOB",P."P\_CONTACT\_NO",P."P\_STATUS",P."P\_CONDITION",L.ADMIT\_DATE,L.RELEASE\_DATE

FROM PATIENT P,LOG\_PATIENT L

WHERE P.P\_ID=L.PATIENT\_ID;

CREATE OR REPLACE FORCE VIEW ASSIST\_PATIENT\_VIEW("P\_ID", "P\_NAME", "R\_ID", "R\_NAME") AS

SELECT P.P\_ID,P\_NAME,R.R\_ID,R\_NAME

FROM PATIENT P,RECEPTIONIST R,ASSIST\_PATIENT A

WHERE P.P\_ID=A.P\_ID AND R.R\_ID=A.R\_ID;

CREATE OR REPLACE FORCE VIEW PATIENT\_ROOM\_VIEW("P\_ID", "P\_NAME", "P\_ADDRESS", "P\_GENDER", "P\_DOB", "P\_CONTACT\_NO", "P\_STATUS", "P\_CONDITION", "ROOM\_NO") AS

select P."P\_ID",P."P\_NAME",P."P\_ADDRESS",P."P\_GENDER",P."P\_DOB",P."P\_CONTACT\_NO",P."P\_STATUS",P."P\_CONDITION", AR.ROOM\_NO

from PATIENT P,ASSIGNED\_ROOM AR

where P.P\_ID=AR.P\_ID AND P.P\_CONDITION='Not Released';

CREATE OR REPLACE FORCE VIEW LOGIN\_VIEW\_REC("U\_TYPE", "LOGIN\_DATE", "R\_NAME", "R\_ID") AS

SELECT U.U\_TYPE,L.LOGIN\_DATE,R.R\_NAME,R\_ID

FROM USER\_TYPE U,LOGIN\_INFO L,RECEPTIONIST R

WHERE U.ID=L.LOGGED\_PERSON AND U.ID=R.R\_ID AND L.LOGGED\_PERSON=R.R\_ID;

CREATE OR REPLACE FORCE VIEW LOGIN\_VIEW\_ADMIN("U\_TYPE", "LOGIN\_DATE", "ADMIN\_NAME") AS

SELECT U.U\_TYPE,L.LOGIN\_DATE,A.ADMIN\_NAME

FROM USER\_TYPE U,LOGIN\_INFO L,ADMIN A

WHERE U.ID=L.LOGGED\_PERSON AND U.ID=A.ADMIN\_ID AND L.LOGGED\_PERSON=A.ADMIN\_ID;

**Sequences**

CREATE SEQUENCE ADDRESS\_SEQ MINVALUE 1 MAXVALUE 999999999999999999999999999 INCREMENT BY 1 START WITH 1 NOCACHE NOORDER NOCYCLE;

CREATE SEQUENCE ADMIN\_SEQ MINVALUE 1 MAXVALUE 999999999999999999999999999 INCREMENT BY 1 START WITH 1 NOCACHE NOORDER NOCYCLE

CREATE SEQUENCE ASSIGNED\_ROOM\_SEQ MINVALUE 1 MAXVALUE 999999999999999999999999999 INCREMENT BY 1 START WITH 1 NOCACHE NOORDER NOCYCLE

CREATE SEQUENCE ASSIST\_PATIENT\_SEQ MINVALUE 1 MAXVALUE 999999999999999999999999999 INCREMENT BY 1 START WITH 1 NOCACHE NOORDER NOCYCLE

CREATE SEQUENCE ATTEND\_PATIENT\_SEQ MINVALUE 1 MAXVALUE 999999999999999999999999999 INCREMENT BY 1 START WITH 1 NOCACHE NOORDER NOCYCLE

CREATE SEQUENCE DOCTOR\_SEQ MINVALUE 1 MAXVALUE 999999999999999999999999999 INCREMENT BY 1 START WITH 306 NOCACHE NOORDER NOCYCLE

CREATE SEQUENCE GOVERNS\_ROOM\_SEQ MINVALUE 1 MAXVALUE 999999999999999999999999999 INCREMENT BY 1 START WITH 1 NOCACHE NOORDER NOCYCLE

CREATE SEQUENCE LOG\_PATIENT\_SEQ MINVALUE 1 MAXVALUE 999999999999999999999999999 INCREMENT BY 1 START WITH 1 NOCACHE NOORDER NOCYCLE

CREATE SEQUENCE NURSE\_SEQ MINVALUE 1 MAXVALUE 999999999999999999999999999 INCREMENT BY 1 START WITH 406 NOCACHE NOORDER NOCYCLE

