



American International University- Bangladesh

ADVANCE DATABASE MANAGEMENT SYSTEM

SECTION: B

GROUP NAME: ROYAL BLUE

Mid-Term Project

Fall 20-21

Project Topic: Hospital Management System

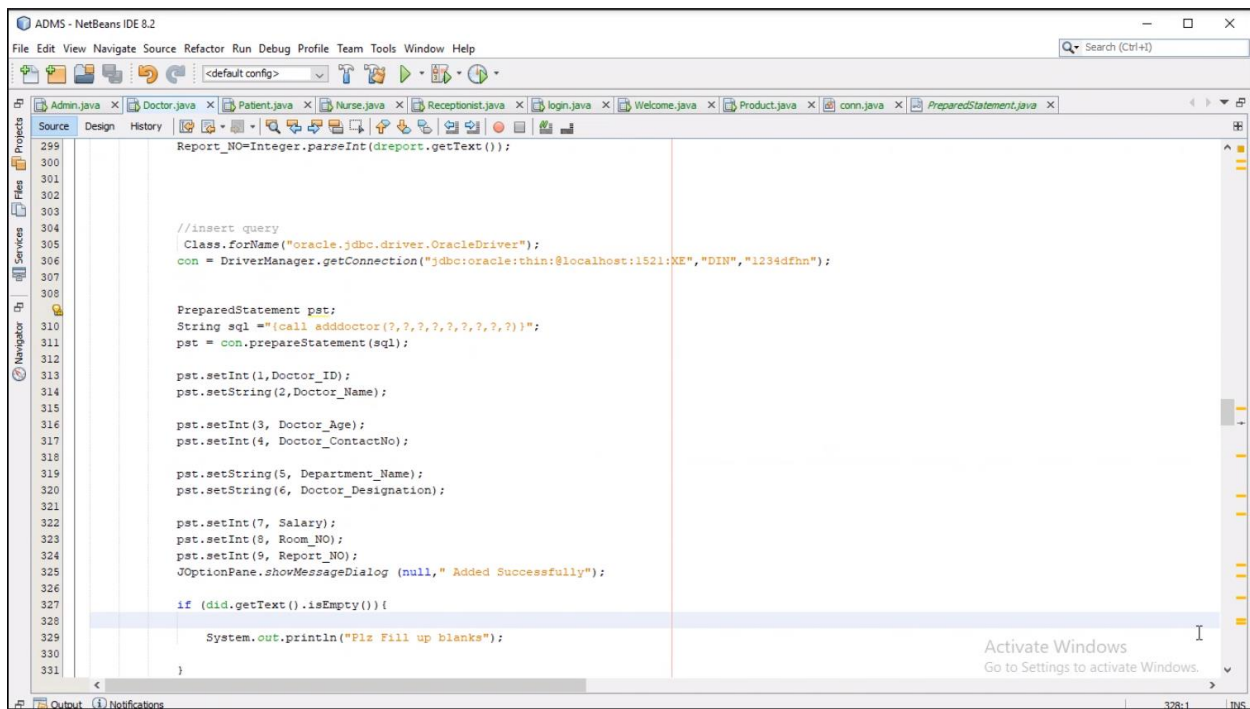
With respectfully submitted to Rezwan Ahmed

Student Name	Student ID
DIN MOHAMMAD DOHAN	17-34465-2
SHAILY SARKER	17-35951-3
SAMIA ISLAM OYSE	18-37543-1
SAYED MAHMUDUL HASSAN	18-38095-2

Final Project Deliverable

Question No 01:

Yes, Oracle database has been used successfully.



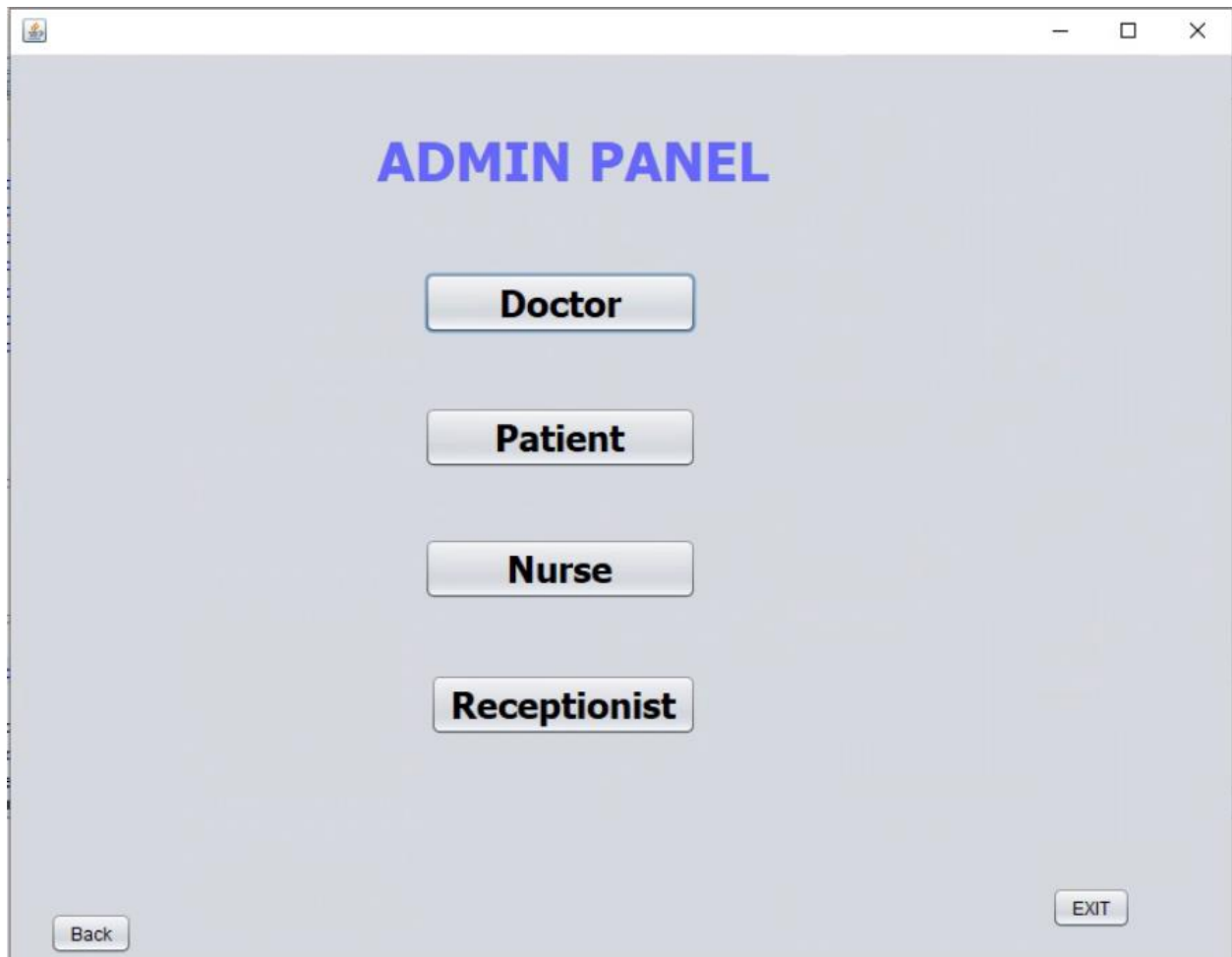
The screenshot displays the NetBeans IDE 8.2 interface. The main editor window shows a Java file named `PreparedStatement.java` with the following code:

```
299 Report_NO=Integer.parseInt(dreport.getText());
300
301
302
303
304 //insert query
305 Class.forName("oracle.jdbc.driver.OracleDriver");
306 con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE","DIN","1234dfnn");
307
308
309 PreparedStatement pst;
310 String sql ="(call adddoctor(?,?,?,?,?,?,?,?))";
311 pst = con.prepareStatement(sql);
312
313 pst.setInt(1,Doctor_ID);
314 pst.setString(2,Doctor_Name);
315
316 pst.setInt(3, Doctor_Age);
317 pst.setInt(4, Doctor_ContactNo);
318
319 pst.setString(5, Department_Name);
320 pst.setString(6, Doctor_Designation);
321
322 pst.setInt(7, Salary);
323 pst.setInt(8, Room_NO);
324 pst.setInt(9, Report_NO);
325 JOptionPane.showMessageDialog (null," Added Successfully");
326
327 if (did.getText().isEmpty()){
328
329     System.out.println("Plz Fill up blanks");
330
331 }
```

