Department Information Systems and Technologies

CTIS259 Database Management Systems and Applications

2025 – 2026 Fall

Lab Guide 20

Instructor	: Nimet Ceren SERİM	**Week:**	13
Assistant	: Hatice Zehra YILMAZ, Engin Zafer KIRAÇBEDEL	**Date:**	08-09-12.2025

Aim of this lab session: **1.** PL/SQL Practice on Health Care Management System Database

ORACLE Server Configurations:

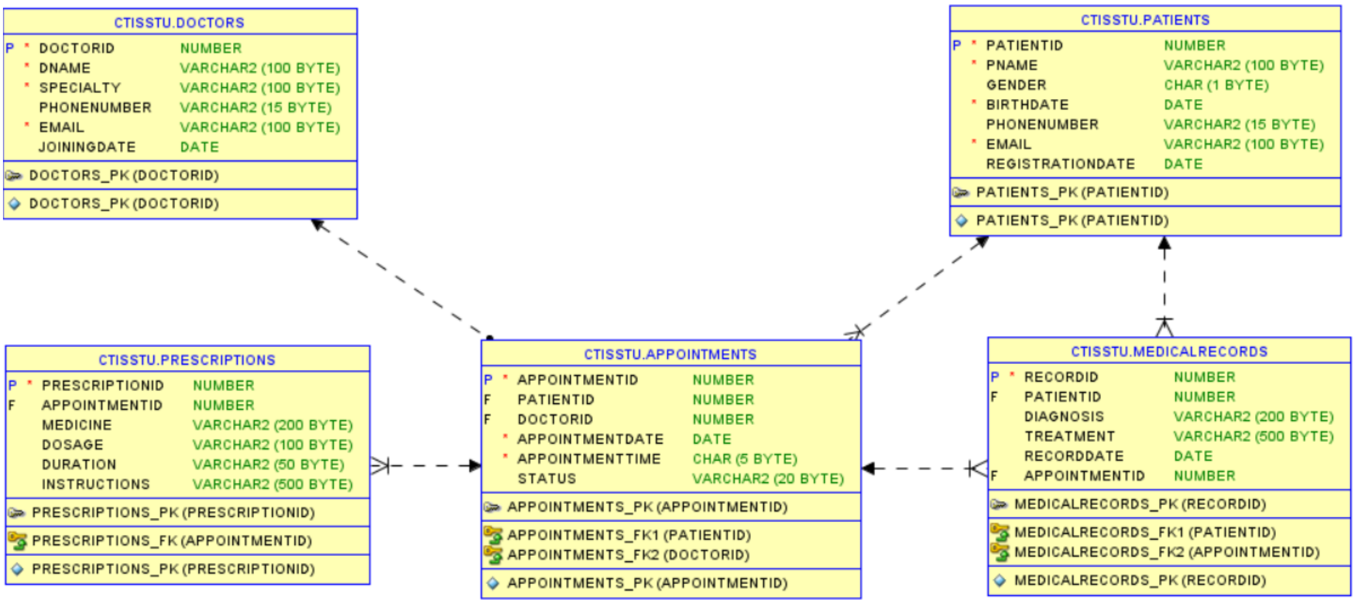
IP Address: 139.179.33.231

Port number: 1522

SID: orclctis

Please USE stu (fYourStudentID) accounts

The Healthcare Management System is designed to manage and organize information about patients, doctors, appointments, prescriptions, and medical records. It ensures that healthcare providers can maintain accurate records, schedule appointments, and deliver quality medical services.