CREATE SEQUENCE PATIENT\_SEQ MINVALUE 1 MAXVALUE 999999999999999999999999999 INCREMENT BY 1 START WITH 108 NOCACHE NOORDER NOCYCLE

CREATE SEQUENCE RECEPTIONIST\_SEQ MINVALUE 1 MAXVALUE 999999999999999999999999999 INCREMENT BY 1 START WITH 506 NOCACHE NOORDER NOCYCLE

CREATE SEQUENCE ROOM\_SEQ MINVALUE 1 MAXVALUE 999999999999999999999999999 INCREMENT BY 1 START WITH 120 NOCACHE NOORDER NOCYCLE

CREATE SEQUENCE USER\_TYPE\_SEQ MINVALUE 1 MAXVALUE 999999999999999999999999999 INCREMENT BY 1 START WITH 1 NOCACHE NOORDER NOCYCLE

CREATE SEQUENCE LOGIN\_INFO\_SEQ MINVALUE 1 MAXVALUE 999999999999999999999999999 INCREMENT BY 1 START WITH 1 NOCACHE NOORDER NOCYCLE

**Functions**

create or replace FUNCTION FIND\_DOCTOR\_ID(doctorid IN DOCTOR.D\_ID%TYPE)

RETURN BOOLEAN AS

FLAG NUMBER(5):=0;

CURSOR DOC\_REC IS

SELECT D\_ID FROM ATTEND\_PATIENT;

BEGIN

FOR I IN DOC\_REC LOOP

IF I.D\_ID=doctorid THEN

FLAG:=1;

END IF;

END LOOP;

IF FLAG=1 THEN

RETURN TRUE;

ELSE

RETURN FALSE;

END IF;

END FIND\_DOCTOR\_ID;

create or replace FUNCTION FIND\_NURSE\_ID(nurseid IN NURSE.N\_ID%TYPE)

RETURN BOOLEAN AS

FLAG NUMBER(5):=0;

CURSOR NURSE\_REC IS

SELECT N\_ID FROM GOVERNS\_ROOM;

BEGIN

FOR I IN NURSE\_REC LOOP

IF I.N\_ID=nurseid THEN

FLAG:=1;

END IF;

END LOOP;

IF FLAG=1 THEN

RETURN TRUE;

ELSE

RETURN FALSE;

END IF;

END FIND\_NURSE\_ID;

create or replace FUNCTION FIND\_PATIENT\_ASSIGNED\_ROOM(patientid IN ASSIGNED\_ROOM.P\_ID%TYPE)

RETURN BOOLEAN AS

FLAG NUMBER(5):=0;

CURSOR P\_REC IS

SELECT P\_ID FROM ASSIGNED\_ROOM ;

BEGIN

FOR I IN P\_REC LOOP

IF I.P\_ID=patientid THEN

FLAG:=1;

END IF;

END LOOP;

IF FLAG=1 THEN

RETURN TRUE;

ELSE

RETURN FALSE;

END IF;

END FIND\_PATIENT\_ASSIGNED\_ROOM;

create or replace FUNCTION FIND\_PATIENT\_ASSIST\_PATIENT(patientid IN ASSIST\_PATIENT.P\_ID%TYPE)

RETURN BOOLEAN AS

FLAG NUMBER(5):=0;

CURSOR P\_REC IS

SELECT P\_ID FROM ASSIST\_PATIENT ;

BEGIN

FOR I IN P\_REC LOOP

IF I.P\_ID=patientid THEN

FLAG:=1;

END IF;

END LOOP;

IF FLAG=1 THEN

RETURN TRUE;

ELSE

RETURN FALSE;

END IF;

END FIND\_PATIENT\_ASSIST\_PATIENT;

create or replace FUNCTION FIND\_PATIENT\_ATTEND\_PATIENT(patientid IN ATTEND\_PATIENT.P\_ID%TYPE)

RETURN BOOLEAN AS

FLAG NUMBER(5):=0;

CURSOR P\_REC IS

SELECT P\_ID FROM ATTEND\_PATIENT ;

BEGIN

FOR I IN P\_REC LOOP

IF I.P\_ID=patientid THEN

FLAG:=1;

END IF;

END LOOP;

IF FLAG=1 THEN

RETURN TRUE;

ELSE

RETURN FALSE;

END IF;

END FIND\_PATIENT\_ATTEND\_PATIENT;

create or replace FUNCTION FIND\_PATIENT\_LOG\_PATIENT(patientid IN LOG\_PATIENT.PATIENT\_ID%TYPE)

RETURN BOOLEAN AS

FLAG NUMBER(5):=0;

CURSOR P\_REC IS

SELECT PATIENT\_ID FROM LOG\_PATIENT ;