The IDE interface includes a menu bar (File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help), a toolbar, and a project explorer on the left. The status bar at the bottom shows "Output" and "Notifications".

Question No 02:

Yes, we have created Admin Panel.



Question No 03:

Yes, we have maintained proper interface to communicate with database.





A screenshot of a graphical user interface window titled "Login Form". The window has a standard title bar with a small icon on the left and minimize, maximize, and close buttons on the right. The main area has a light gray background. At the top center, the text "Login Form" is displayed in a bold, blue, serif font. Below this, there are two labels: "Username:" and "Password:", both in a bold, black, serif font. To the right of "Username:" is a white rectangular text input field with a thin blue border. To the right of "Password:" is a white rectangular text input field with a thin gray border. Below the "Password:" field and to the right, there is a rounded rectangular button with a light gray gradient and a thin gray border, containing the text "Login" in a bold, black, serif font. At the bottom left of the window is another rounded rectangular button with a light gray gradient and a thin gray border, containing the text "Back" in a bold, black, serif font. At the bottom right is a third rounded rectangular button with a light gray gradient and a thin gray border, containing the text "Exit" in a bold, black, serif font.

Login Form

Username:

Password

Login

Back **Exit**

— □ ×

Login Form

Username:

SHUVRO

Password

123456

Message ×

i

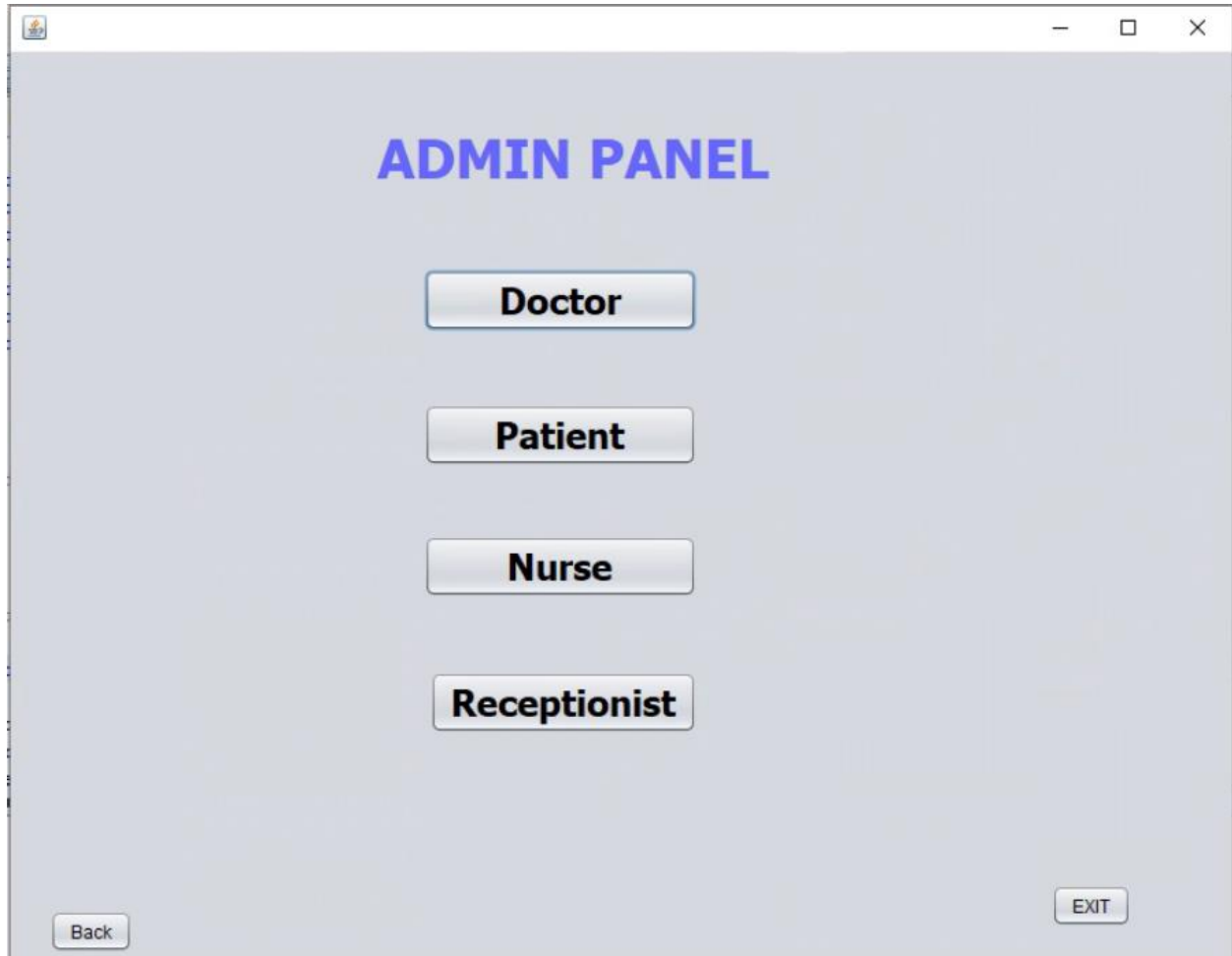
Invalid Username or Password!

OK

Login

Back

Exit



Doctor Details

Doctor_ID:	<input type="text"/>	Department_Name:	<input type="text"/>	<input type="button" value="Search"/>
Doctor_Name:	<input type="text"/>	Doctor_Designation:	<input type="text"/>	<input type="button" value="Add Doctor"/>
Doctor_Age:	<input type="text"/>	Salary:	<input type="text"/>	<input type="button" value="Remove Doctor"/>
Doctor_ContactNo:	<input type="text"/>	Room_NO:	<input type="text"/>	<input type="button" value="Update"/>
		Report_NO:	<input type="text"/>	

Activate Windows
Go to Settings to activate Windows.