CTISSTU.DOCTORS
P * DOCTORID NUMBER
* DNAME VARCHAR2 (100 BYTE)
* SPECIALTY VARCHAR2 (100 BYTE)
PHONENUMBER VARCHAR2 (15 BYTE)
* EMAIL VARCHAR2 (100 BYTE)
JOININGDATE DATE
DOCTORS_PK (DOCTORID)
DOCTORS_PK (DOCTORID)
CTISSTU.PATIENTS
P * PATIENTID NUMBER
* PNAME VARCHAR2 (100 BYTE)
GENDER CHAR (1 BYTE)
* BIRTHDATE DATE
PHONENUMBER VARCHAR2 (15 BYTE)
* EMAIL VARCHAR2 (100 BYTE)
REGISTRATIONDATE DATE
PATIENTS_PK (PATIENTID)
PATIENTS_PK (PATIENTID)
CTISSTU.PRESCRIPTIONS
P * PRESCRIPTIONID NUMBER
F APPOINTMENTID NUMBER
MEDICINE VARCHAR2 (200 BYTE)
DOSAGE VARCHAR2 (100 BYTE)
DURATION VARCHAR2 (50 BYTE)
INSTRUCTIONS VARCHAR2 (500 BYTE)
PRESCRIPTIONS_PK (PRESCRIPTIONID)
PRESCRIPTIONS_FK (APPOINTMENTID)
PRESCRIPTIONS_PK (PRESCRIPTIONID)
CTISSTU.APPOINTMENTS
P * APPOINTMENTID NUMBER
F PATIENTID NUMBER
F DOCTORID NUMBER
* APPOINTMENTDATE DATE
* APPOINTMENTTIME CHAR (5 BYTE)
STATUS VARCHAR2 (20 BYTE)
APPOINTMENTS_PK (APPOINTMENTID)
APPOINTMENTS_FK1 (PATIENTID)
APPOINTMENTS_FK2 (DOCTORID)
APPOINTMENTS_PK (APPOINTMENTID)
CTISSTU.MEDICALRECORDS
P * RECORDID NUMBER
F PATIENTID NUMBER
DIAGNOSIS VARCHAR2 (200 BYTE)
TREATMENT VARCHAR2 (500 BYTE)
RECORDDATE DATE
F APPOINTMENTID NUMBER
MEDICALRECORDS_PK (RECORDID)
MEDICALRECORDS_FK1 (PATIENTID)
MEDICALRECORDS_FK2 (APPOINTMENTID)
MEDICALRECORDS_PK (RECORDID)

Creating tables and data insertions

Please download **HealthDB.zip** file from moodle: **Labguides/LG20.** Zip file contains **database creation and data insertion scripts for Health Care Management database.** Create the below tables in your account and insert data into the tables using **the given scripts** file.

Table PATIENTS:

Field Name	**Data Type**	**Size**	**Constraints**
PatientID	NUMBER		PRIMARY KEY
pName	VARCHAR2	100	NOT NULL
Gender	CHAR	1	'M' or 'F'
BirthDate	DATE		NOT NULL
PhoneNumber	VARCHAR2	15	
Email	VARCHAR2	100	NOT NULL
RegistrationDate	DATE		DEFAULT SYSDATE

Table DOCTORS:

Field Name	**Data Type**	**Size**	**Constraints**
DoctorID	NUMBER		PRIMARY KEY
DName	VARCHAR2	100	NOT NULL
Specialty	VARCHAR2	100	NOT NULL
PhoneNumber	VARCHAR2	15	
Email	VARCHAR2	100	NOT NULL
JoiningDate	DATE		DEFAULT SYSDATE

Table APPOINTMENTS:

Field Name	**Data Type**	**Size**	**Constraints**
AppointmentID	NUMBER		PRIMARY KEY
PatientID	NUMBER		FOREIGN KEY
DoctorID	NUMBER		FOREIGN KEY
AppointmentDate	DATE		NOT NULL
AppointmentTime	CHAR	5	NOT NULL
Status	VARCHAR2	20	DEFAULT 'Scheduled', CHECK: 'Scheduled'/'Completed'/'Cancelled'

Table PRESCRIPTIONS:

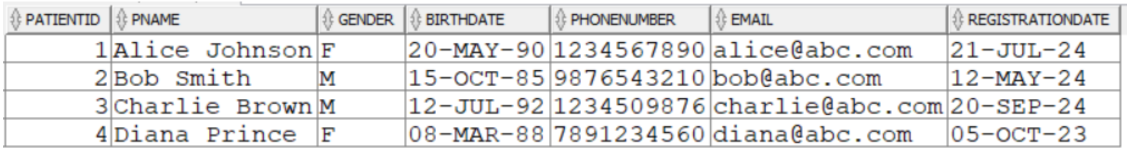
Field Name	**Data Type**	**Size**	**Constraints**
PrescriptionID	NUMBER		PRIMARY KEY
AppointmentID	NUMBER		FOREIGN KEY
Medicine	VARCHAR2	200	
Dosage	VARCHAR2	100	
Duration	VARCHAR2	50	
Instructions	VARCHAR2	500	

Table MEDICALREPORTS:

Field Name	**Data Type**	**Size**	**Constraints**
RecordID	NUMBER		PRIMARY KEY
PatientID	NUMBER		FOREIGN KEY
Diagnosis	VARCHAR2	200	
Treatment	VARCHAR2	500	
RecordDate	DATE		DEFAULT SYSDATE
AppointmentID	NUMBER		FOREIGN KEY