BEGIN

FOR I IN P\_REC LOOP

IF I.PATIENT\_ID=patientid THEN

FLAG:=1;

END IF;

END LOOP;

IF FLAG=1 THEN

RETURN TRUE;

ELSE

RETURN FALSE;

END IF;

END FIND\_PATIENT\_LOG\_PATIENT;

create or replace FUNCTION FIND\_RECEPTIONIST\_ID(rid in RECEPTIONIST.R\_ID%TYPE)

RETURN BOOLEAN AS

FLAG NUMBER(5):=0;

CURSOR RECEPTIONIST\_REC IS

SELECT R\_ID FROM ASSIST\_PATIENT;

BEGIN

FOR I IN RECEPTIONIST\_REC LOOP

IF I.R\_ID=rid THEN

FLAG:=1;

END IF;

END LOOP;

IF FLAG=1 THEN

RETURN TRUE;

ELSE

RETURN FALSE;

END IF;

END FIND\_RECEPTIONIST\_ID;

create or replace FUNCTION FIND\_ROOMNO\_ASSIGNEDROOM(roomid IN ROOM.ROOM\_NO%TYPE)

RETURN BOOLEAN AS

FLAG NUMBER(5):=0;

CURSOR ROOM\_REC IS

SELECT ROOM\_NO FROM ASSIGNED\_ROOM;

BEGIN

FOR I IN ROOM\_REC LOOP

IF I.ROOM\_NO=roomid THEN

FLAG:=1;

END IF;

END LOOP;

IF FLAG=1 THEN

RETURN TRUE;

ELSE

RETURN FALSE;

END IF;

END FIND\_ROOMNO\_ASSIGNEDROOM;

create or replace FUNCTION FIND\_ROOMNO\_GOVERNSROOM(roomid IN ROOM.ROOM\_NO%TYPE)

RETURN BOOLEAN AS

FLAG NUMBER(5):=0;

CURSOR ROOM\_REC IS

SELECT ROOM\_NO FROM GOVERNS\_ROOM;

BEGIN

FOR I IN ROOM\_REC LOOP

IF I.ROOM\_NO=roomid THEN

FLAG:=1;

END IF;

END LOOP;

IF FLAG=1 THEN

RETURN TRUE;

ELSE

RETURN FALSE;

END IF;

END FIND\_ROOMNO\_GOVERNSROOM;

create or replace FUNCTION RECEPTIONIST\_EMAIL(uid in RECEPTIONIST.R\_EMAIL%TYPE)

RETURN RECEPTIONIST.R\_EMAIL%TYPE AS

email RECEPTIONIST.R\_EMAIL%TYPE;

BEGIN

SELECT R\_EMAIL INTO email FROM RECEPTIONIST WHERE R\_ID=uid;

RETURN email;

END RECEPTIONIST\_EMAIL;

create or replace FUNCTION USER\_EMAIL(uid in USER\_TYPE.U\_EMAIL%TYPE)

RETURN USER\_TYPE.U\_EMAIL%TYPE AS

email USER\_TYPE.U\_EMAIL%TYPE;

BEGIN

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

RETURN email;

END USER\_EMAIL;

**Packages**

create or replace PACKAGE DELETE\_PACKAGE AS

PROCEDURE DELETE\_DOCTOR(doctorid in DOCTOR.D\_ID%TYPE) ;

PROCEDURE DELETE\_NURSE(nurseid in NURSE.N\_ID%TYPE) ;

PROCEDURE DELETE\_RECEPTIONIST(recid in RECEPTIONIST.R\_ID%TYPE) ;

PROCEDURE DELETE\_ROOM(roomid in ROOM.ROOM\_NO%TYPE) ;

PROCEDURE DELETE\_PATIENT(patientid IN PATIENT.P\_ID%TYPE);

END DELETE\_PACKAGE;

create or replace PACKAGE BODY DELETE\_PACKAGE AS

PROCEDURE DELETE\_DOCTOR(doctorid in DOCTOR.D\_ID%TYPE) AS

BEGIN

DELETE FROM DOCTOR WHERE D\_ID=doctorid;

IF FIND\_DOCTOR\_ID(doctorid)=TRUE THEN

DELETE FROM ATTEND\_PATIENT WHERE D\_ID=doctorid;

END IF;

END DELETE\_DOCTOR;

PROCEDURE DELETE\_NURSE(nurseid in NURSE.N\_ID%TYPE) AS

BEGIN

DELETE FROM NURSE WHERE N\_ID=nurseid;

IF FIND\_NURSE\_ID(nurseid)=TRUE THEN

DELETE FROM GOVERNS\_ROOM WHERE N\_ID=nurseid;