Doctor Details

Doctor_ID:	<input type="text" value="103"/>	Department_Name:	<input type="text" value="Cardiology"/>	<input type="button" value="Search"/>
Doctor_Name:	<input type="text" value="Dr.Saifulla Khan"/>	Doctor_Designation:	<input type="text" value="Assistant"/>	<input type="button" value="Add Doctor"/>
Doctor_Age:	<input type="text" value="50"/>	Salary:	<input type="text" value="50000"/>	<input type="button" value="Remove Doctor"/>
Doctor_ContactNo:	<input type="text" value="1821212"/>	Room_NO:	<input type="text" value="503"/>	<input type="button" value="Update"/>
		Report_NO:	<input type="text" value="303"/>	

Patient Details

Patient_ID:	<input type="text"/>	Patient_Address:	<input type="text"/>	<input type="button" value="Search"/>
Patient_Name:	<input type="text"/>	Department_Name:	<input type="text"/>	<input type="button" value="Add Patient"/>
Patient_Age:	<input type="text"/>	Room_NO:	<input type="text"/>	<input type="button" value="Remove Patient"/>
Patient_ContactNo:	<input type="text"/>	Report_NO:	<input type="text"/>	<input type="button" value="Update"/>

Activate Windows

Patient Details

Patient_ID:	<input type="text" value="3"/>	Patient_Address:	<input type="text" value="Badda"/>	<input type="button" value="Search"/>
Patient_Name:	<input type="text" value="Sara"/>	Department_Name:	<input type="text" value="Cardiology"/>	<input type="button" value="Add Patient"/>
Patient_Age:	<input type="text" value="22"/>	Room_NO:	<input type="text" value="503"/>	<input type="button" value="Remove Patient"/>
Patient_ContactNo:	<input type="text" value="1521322246"/>	Report_NO:	<input type="text" value="303"/>	<input type="button" value="Update"/>

Activate Windows

Nurse Details

Nurse_ID:	<input type="text"/>	Department_Name:	<input type="text"/>	<input type="button" value="Search"/>
Nurse_Name:	<input type="text"/>	Nurse_Designation:	<input type="text"/>	<input type="button" value="Add"/>
Nurse_Age:	<input type="text"/>	Salary:	<input type="text"/>	<input type="button" value="Remove"/>
Nurse_ContactNo:	<input type="text"/>	Room_NO :	<input type="text"/>	<input type="button" value="Update"/>

Nurse Details

Nurse_ID:	<input type="text" value="204"/>	Department_Name:	<input type="text" value="Cardiology"/>	<input type="button" value="Search"/>
Nurse_Name:	<input type="text" value="Asif Alam"/>	Nurse_Designation:	<input type="text" value="Junior Nurse"/>	<input type="button" value="Add"/>
Nurse_Age:	<input type="text" value="27"/>	Salary:	<input type="text" value="50000"/>	<input type="button" value="Remove"/>
Nurse_ContactNo:	<input type="text" value="1899212"/>	Room_NO :	<input type="text" value="504"/>	<input type="button" value="Update"/>

Receptionist Details

Receptionist_ID:

Receptionist_Uname:

Search

Receptionist_Name:

Receptionist_Designation:

Add

Receptionist_Age:

Salary:

Remove

Update

Display

Receptionist_ID	Receptionist_Name	Receptionist_Age	Receptionist_Uname	Receptionist_Designation	Salary
-----------------	-------------------	------------------	--------------------	--------------------------	--------

Back

Exit

Receptionist Details

Receptionist_ID:

Receptionist_Uname:

Search

Receptionist_Name:

Receptionist_Designation:

Add

Receptionist_Age:

Salary:

Remove

Update

Display

Receptionist_ID	Receptionist_Name	Receptionist_Age	Receptionist_Uname	Receptionist_Designation	Salary
-----------------	-------------------	------------------	--------------------	--------------------------	--------

Back

Exit

Question No 04:

Different type of searching and advanced searching (SQL Queries)

1. Display NAME, GENDER of the RECEPTIONISTS from RECEPTIONIST table whose Designation As 'Senior'.

Answer: Select RECEPTIONIST_ID, RECEPTIONIST_NAME, RECEPTIONIST_GENDER from RECEPTIONIST WHERE RECEPTIONIST_ID IN (SELECT RECEPTIONIST_ID from RECEPTIONIST where RECEPTIONIST_DESIGNATION ='Senior');

2. Update NURSE SALARY from NURSE table which NURSE age is 28, they will get extra 1000 with their SALARY.

Answer: UPDATE NURSE SET SALARY = SALARY + 1000 WHERE NURSE_AGE IN (SELECT NURSE_AGE FROM NURSE WHERE NURSE_AGE=28);

3. Find out the list of whose working at same DEPARTMENT as 'Cardiology from DOCTOR & NURSE table.

Answer: SELECT 'DOCTOR' AS TYPE, Doctor_Name, Department_Name FROM DOCTOR WHERE Department_Name = 'Cardiology' UNION SELECT 'NURSE', NURSE_Name, Department_Name FROM NURSE WHERE Department_Name = 'Cardiology' order by Department_Name;

4. Find when Patient admit into hospital

Answer:

```
BEGIN
FOR PATIENT IN (SELECT PATIENT_NAME, AdmitDate
FROM PATIENT)
LOOP
dbms_output.put_line (PATIENT.PATIENT_NAME || ':' ||
PATIENT.AdmitDate);
END LOOP;
END;
```

5. Find the Admin id where the id is 0 to 4

Answer:

```
DECLARE
n_ADMIN_ID ADMIN.ADMIN_ID%TYPE := 0;
BEGIN
LOOP
n_ADMIN_ID := n_ADMIN_ID + 1;
IF
n_ADMIN_ID >= 4 THEN
EXIT;
END IF;
dbms_output.put_line('ADMIN' || ':' || n_ADMIN_ID );
END LOOP;
END;
```

6. Find the doctor name where the doctor id is 108

Answer:

```
DECLARE
d_DOCTOR DOCTOR%ROWTYPE;
BEGIN
SELECT * INTO d_DOCTOR
FROM DOCTOR
WHERE DOCTOR_ID = 108;
dbms_output.put_line(d_DOCTOR.DOCTOR_NAME || ':' ||
d_DOCTOR.DOCTOR_ID );
END;
```

Question No 05:

Use sequence to maintain primary key in at least 2 tables.