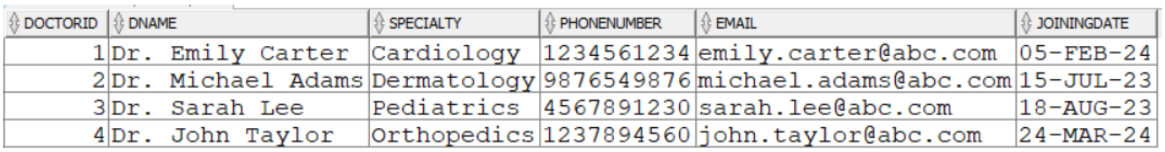
A. Data Insertion

- The records as given below for the table of **PATIENTS**.



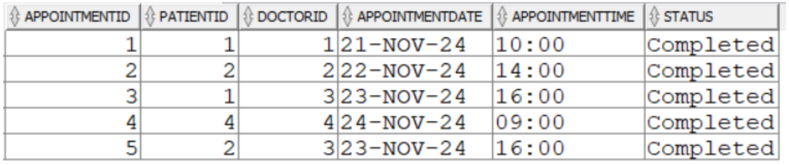
PATIENTID	PNAME	GENDER	BIRTHDATE	PHONENUMBER	EMAIL	REGISTRATIONDATE
1	Alice Johnson	F	20-MAY-90	1234567890	alice@abc.com	21-JUL-24
2	Bob Smith	M	15-OCT-85	9876543210	bob@abc.com	12-MAY-24
3	Charlie Brown	M	12-JUL-92	1234509876	charlie@abc.com	20-SEP-24
4	Diana Prince	F	08-MAR-88	7891234560	diana@abc.com	05-OCT-23

- The records as given below for the table of **DOCTORS**.



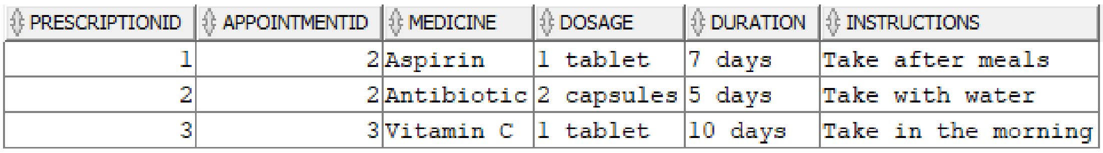
DOCTORID	DNAME	SPECIALTY	PHONENUMBER	EMAIL	JOININGDATE
1	Dr. Emily Carter	Cardiology	1234561234	emily.carter@abc.com	05-FEB-24
2	Dr. Michael Adams	Dermatology	9876549876	michael.adams@abc.com	15-JUL-23
3	Dr. Sarah Lee	Pediatrics	4567891230	sarah.lee@abc.com	18-AUG-23
4	Dr. John Taylor	Orthopedics	1237894560	john.taylor@abc.com	24-MAR-24

- The records as given below for the table of **APPOINTMENTS**.



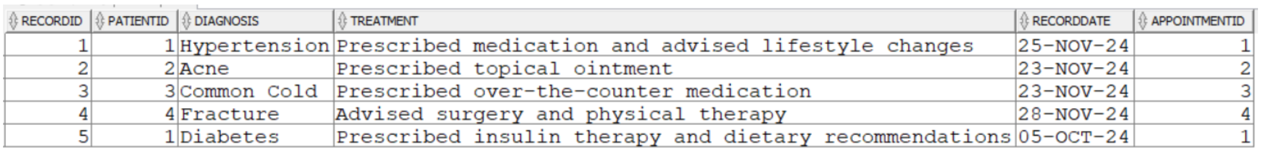
APPOINTMENTID	PATIENTID	DOCTORID	APPOINTMENTDATE	APPOINTMENTTIME	STATUS
1	1	1	21-NOV-24	10:00	Completed
2	2	2	22-NOV-24	14:00	Completed
3	1	3	23-NOV-24	16:00	Completed
4	4	4	24-NOV-24	09:00	Completed
5	2	3	23-NOV-24	16:00	Completed

- The records as given below for the table of **PRESCRIPTIONS**.