END IF;

END DELETE\_NURSE;

PROCEDURE DELETE\_RECEPTIONIST(recid in RECEPTIONIST.R\_ID%TYPE) AS

BEGIN

DELETE FROM RECEPTIONIST WHERE R\_ID=recid;

DELETE FROM USER\_TYPE WHERE ID=recid;

IF FIND\_RECEPTIONIST\_ID(recid)=TRUE THEN

DELETE FROM ASSIST\_PATIENT WHERE R\_ID=recid;

END IF;

END DELETE\_RECEPTIONIST;

PROCEDURE DELETE\_ROOM(roomid IN ROOM.ROOM\_NO%TYPE) AS

roomid\_assignedroom ROOM.ROOM\_NO%TYPE;

roomid\_governsroom ROOM.ROOM\_NO%TYPE;

BEGIN

DELETE FROM ROOM WHERE ROOM\_NO=roomid;

IF FIND\_ROOMNO\_ASSIGNEDROOM(roomid)=TRUE THEN

DELETE FROM ASSIGNED\_ROOM WHERE ROOM\_NO=roomid;

END IF;

IF FIND\_ROOMNO\_GOVERNSROOM(roomid)=TRUE THEN

DELETE FROM GOVERNS\_ROOM WHERE ROOM\_NO=roomid;

END IF;

END DELETE\_ROOM;

PROCEDURE DELETE\_PATIENT(patientid IN PATIENT.P\_ID%TYPE) AS

BEGIN

DELETE FROM PATIENT WHERE P\_ID=patientid;

IF FIND\_PATIENT\_ASSIGNED\_ROOM(patientid)=TRUE THEN

DELETE FROM ASSIGNED\_ROOM WHERE P\_ID=patientid;

END IF;

IF FIND\_PATIENT\_ASSIST\_PATIENT(patientid)=TRUE THEN

DELETE FROM ASSIST\_PATIENT WHERE P\_ID=patientid;

END IF;

IF FIND\_PATIENT\_ATTEND\_PATIENT(patientid)=TRUE THEN

DELETE FROM ATTEND\_PATIENT WHERE P\_ID=patientid;

END IF;

IF FIND\_PATIENT\_LOG\_PATIENT(patientid)=TRUE THEN

DELETE FROM LOG\_PATIENT WHERE PATIENT\_ID=patientid;

END IF;

END DELETE\_PATIENT;

END DELETE\_PACKAGE;

create or replace PACKAGE EDIT\_PACKAGE AS

PROCEDURE EDIT\_DOCTOR(id in DOCTOR.D\_ID%TYPE,name in DOCTOR.D\_NAME%TYPE,address in DOCTOR.D\_ADDRESS%TYPE,contact in DOCTOR.D\_CONTACT\_NO%TYPE,salary in DOCTOR.D\_SALARY%TYPE,dob in DOCTOR.D\_JOINING\_DATE%TYPE,designation in DOCTOR.D\_DESIGNATION%TYPE);

PROCEDURE EDIT\_NURSE(id in NURSE.N\_ID%TYPE,name in NURSE.N\_NAME%TYPE,address in NURSE.N\_ADDRESS%TYPE,contact in NURSE.N\_CONTACT\_NO%TYPE,salary in NURSE.N\_SALARY%TYPE,dob in NURSE.N\_JOINING\_DATE%TYPE,designation in NURSE.N\_DESIGNATION%TYPE);

PROCEDURE EDIT\_RECEPTIONIST(id in RECEPTIONIST.R\_ID%TYPE,email in RECEPTIONIST.R\_EMAIL%TYPE,name in RECEPTIONIST.R\_NAME%TYPE,address in RECEPTIONIST.R\_ADDRESS%TYPE,contact in RECEPTIONIST.R\_CONTACT\_NO%TYPE,salary in RECEPTIONIST.R\_SALARY%TYPE,dob in RECEPTIONIST.R\_JOINING\_DATE%TYPE,designation in RECEPTIONIST.R\_DESIGNATION%TYPE);

PROCEDURE EDIT\_ROOM(roomid in ROOM.ROOM\_NO%TYPE,roomtype in ROOM.ROOM\_TYPE%TYPE,roomcost IN ROOM.ROOM\_COST%TYPE);

PROCEDURE EDIT\_PATIENT(id in PATIENT.P\_ID%TYPE,name in PATIENT.P\_NAME%TYPE,contact in PATIENT.P\_CONTACT\_NO%TYPE,address in PATIENT.P\_ADDRESS%TYPE,gender in PATIENT.P\_GENDER%TYPE,dob IN PATIENT.P\_DOB%TYPE);