1)

```
CREATE SEQUENCE doctor_seq  
INCREMENT BY 10  
START WITH 101  
MINVALUE 10  
MAXVALUE 200  
CYCLE  
CACHE 2;
```

```
INSERT INTO DOCTOR  
VALUES (doctor_seq.nextval, 'Dr. Khan', 50, 01821212, 'Cardiology', 'Assistant',  
50000, 300, 901);
```

```
INSERT INTO DOCTOR  
VALUES (doctor_seq.nextval, 'Dr. Mahmud', 50, 01821212, 'Cardiology', 'Assistant',  
50000, 300, 901);
```

```
Select *from DOCTOR;
```

```

CREATE SEQUENCE doctor_seq
  INCREMENT BY 10
  START WITH 101
  MINVALUE 10
  MAXVALUE 200
  CYCLE
  CACHE 2;

INSERT INTO DOCTOR
VALUES
  (doctor_seq.nextval,'Dr. Mahmud',50,01821212,'Cardiology', 'Assistant',50000,300,901)

select *from DOCTOR

```

Results Explain Describe Saved SQL History

DOCTOR_ID	DOCTOR_NAME	DOCTOR_AGE	DOCTOR_CONTACTNO	DEPARTMENT_NAME	DOCTOR_DESIGNATION	SALARY	ROOM_NO	REPORT_NO
101	Dr.Hridita Sraon	27	13546436	Cardiology	Assistant	50000	300	901
108	Dr.Rashed Islam	45	1721212	Diagnostic	Assistant Professor	60000	305	902
115	Dr.Ripa	41	1321212	Orthopedics	Assistant professor	45000	303	903
111	Dr.Seemaa	38	11721212	Neurology	Professor	65000	301	904
198	Dr. Uzzal Kumar	55	136255212	Neurology	Professor	55000	302	908
128	Dr. Hasan Mahabub	51	166255212	Cardiology	Assistant Professor	55000	304	905
200	Dr. Samia	27	187683454	Cardiology	Lecturer	40000	300	901
121	Dr. Khan	50	1821212	Cardiology	Assistant	50000	300	901
131	Dr. Mahmud	50	1821212	Cardiology	Assistant	50000	300	901

```

INSERT INTO DOCTOR
VALUES
  (doctor_seq.nextval,'Dr. Khan',50,01821212,'Cardiology', 'Assistant',50000,300,901)

select *from DOCTOR

```

Results Explain Describe Saved SQL History

DOCTOR_ID	DOCTOR_NAME	DOCTOR_AGE	DOCTOR_CONTACTNO	DEPARTMENT_NAME	DOCTOR_DESIGNATION	SALARY	ROOM_NO	REPORT_NO
101	Dr.Hridita Sraon	27	13546436	Cardiology	Assistant	50000	300	901
108	Dr.Rashed Islam	45	1721212	Diagnostic	Assistant Professor	60000	305	902
115	Dr.Ripa	41	1321212	Orthopedics	Assistant professor	45000	303	903
111	Dr.Seemaa	38	11721212	Neurology	Professor	65000	301	904
198	Dr. Uzzal Kumar	55	136255212	Neurology	Professor	55000	302	908
128	Dr. Hasan Mahabub	51	166255212	Cardiology	Assistant Professor	55000	304	905
200	Dr. Samia	27	187683454	Cardiology	Lecturer	40000	300	901
121	Dr. Khan	50	1821212	Cardiology	Assistant	50000	300	901
131	Dr. Mahmud	50	1821212	Cardiology	Assistant	50000	300	901

9 rows returned in 0.00 seconds [Download](#)

2) CREATE SEQUENCE patient_seq

INCREMENT BY 5

START WITH 201

MINVALUE 10

MAXVALUE 300

NOCYCLE

CACHE 2;

INSERT INTO PATIENT

VALUES (patient_seq.nextval,' Abcde', 28, 01655223246,'khilkhet','Orthopedic', 304, 907);

Select *from PATIENT;

```
CREATE SEQUENCE patient_seq
  INCREMENT BY 5
  START WITH 201
  MINVALUE 10
  MAXVALUE 300
  NOCYCLE
  CACHE 2;
```

Results Explain Describe Saved SQL History

Sequence created.

```
INSERT INTO PATIENT
VALUES
(patient_seq.nextval, 'Abcde', 28, 01655223246, 'khilkhet', 'Orthopedic', 304, 907)
```

Results Explain Describe Saved SQL History

1 row(s) inserted.

Question No 06:

Minimum 5/6 view (min 2 complex among them)

View:

1)

```
CREATE OR REPLACE VIEW PatientInfo_View
AS
SELECT Patient_Name ,Patient_ID, AdmitDate,Report_NO
FROM PATIENT
WHERE Room_NO = 300;
```

```
SELECT * FROM PatientInfo_View;
```

```
CREATE OR REPLACE VIEW PatientInfo_View
AS
SELECT Patient_Name , Patient_ID, AdmitDate,Report_NO
FROM PATIENT
WHERE Room_NO = 300;
```

```
SELECT * FROM PatientInfo_View
```

Results Explain Describe Saved SQL History

View created.

```
CREATE OR REPLACE VIEW PatientInfo_View
AS
SELECT Patient_Name , Patient_ID, AdmitDate,Report_NO
FROM PATIENT
WHERE Room_NO = 300;
```

```
SELECT * FROM PatientInfo_View
```

Results Explain Describe Saved SQL History

PATIENT_NAME	PATIENT_ID	ADMITDATE	REPORT_NO
Samia Islam	201	21-july2020	901
Saiful Khan	206	22-april2020	905

2 rows returned in 0.00 seconds

[Download](#)

2)

```
CREATE OR REPLACE VIEW Doctor_detp
AS
SELECT DOCTOR.DOCTOR_NAME,DOCTOR.DOCTOR_ID,DOCTOR.Salary,
CAST(DOCTOR.Doctor_JoiningDate AS Date) AS Hireing_DATE,
DEPARTMENT.Department_Name ASNameofDepartment
FROM DOCTOR
JOIN Department ON
DOCTOR.Department_Name = DEPARTMENT.Department_Name

SELECT * FROM Doctor_detp;
```

```
CREATE OR REPLACE VIEW Doctor_detp
AS
SELECT DOCTOR.DOCTOR_NAME,DOCTOR.DOCTOR_ID,DOCTOR.Salary,
CAST(DOCTOR.Doctor_JoiningDate AS Date) AS Hireing_DATE,
DEPARTMENT.Department_Name AS NameofDepartment
FROM DOCTOR
JOIN Department ON
DOCTOR.Department_Name = DEPARTMENT.Department_Name
SELECT * FROM Doctor_detp
```

Results Explain Describe Saved SQL History

DOCTOR_NAME	DOCTOR_ID	SALARY	HIREING_DATE	NAMEOFDEPARTMENT
Dr.Saifulla Khan	101	50000	02/05/2095	Cardiology
Dr.Rashed Islam	108	60000	03/09/2092	Diagnostic
Dr.Ripa	115	45000	11/05/2095	Orthopedics
Dr.Seemaa	111	65000	01/13/2099	Neurology
Dr. Uzzal Kumar	198	55000	11/13/2092	Neurology
Dr. Hasan Mahabub	128	55000	07/28/2099	Cardiology

6 rows returned in 0.12 seconds

[Download](#)