PRESCRIPTIONID	APPOINTMENTID	MEDICINE	DOSAGE	DURATION	INSTRUCTIONS
1	2	Aspirin	1 tablet	7 days	Take after meals
2	2	Antibiotic	2 capsules	5 days	Take with water
3	3	Vitamin C	1 tablet	10 days	Take in the morning

- The records as given below for the table of **MEDICALRECORDS**.



RECORDID	PATIENTID	DIAGNOSIS	TREATMENT	RECORDDATE	APPOINTMENTID
1	1	Hypertension	Prescribed medication and advised lifestyle changes	25-NOV-24	1
2	2	Acne	Prescribed topical ointment	23-NOV-24	2
3	3	Common Cold	Prescribed over-the-counter medication	23-NOV-24	3
4	4	Fracture	Advised surgery and physical therapy	28-NOV-24	4
5	1	Diabetes	Prescribed insulin therapy and dietary recommendations	05-OCT-24	1

- **Do NOT forget to COMMIT changes!!!**

Creating Procedures & Functions

Create the specified procedures or functions and write the anonymous PL/SQL blocks to execute them.

Q1. Appointment Scheduling Procedure

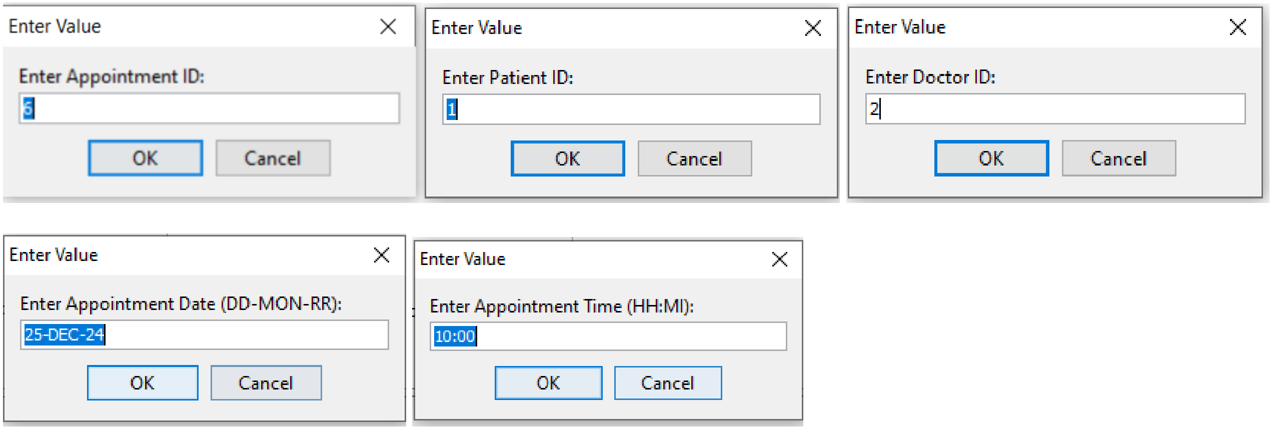
Write a procedure that takes the *appointment Id, Patient ID, Doctor ID, Appointment Date* and *Appointment time* as input parameters, then checks if the appointment slot is available before scheduling. If the slot is unavailable, a warning message will be displayed otherwise new appointment will be inserted into the appointments table.

Optionally, verify if the ***appointment ID*** *already exists and confirm that both the* ***patient*** *and* ***doctor*** *are valid*. (You may use sql%found attribute or NO_DATA_FOUND exception)

Write an anonymous block that displays appropriate prompts for the substitution variables, reads the information of the appointment from the user and test your procedure.

Sample Input Data:

```
Appointment ID: 6
Patient ID: 1
Doctor ID: 2
Appointment Date: 25-DEC-24
Appointment Time: 10:00
```