END EDIT\_PACKAGE;

create or replace PACKAGE BODY EDIT\_PACKAGE AS

PROCEDURE EDIT\_DOCTOR(id in DOCTOR.D\_ID%TYPE,name in DOCTOR.D\_NAME%TYPE,address in DOCTOR.D\_ADDRESS%TYPE,contact in DOCTOR.D\_CONTACT\_NO%TYPE,salary in DOCTOR.D\_SALARY%TYPE,dob in DOCTOR.D\_JOINING\_DATE%TYPE,designation in DOCTOR.D\_DESIGNATION%TYPE) AS

BEGIN

UPDATE DOCTOR SET D\_NAME=name, D\_ADDRESS=address, D\_CONTACT\_NO=contact,D\_SALARY=salary,D\_JOINING\_DATE=DOB,D\_DESIGNATION=designation WHERE D\_ID=id;

END EDIT\_DOCTOR;

PROCEDURE EDIT\_NURSE(id in NURSE.N\_ID%TYPE,name in NURSE.N\_NAME%TYPE,address in NURSE.N\_ADDRESS%TYPE,contact in NURSE.N\_CONTACT\_NO%TYPE,salary in NURSE.N\_SALARY%TYPE,dob in NURSE.N\_JOINING\_DATE%TYPE,designation in NURSE.N\_DESIGNATION%TYPE) AS

BEGIN

UPDATE NURSE SET N\_NAME=name, N\_ADDRESS=address, N\_CONTACT\_NO=contact,N\_SALARY=salary,N\_JOINING\_DATE=DOB,N\_DESIGNATION=designation WHERE N\_ID=id;

END EDIT\_NURSE;

PROCEDURE EDIT\_RECEPTIONIST(id in RECEPTIONIST.R\_ID%TYPE,email in RECEPTIONIST.R\_EMAIL%TYPE,name in RECEPTIONIST.R\_NAME%TYPE,address in RECEPTIONIST.R\_ADDRESS%TYPE,contact in RECEPTIONIST.R\_CONTACT\_NO%TYPE,salary in RECEPTIONIST.R\_SALARY%TYPE,dob in RECEPTIONIST.R\_JOINING\_DATE%TYPE,designation in RECEPTIONIST.R\_DESIGNATION%TYPE) AS

oldemail RECEPTIONIST.R\_EMAIL%TYPE;

newemail RECEPTIONIST.R\_EMAIL%TYPE;

userid USER\_TYPE.ID%TYPE:=id;

BEGIN

oldemail:=RECEPTIONIST\_EMAIL(id);

UPDATE RECEPTIONIST SET R\_NAME=name,R\_EMAIL=email, R\_ADDRESS=address, R\_CONTACT\_NO=contact,R\_SALARY=salary,R\_JOINING\_DATE=DOB,R\_DESIGNATION=designation WHERE R\_ID=id;

newemail:=RECEPTIONIST\_EMAIL(id);

IF oldemail!=newemail THEN

UPDATE USER\_TYPE SET U\_EMAIL=newemail WHERE ID=userid;

END IF;

END EDIT\_RECEPTIONIST;

PROCEDURE EDIT\_ROOM(roomid in ROOM.ROOM\_NO%TYPE,roomtype in ROOM.ROOM\_TYPE%TYPE,roomcost IN ROOM.ROOM\_COST%TYPE) AS

BEGIN

UPDATE ROOM SET ROOM\_TYPE=roomtype,ROOM\_COST=roomcost WHERE ROOM\_NO=roomid;

END EDIT\_ROOM;

PROCEDURE EDIT\_PATIENT(id in PATIENT.P\_ID%TYPE,name in PATIENT.P\_NAME%TYPE,contact in PATIENT.P\_CONTACT\_NO%TYPE,address in PATIENT.P\_ADDRESS%TYPE,gender in PATIENT.P\_GENDER%TYPE,dob IN PATIENT.P\_DOB%TYPE) AS

BEGIN

UPDATE PATIENT SET P\_NAME=name,P\_CONTACT\_NO=contact,P\_ADDRESS=address,P\_GENDER=gender,P\_DOB=dob where P\_ID=id;

END EDIT\_PATIENT;

END EDIT\_PACKAGE;

**Procedures:**

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;

**Triggers:**

CREATE OR REPLACE TRIGGER PATIENT\_LOG

AFTER INSERT ON PATIENT

REFERENCING OLD AS OLD NEW AS NEW

FOR EACH ROW

BEGIN

INSERT INTO LOG\_PATIENT(PATIENT\_ID,ADMIT\_DATE) VALUES(:NEW.P\_ID,SYSDATE);

END;