3)

```
CREATE OR REPLACE VIEW nurse_salRange  
AS  
SELECT NURSE_NAME, AVG(salary) average_salary  
FROM NURSE  
GROUP BY NURSE_NAME  
HAVING AVG(salary) > 12000;
```

```
SELECT * FROM nurse_salRange;
```

```
CREATE OR REPLACE VIEW nurse_salRange  
AS  
SELECT NURSE_NAME, AVG(salary) average_salary  
FROM NURSE  
GROUP BY NURSE_NAME  
HAVING AVG(salary) > 12000;  
SELECT * FROM nurse_salRange
```

Results Explain Describe Saved SQL History

NURSE_NAME	AVERAGE_SALARY
Liya	13000
Keya	13000

2 rows returned in 0.00 seconds [Download](#)

4)

```
CREATE OR REPLACE VIEW RoomList(Doctor_ID,Patient_ID,Nurse_ID ,Room_NO)
AS
SELECT Doctor.Doctor_ID,PATIENT.Patient_ID,NURSE.Nurse_ID,ROOM.Room_NO
FROM DOCTOR, PATIENT,NURSE,ROOM
WHERE ROOM.Room_NO = 302 ;
SELECT * FROM RoomList
```

```
CREATE OR REPLACE VIEW RoomList
(Doctor_ID,Patient_ID,Nurse_ID ,Room_NO)
AS
SELECT Doctor.Doctor_ID,PATIENT.Patient_ID,NURSE.Nurse_ID, ROOM.Room_NO
FROM DOCTOR, PATIENT, NURSE, ROOM
WHERE ROOM.Room_NO = 302 ;
SELECT * FROM RoomList
```

Results Explain Describe Saved SQL History

DOCTOR_ID	PATIENT_ID	NURSE_ID	ROOM_NO
101	201	501	302
101	201	502	302
101	201	506	302
101	201	514	302
101	201	527	302
101	201	533	302
101	201	555	302
101	202	501	302
101	202	502	302
101	202	506	302
More than 10 rows available. Increase rows selector to view more rows.			

5)

```
CREATE OR REPLACE VIEW NURSE_View
```

```
AS
```

```
SELECT Nurse_Name, Salary, Nurse_AGE, Department_Name, Nurse_Designation
```

```
FROM NURSE
```

```
WHERE Nurse_AGE >= 23;
```

```
SELECT * FROM NURSE_View;
```

```
CREATE OR REPLACE VIEW NURSE_View
```

```
AS
```

```
SELECT Nurse_Name, Salary, Nurse_AGE, Department_Name, Nurse_Designation
```

```
FROM NURSE
```

```
WHERE Nurse_AGE >= 23;
```

```
SELECT * FROM NURSE_View
```

Results Explain Describe Saved SQL History

NURSE_NAME	SALARY	NURSE_AGE	DEPARTMENT_NAME	NURSE_DESIGNATION
Nur	12000	28	Orthopedicsâ€™™	Junior Nurse
Joy	11000	23	Diagnostic	Junior Nurse

2 rows returned in 0.00 seconds

[Download](#)

6)

```
CREATE OR REPLACE FORCE VIEW joining_Date
```

```
AS
```

```
SELECT Receptionist_ID,Receptionist_Name,Receptionist_Age,Salary
```

```
FROM RECEPTIONIST
```

```
where Receptionist_JoiningDate between to_date('01-JAN-2007','DD-MM-YY') and
```

```
to_date('15-DEC-2007','DD-MM-YY');
```

```
SELECT * FROM joining_Date
```

```
CREATE OR REPLACE FORCE VIEW joining_Date
AS
SELECT Receptionist_ID, Receptionist_Name, Receptionist_Age, Salary
FROM RECEPTIONIST
where Receptionist_JoiningDate between to_date('01-JAN-2007','DD-MM-YY') and to_date('15-DEC-2007','DD-MM-YY');

SELECT * FROM joining_Date
```

Results Explain Describe Saved SQL History

RECEPTIONIST_ID	RECEPTIONIST_NAME	RECEPTIONIST_AGE	SALARY
704	Alif	30	19000
705	Sajib	26	11000
706	Rahim	28	11000

3 rows returned in 0.09 seconds [Download](#)

Question No 07:

Minimum 6 procedures and functions which must be embedded in the system

Procedures and functions:

1) Add for Doctor

```
CREATE OR REPLACE PROCEDURE adddoctor (did IN DOCTOR.DOCTOR_ID%TYPE,dname IN
DOCTOR.DOCTOR_NAME%TYPE,dage IN DOCTOR.DOCTOR_AGE%TYPE,dcont IN
DOCTOR.DOCTOR_CONTACTNO%TYPE,ddept IN DOCTOR.DEPARTMENT_NAME%TYPE,ddesig IN
DOCTOR.DOCTOR_DESIGNATION%TYPE,dsal IN DOCTOR.SALARY%TYPE,droom IN
DOCTOR.ROOM_NO%TYPE,dreport IN DOCTOR.REPORT_NO%TYPE)
IS
BEGIN
INSERT INTO Doctor (DOCTOR_ID, DOCTOR_NAME, DOCTOR_AGE, DOCTOR_CONTACTNO,
DEPARTMENT_NAME, DOCTOR_DESIGNATION,SALARY,ROOM_NO,REPORT_NO)
        values (did,dname,dage,dcont,ddept,ddesig,dsal,droom,dreport);
DBMS_OUTPUT.PUT_LINE('DONE DONE DONE ');

END;
```

```
CREATE OR REPLACE PROCEDURE adddoctor (did IN DOCTOR.DOCTOR_ID%TYPE,dname IN DOCTOR.DOCTOR_NAME%TYPE,dage IN DOCTOR.DOCTOR_AGE%TYPE,dcont IN
DOCTOR.DOCTOR_CONTACTNO%TYPE,ddept IN DOCTOR.DEPARTMENT_NAME%TYPE,ddesig IN DOCTOR.DOCTOR_DESIGNATION%TYPE,dsal IN DOCTOR.SALARY%TYPE,droom IN
DOCTOR.ROOM_NO%TYPE,dreport IN DOCTOR.REPORT_NO%TYPE)
IS
BEGIN
INSERT INTO Doctor (DOCTOR_ID,DOCTOR_NAME,DOCTOR_AGE, DOCTOR_CONTACTNO,DEPARTMENT_NAME, DOCTOR_DESIGNATION,SALARY,ROOM_NO,REPORT_NO)
        values (did,dname,dage,dcont,ddept,ddesig,dsal,droom,dreport); DBMS_OUTPUT.PUT_LINE('DONE DONE DONE ');
END;
```

Results Explain Describe Saved SQL History

Procedure created.

2) Delete for Doctor

```
CREATE OR REPLACE PROCEDURE delete_doctor (did IN DOCTOR.DOCTOR_ID%TYPE)
```

```
AS
```

```
BEGIN
```

```
DELETE from Doctor WHERE DOCTOR_ID =did;
```

```
END;
```

```
CREATE OR REPLACE PROCEDURE delete_doctor (did IN DOCTOR.DOCTOR_ID%TYPE)
AS
BEGIN
DELETE from Doctor WHERE DOCTOR_ID =did ;
END;
```

Results Explain Describe Saved SQL History

Procedure created.