Enter Value

Enter Appointment ID: 6

Enter Value

Enter Patient ID: 1

Enter Value

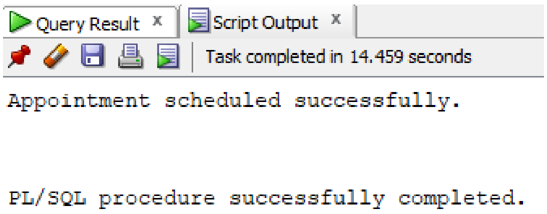
Enter Doctor ID: 2

Enter Value

Enter Appointment Date (DD-MON-RR): 25-DEC-24

Enter Value

Enter Appointment Time (HH:MI): 10:00

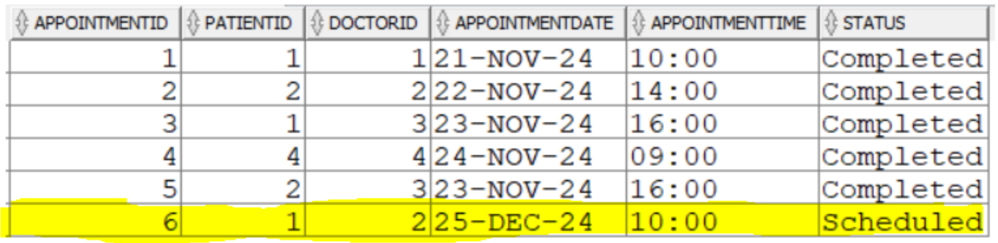


```
Task completed in 14.459 seconds

Appointment scheduled successfully.


PL/SQL procedure successfully completed.
```

Updated Version of the Table Appointments



APPOINTMENTID	PATIENTID	DOCTORID	APPOINTMENTDATE	APPOINTMENTTIME	STATUS
1	1	1	21-NOV-24	10:00	Completed
2	2	2	22-NOV-24	14:00	Completed
3	1	3	23-NOV-24	16:00	Completed
4	4	4	24-NOV-24	09:00	Completed
5	2	3	23-NOV-24	16:00	Completed
6	1	2	25-DEC-24	10:00	Scheduled

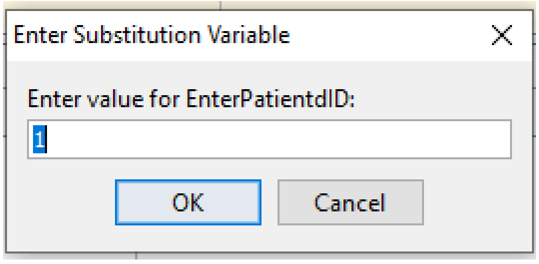
Q2. Patient Medical History Report Procedure

Write a procedure that takes a patient Id as input parameter and displays all medical records for the patient. If there is no medical report for the patient, an appropriate message will be displayed.

Write an anonymous block to read the patient id from the user and test your procedure.

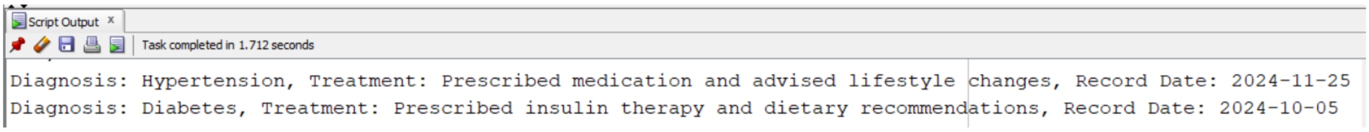
“An explicit cursor does not raise the NO_DATA_FOUND exception. The NO_DATA_FOUND exception is raised only in specific contexts, such as when a SELECT INTO statement returns no rows, but not during a fetch operation from an explicit cursor”

Example Run:



Enter Substitution Variable

Enter value for EnterPatientdID: 1

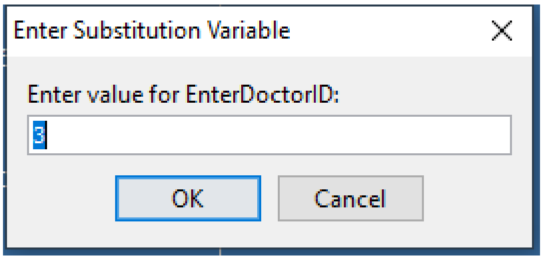


```
Diagnosis: Hypertension, Treatment: Prescribed medication and advised lifestyle changes, Record Date: 2024-11-25
Diagnosis: Diabetes, Treatment: Prescribed insulin therapy and dietary recommendations, Record Date: 2024-10-05
```

Q3. Doctor Schedule

Write a function that takes the doctor id as input parameter, retrieves all appointments for the given doctor. The function counts and returns the number of appointments of the doctor.

Write anonymous block, read the doctor id from the user and test the function.
If there is no appointment, then display a warning.