3) Update for Doctor

```
CREATE OR REPLACE PROCEDURE UPDATE_DOCTOR (did IN DOCTOR.DOCTOR_ID%TYPE, dname
IN DOCTOR.DOCTOR_NAME%TYPE,dage IN DOCTOR.DOCTOR_AGE%TYPE,dcont IN
DOCTOR.DOCTOR_CONTACTNO%TYPE, ddept IN DOCTOR.DEPARTMENT_NAME%TYPE, ddesig
IN DOCTOR.DOCTOR_DESIGNATION%TYPE, dsal IN DOCTOR.SALARY%TYPE,droom IN
DOCTOR.ROOM_NO%TYPE,dreport IN DOCTOR.REPORT_NO%TYPE)
```

```
AS
```

```
BEGIN
```

```
UPDATE DOCTOR SET
DOCTOR_NAME=dname,DOCTOR_AGE=dage,DOCTOR_CONTACTNO=dcont,
DEPARTMENT_NAME=ddept, DOCTOR_DESIGNATION=ddesig, SALARY=dsal,
ROOM_NO=droom,REPORT_NO=dreport where DOCTOR_ID=did;
```

```
END;
```

```
CREATE OR REPLACE PROCEDURE UPDATE_DOCTOR(did IN DOCTOR.DOCTOR_ID%TYPE, dname IN DOCTOR.DOCTOR_NAME%TYPE,dage IN DOCTOR.DOCTOR_AGE%TYPE,dcont IN
DOCTOR.DOCTOR_CONTACTNO%TYPE,ddept IN DOCTOR.DEPARTMENT_NAME%TYPE,ddesig IN DOCTOR.DOCTOR_DESIGNATION%TYPE,dsal IN
DOCTOR.SALARY%TYPE,droom IN DOCTOR.ROOM_NO%TYPE,dreport IN DOCTOR.REPORT_NO%TYPE)
AS
BEGIN
    UPDATE DOCTOR SET DOCTOR_NAME=dname, DOCTOR_AGE=dage, DOCTOR_CONTACTNO=dcont, DEPARTMENT_NAME=ddept, DOCTOR_DESIGNATION=ddesig, SALARY=dsal,
    ROOM_NO=droom ,REPORT_NO=dreport where DOCTOR_ID=did;
END;
```

Results Explain Describe Saved SQL History

Procedure created.

4) Add for Patient

```
CREATE OR REPLACE PROCEDURE addpatient (pid IN PATIENT.PATIENT_ID%TYPE,  
pname IN PATIENT.PATIENT_NAME%TYPE,  
page IN PATIENT.PATIENT_AGE%TYPE,  
pcont IN PATIENT.PATIENT_CONTACTNO%TYPE,  
padd IN PATIENT.PATIENT_ADDRESS%TYPE,  
pdept IN PATIENT.DEPARTMENT_NAME%TYPE,  
proom IN PATIENT.ROOM_NO%TYPE,  
preport IN PATIENT.REPORT_NO%TYPE)  
IS  
BEGIN  
INSERT INTO PATIENT  
(PATIENT_ID,PATIENT_NAME,PATIENT_AGE,PATIENT_CONTACTNO,PATIENT_ADDRESS,DEP  
ARTMENT_NAME,ROOM_NO,REPORT_NO)  
values (pid,pname,page,pcont,padd,pdept,proom,preport);  
DBMS_OUTPUT.PUT_LINE('DONE DONE DONE ');  
END;
```

```
CREATE OR REPLACE PROCEDURE addpatient (pid IN PATIENT.PATIENT_ID%TYPE,  
pname IN PATIENT.PATIENT_NAME%TYPE,  
page IN PATIENT.PATIENT_AGE%TYPE,  
pcont IN PATIENT.PATIENT_CONTACTNO%TYPE,  
padd IN PATIENT.PATIENT_ADDRESS%TYPE,  
pdept IN PATIENT.DEPARTMENT_NAME%TYPE,  
proom IN PATIENT.ROOM_NO%TYPE,  
preport IN PATIENT.REPORT_NO%TYPE)  
IS  
BEGIN  
INSERT INTO PATIENT (PATIENT_ID,PATIENT_NAME,PATIENT_AGE,PATIENT_CONTACTNO,PATIENT_ADDRESS,DEPARTMENT_NAME,ROOM_NO,REPORT_NO)  
values (pid,pname,page,pcont,padd,pdept,proom,preport);  
DBMS_OUTPUT.PUT_LINE('DONE DONE DONE ');  
END;
```

[Results](#) [Explain](#) [Describe](#) [Saved SQL](#) [History](#)

Procedure created.

5) Delete for Patient

```
CREATE OR REPLACE PROCEDURE delete_patient (pid IN PATIENT.PATIENT_ID%TYPE)
AS
BEGIN
DELETE from PATIENT WHERE PATIENT_ID =pid ;
END;
```

```
CREATE OR REPLACE PROCEDURE delete_patient (pid IN PATIENT.PATIENT_ID%TYPE)
AS
BEGIN
DELETE from PATIENT WHERE PATIENT_ID =pid ;
END;
```

Results Explain Describe Saved SQL History

Procedure created.

6) Delete for Nurse

```
CREATE OR REPLACE PROCEDURE delete_NURSE (nid IN NURSE.NURSE_ID%TYPE)
AS
BEGIN
DELETE from NURSE WHERE NURSE_ID =nid ;
END;
```

```
CREATE OR REPLACE PROCEDURE delete_NURSE (nid IN NURSE.NURSE_ID%TYPE)
AS
BEGIN
DELETE from NURSE WHERE NURSE_ID =nid ;
END;
```

Results	Explain	Describe	Saved SQL	History
----------------	---------	----------	-----------	---------

Procedure created.

Question No 08:

Minimum 4 triggers which must be implemented in the system.

Triggers:

1) Every NURSE can be only register 8am to 5 pm

```
Create or replace trigger Restricted_Insert_NURSE
Before Insert on NURSE
Begin
if(TO_CHAR(SYSTIMESTAMP,'HH24:MI') Not between '05:00' AND '17:00')
then
raise_application_error(-20123,'You can add NURSE between 5:00 AM TO
5:00 PM ONLY');
End if;
End;
Alter trigger Restricted_Insert_NURSE enable;
```

```
Create or replace trigger Restricted_Insert_NURSE
Before Insert on NURSE
Begin
if(TO_CHAR(SYSTIMESTAMP,'HH24:MI') Not between '05:00' AND '17:00') then
raise_application_error(-20123,'You can add NURSE between 5:00 AM TO 5:00 PM ONLY');
End if;
End;
Alter trigger Restricted_Insert_NURSE enable;
```

Results Explain Describe Saved SQL History

Trigger created.