Enter Substitution Variable

Enter value for EnterDoctorID: 3

Script Output

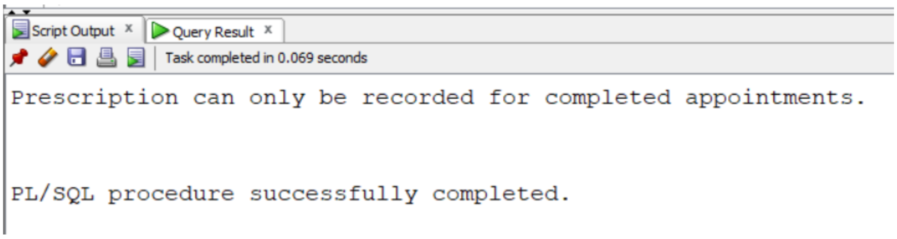
```
Appointment Date: 2024-11-23, Patient Name: Charlie Brown, Status: Scheduled
Appointment Date: 2024-11-23, Patient Name: Bob Smith, Status: Scheduled
Total Appointments: 2


PL/SQL procedure successfully completed.
```

Q4. Prescription Recording Procedure

Write a procedure that validates if an appointment is marked as **completed** before allowing a prescription to be recorded. **Test your procedure with the following anonymous block;**

```
SET SERVEROUTPUT ON;
DECLARE
    PrescriptionID NUMBER := 4;
    AppointmentID NUMBER := 6;
    Medicine VARCHAR2(100) := 'Paracetamol';
    Dosage VARCHAR2(50) := '500mg';
    Duration VARCHAR2(50) := '5 days';
    Instructions VARCHAR2(200) := 'Take 1 tablet every 6 hours';
BEGIN
    -- Call the RecordPrescription procedure with sample data
    RecordPrescription (PrescriptionID, AppointmentID, Medicine, Dosage, Duration, Instructions);
END;
/
```



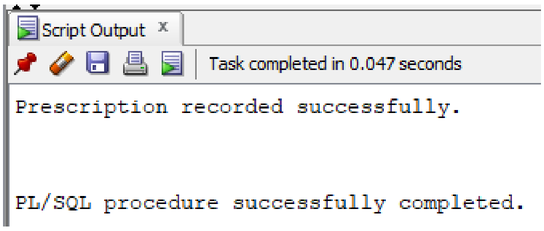
```
Prescription can only be recorded for completed appointments.


PL/SQL procedure successfully completed.
```

Now insert a new record and reexecute the anonymous block but this time Appointment ID will be 7;

```
INSERT INTO Appointments
(AppointmentID, PatientID, DoctorID, AppointmentDate, AppointmentTime, Status)
VALUES (7, 1, 1, TO_DATE('2024-12-14', 'YYYY-MM-DD'), '10:00', 'Completed');
```

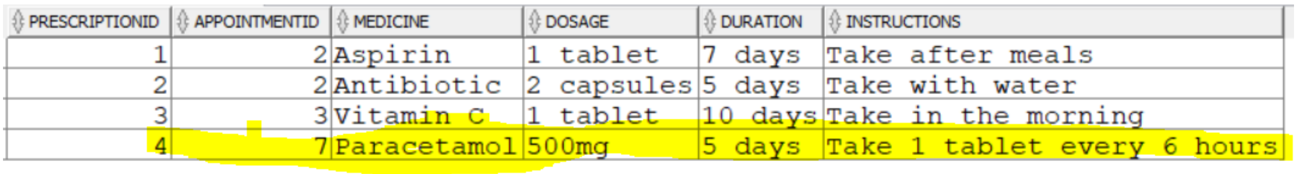
```
SET SERVEROUTPUT ON;
DECLARE
    PrescriptionID NUMBER := 4;
    AppointmentID NUMBER := 7;
    Medicine VARCHAR2(100) := 'Paracetamol';
    Dosage VARCHAR2(50) := '500mg';
    Duration VARCHAR2(50) := '5 days';
    Instructions VARCHAR2(200) := 'Take 1 tablet every 6 hours';
BEGIN
    -- Call the RecordPrescription procedure with sample data
    RecordPrescription (PrescriptionID, AppointmentID, Medicine, Dosage, Duration, Instructions);
END;
/
```



```
Prescription recorded successfully.


PL/SQL procedure successfully completed.
```

Updated version of the precriptions table



PRESCRIPTIONID	APPOINTMENTID	MEDICINE	DOSAGE	DURATION	INSTRUCTIONS
1	2	Aspirin	1 tablet	7 days	Take after meals
2	2	Antibiotic	2 capsules	5 days	Take with water
3	3	Vitamin C	1 tablet	10 days	Take in the morning
4	7	Paracetamol	500mg	5 days	Take 1 tablet every 6 hours