2) Every DOCTOR can be only register 8am to 5 pm

Create or replace trigger Restricted_Insert

Before Insert on DOCTOR

Begin

if(TO_CHAR(SYSTIMESTAMP,'HH24:MI') Not between '08:00' AND '17:00') then
raise_application_error(-20124,'You can add DOCTOR between 8:00 AM TO 5:00
PM ONLY');

End if;

End;

alter trigger Restricted_Insert enable;

```
Create or replace trigger Restricted_Insert
Before Insert on DOCTOR
Begin
if(TO_CHAR(SYSTIMESTAMP,'HH24:MI') Not between '08:00' AND '17:00') then
raise_application_error(-20124,'You can add DOCTOR between 8:00 AM TO 5:00 PM ONLY');
End if;
End;
alter trigger Restricted_Insert enable;
```

Results Explain Describe Saved SQL History

Trigger created.

3) Every RECEPTIONIST can be only register 8am to 5 pm

Create or replace trigger Restricted_Insert_RECEPTIONIST

Before Insert on RECEPTIONIST

Begin

if(TO_CHAR(SYSTIMESTAMP,'HH24:MI') Not between '08:00' AND '17:00') then

raise_application_error(-20125,'You can add RECEPTIONIST between 8:00 AM TO 5:00 PM ONLY');

End if;

End;

Alter trigger Restricted_Insert_RECEPTIONIST enable;

```
Create or replace trigger Restricted_Insert_RECEPTIONIST
Before Insert on RECEPTIONIST
Begin
if(TO_CHAR(SYSTIMESTAMP,'HH24:MI') Not between '08:00' AND '17:00') then
raise_application_error(-20125,'You can add RECEPTIONIST between 8:00 AM TO 5:00 PM ONLY');
End if;
End;
Alter trigger Restricted_Insert_RECEPTIONIST enable;
```

Results Explain Describe Saved SQL History

Trigger created.

4) Nurse cannot be empty

Create or replace TRIGGER NURSE_Delete

Before delete on NURSE

Declare NURSE_count number (3);

Begin

Select count (NURSE_id) into NURSE_count from NURSE;

if(NURSE_count<1) then

raise_application_error(-20129,'Minimum one NURSE MUST be there');

End if;

End;

```
Create or replace TRIGGER NURSE_Delete
Before delete on NURSE
Declare NURSE_count number (3);
Begin
Select count (NURSE_id) into NURSE_count from NURSE;
if(NURSE_count<1) then
raise_application_error(-20129,'Minimum one NURSE MUST be there');
End if;
End;
```

Results Explain Describe Saved SQL History

Trigger created.

Question No 09:

You may apply package, locking and exception handling which is optional and will carry bonus marks.

PACKAGE

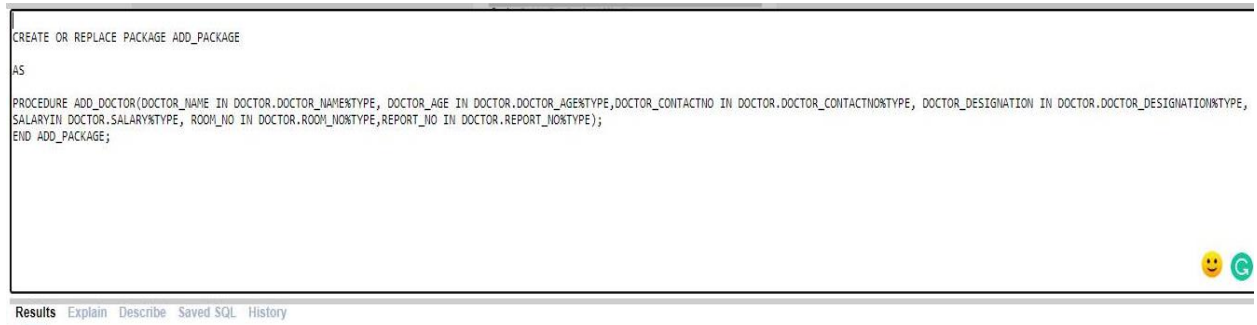
1)

CREATE OR REPLACE PACKAGE ADD_PACKAGE

AS

```
PROCEDURE ADD_DOCTOR(DOCTOR_NAME IN DOCTOR.DOCTOR_NAME%TYPE,
DOCTOR_AGE IN DOCTOR.DOCTOR_AGE%TYPE,DOCTOR_CONTACTNO IN
DOCTOR.DOCTOR_CONTACTNO%TYPE, DOCTOR_DESIGNATION IN
DOCTOR.DOCTOR_DESIGNATION%TYPE, SALARYIN DOCTOR.SALARY%TYPE,
ROOM_NO IN DOCTOR.ROOM_NO%TYPE,REPORT_NO IN
DOCTOR.REPORT_NO%TYPE);
END ADD_PACKAGE;
```

```
CREATE OR REPLACE PACKAGE ADD_PACKAGE
AS
PROCEDURE ADD_DOCTOR(DOCTOR_NAME IN DOCTOR.DOCTOR_NAME%TYPE, DOCTOR_AGE IN DOCTOR.DOCTOR_AGE%TYPE,DOCTOR_CONTACTNO IN DOCTOR.DOCTOR_CONTACTNON%TYPE, DOCTOR_DESIGNATION IN DOCTOR.DOCTOR_DESIGNATION%TYPE,
SALARYIN DOCTOR.SALARY%TYPE, ROOM_NO IN DOCTOR.ROOM_NON%TYPE,REPORT_NO IN DOCTOR.REPORT_NON%TYPE);
END ADD_PACKAGE;
```



Package created.

2)

CREATE OR REPLACE PACKAGE BODY ADD_PACKAGE

AS

PROCEDURE ADD_DOCTOR(DOCTOR_NAME IN DOCTOR.DOCTOR_NAME%TYPE,
DOCTOR_AGE IN DOCTOR.DOCTOR_AGE%TYPE,DOCTOR_CONTACTNO IN
DOCTOR.DOCTOR_CONTACTNO%TYPE, DOCTOR_DESIGNATION IN
DOCTOR.DOCTOR_DESIGNATION%TYPE, SALARY IN DOCTOR.SALARY%TYPE,
ROOM_NO IN DOCTOR.ROOM_NO%TYPE,REPORT_NO IN
DOCTOR.REPORT_NO%TYPE)

AS

BEGIN

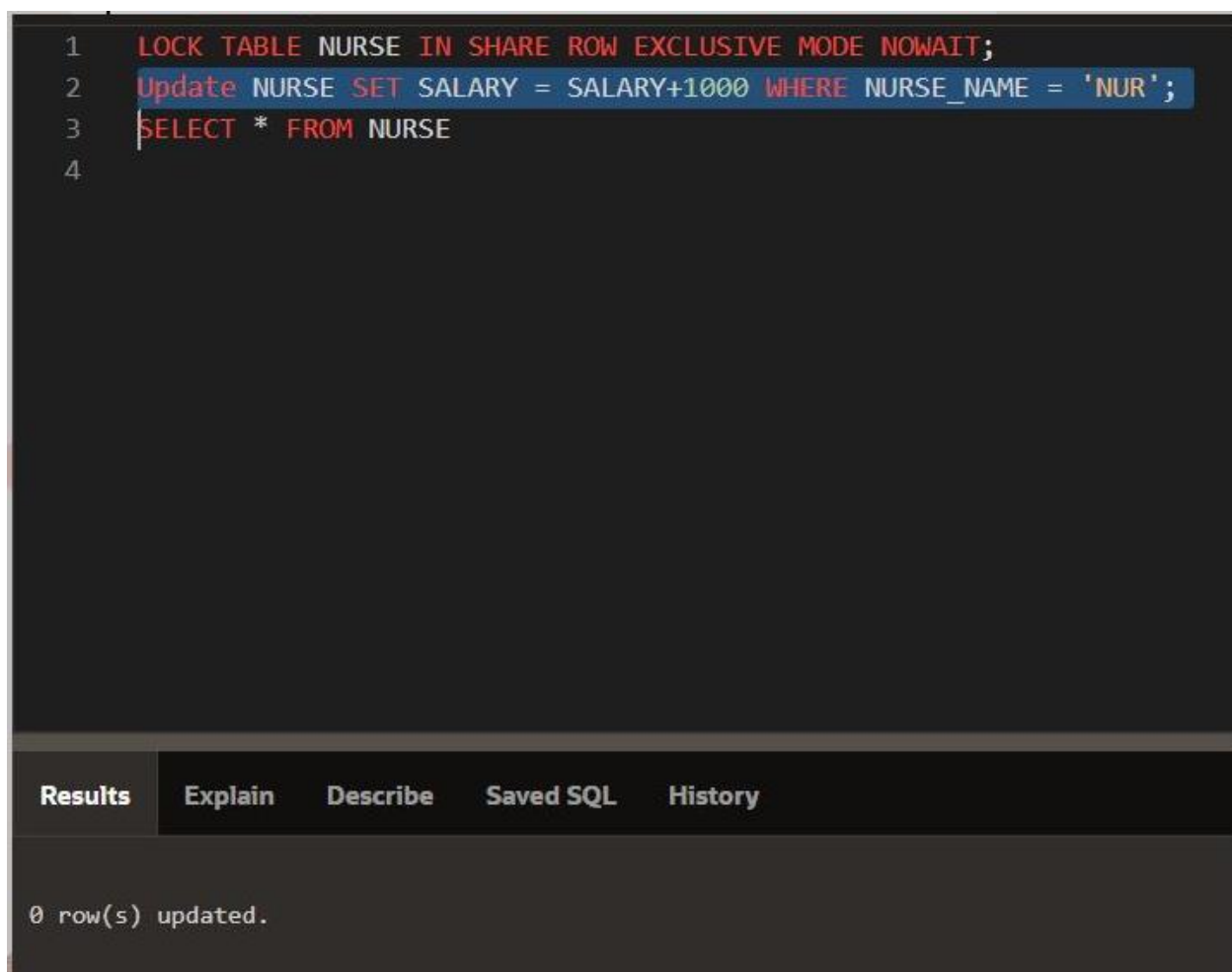
INSERT INTO DOCTOR VALUES (DOCTOR_ID.NEXTVAL,DOCTOR_NAME,
DOCTOR_AGE, DOCTOR_CONTACTNO,DOCTOR_DESIGNATION, SALARY,
DocDesignation);

END ADD_DOCTOR;

LOCKING:

1)

```
LOCK TABLE NURSE IN SHARE ROW EXCLUSIVE MODE NOWAIT;  
Update NURSE SET SALARY = SALARY+1000 WHERE NURSE_NAME = 'NUR';  
SELECT * FROM NURSE;
```



The screenshot shows a SQL IDE interface with a dark background. The SQL editor contains three lines of code: a lock statement, an update statement, and a select statement. The update statement is highlighted with a blue background. Below the editor is a toolbar with buttons for 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The 'Results' button is selected, and the output area shows '0 row(s) updated.'

```
1 LOCK TABLE NURSE IN SHARE ROW EXCLUSIVE MODE NOWAIT;  
2 Update NURSE SET SALARY = SALARY+1000 WHERE NURSE_NAME = 'NUR';  
3 SELECT * FROM NURSE  
4
```

Results Explain Describe Saved SQL History

0 row(s) updated.

Exception handling:

1)

```
DECLARE
n_name NURSE.NURSE_NAME%TYPE;
n_NURSE_ID NURSE.NURSE_ID%TYPE := 556;
BEGIN
SELECT NURSE_NAME into n_name
FROM NURSE
WHERE NURSE_ID = n_NURSE_ID;
dbms_output.put_line( 'NURSE Name is ' || n_name );
EXCEPTION
WHEN NO_DATA_FOUND then
dbms_output.put_line('NurseID' || ':' || n_NURSE_ID || ' ' || 'does not exits');
END;
```

```
DECLARE
n_name NURSE.NURSE_NAME%TYPE;
n_NURSE_ID NURSE.NURSE_ID%TYPE := 556;
BEGIN
SELECT NURSE_NAME into n_name
FROM NURSE
WHERE NURSE_ID = n_NURSE_ID;
dbms_output.put_line( 'NURSE Name is ' || n_name );
EXCEPTION
WHEN NO_DATA_FOUND then
dbms_output.put_line('NurseID' || ':' || n_NURSE_ID || ' ' || 'does not exits');
END;
```

Results Explain Describe Saved SQL History

NurseID:556 does not exits

Statement processed.

```

2) DECLARE
    d_id DOCTOR.DOCTOR_ID%type := 0;
    d_name DOCTOR.DOCTOR_Name%type;
    ex_invalid_id EXCEPTION;
BEGIN
    IF d_id <= 0 THEN
        RAISE ex_invalid_id;
    ELSE
        SELECT DOCTOR_NAME INTO d_name
        FROM DOCTOR
        WHERE DOCTOR_ID = d_id;
        DBMS_OUTPUT.PUT_LINE ('Name: ' || d_name);
    END IF;
EXCEPTION
    WHEN ex_invalid_id THEN
        dbms_output.put_line('ID must be greater than zero!');
    WHEN no_data_found THEN
        dbms_output.put_line('No such customer!');
    WHEN others THEN
        dbms_output.put_line('Error!');
END;

```

```

DECLARE
    d_id DOCTOR.DOCTOR_ID%type := 0;
    d_name DOCTOR.DOCTOR_Name%type;
    ex_invalid_id EXCEPTION;
BEGIN
    IF d_id <= 0 THEN
        RAISE ex_invalid_id;
    ELSE
        SELECT DOCTOR_NAME INTO d_name
        FROM DOCTOR
        WHERE DOCTOR_ID = d_id;
        DBMS_OUTPUT.PUT_LINE ('Name: ' || d_name);
    END IF;
EXCEPTION
    WHEN ex_invalid_id THEN
        dbms_output.put_line('ID must be greater than zero!');
    WHEN no_data_found THEN
        dbms_output.put_line('No such customer!');
    WHEN others THEN
        dbms_output.put_line('Error!');
END;

```

Results Explain Describe Saved SQL History

ID must be greater than zero!

Statement processed.