



## AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH

PROJECT TITLE: DOCTOR APPOINTMENT MANAGEMENT SYSTEM

GROUP MEMBER:

NAME	ID	CONTRIBUTION	TOP NAME
MD. EKRAMUL HASAN NIBIR	20-42740-1	20%	SCENARIO DESCRIPTION, TABLE CREATION, QUERY WRITING, ER DIAGRAM
MD. ABDUR RAHMAN FOYSAL	20-42742-1	20%	DATA INSERTION, RELATIONAL ALGEBRA, ACTIVITY DIAGRAM, PROJECT RPOPOSAL
MD.YOUSUF ALI SHADHIN	20-42783-1	20%	NORMALIZAION, SCHEMA DIAGRAM, UI DESIGN, USE CASE DIAGRAM
TANJEMUL HOQUE TUHIN	19-41020-2	20%	
PURAVI DEBNATH NITU	19-41371-3	20%	

COURSE TITLE: ADVANCE DATABASE MANAGEMENT SYSTEM

SECTION: B

## **Contents**

1	Introduction .....	4
2	Project Proposal.....	4
3	Diagram.....	5
3.1	Class Diagram .....	5
3.2	Use Case Diagram .....	6
3.3	Activity Diagram .....	7
4	Use Interface .....	8
5	Scenario Description .....	10
6	ER Diagram .....	11
7	Normalization .....	11
8	Schema Diagram .....	16
9	Table Creation .....	17
9.1	Screenshots of the created table .....	17
9.2	Queries.....	22
10	Data Insertion .....	30
10.1	Screenshots of data insertion .....	30
10.2	Queries For data insertion .....	35
10.3	Screenshots of .....	39
10.3.1	Create Users .....	39
10.3.2	Assign Roles .....	39
10.3.3	Grant Privileges .....	39
11	Query Writing .....	40
11.1	Single-row-function .....	41

11.2 Group Function .....	41
11.3 Subquery .....	43
11.4 Joining .....	44
11.5 View .....	46
11.6 Synonym .....	47
12 Relational Algebra .....	49
12.1 Project Operation.....	49
12.2 Selection Operation.....	49
12.3 Cartesian Product Operation.....	49
12.4 Union Operation .....	49
12.5 Rename Operation.....	49
13 Conclusion .....	50

### **Introduction:**

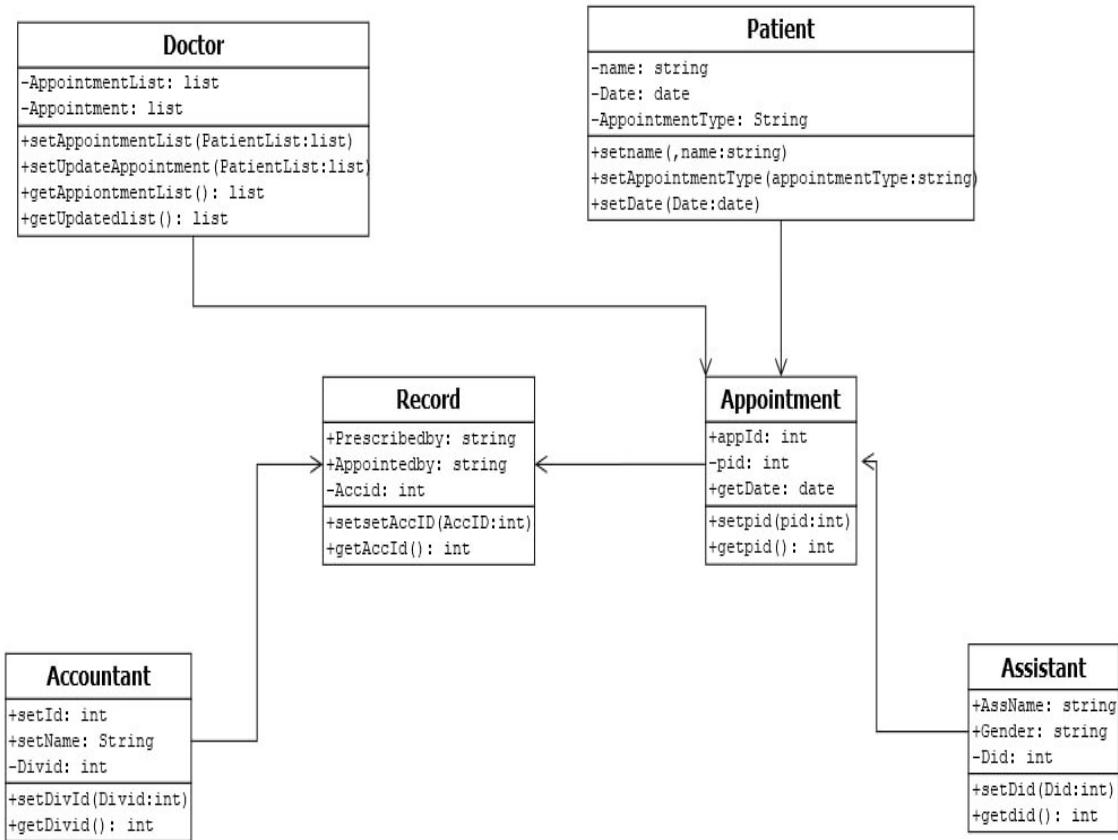
The management of healthcare systems requires efficient and accurate management of patient appointments, medical records, and healthcare providers. The database management system discussed in this case study focuses on the efficient handling of information related to a "Doctor Appointment Management System". The system allows patients to book appointments with doctors, while an assistant deals with managing the appointments. The system also keeps track of patient information, such as their blood group, phone numbers, and gender, and doctor information, such as their hire date, salary, and degrees. Additionally, the system maintains medical records, identifies patients prescribed by doctors, and manages payment to healthcare providers. The system's comprehensive approach ensures that all aspects of patient care and management are seamlessly integrated, resulting in a more efficient and effective healthcare system.

### **Project proposal:**

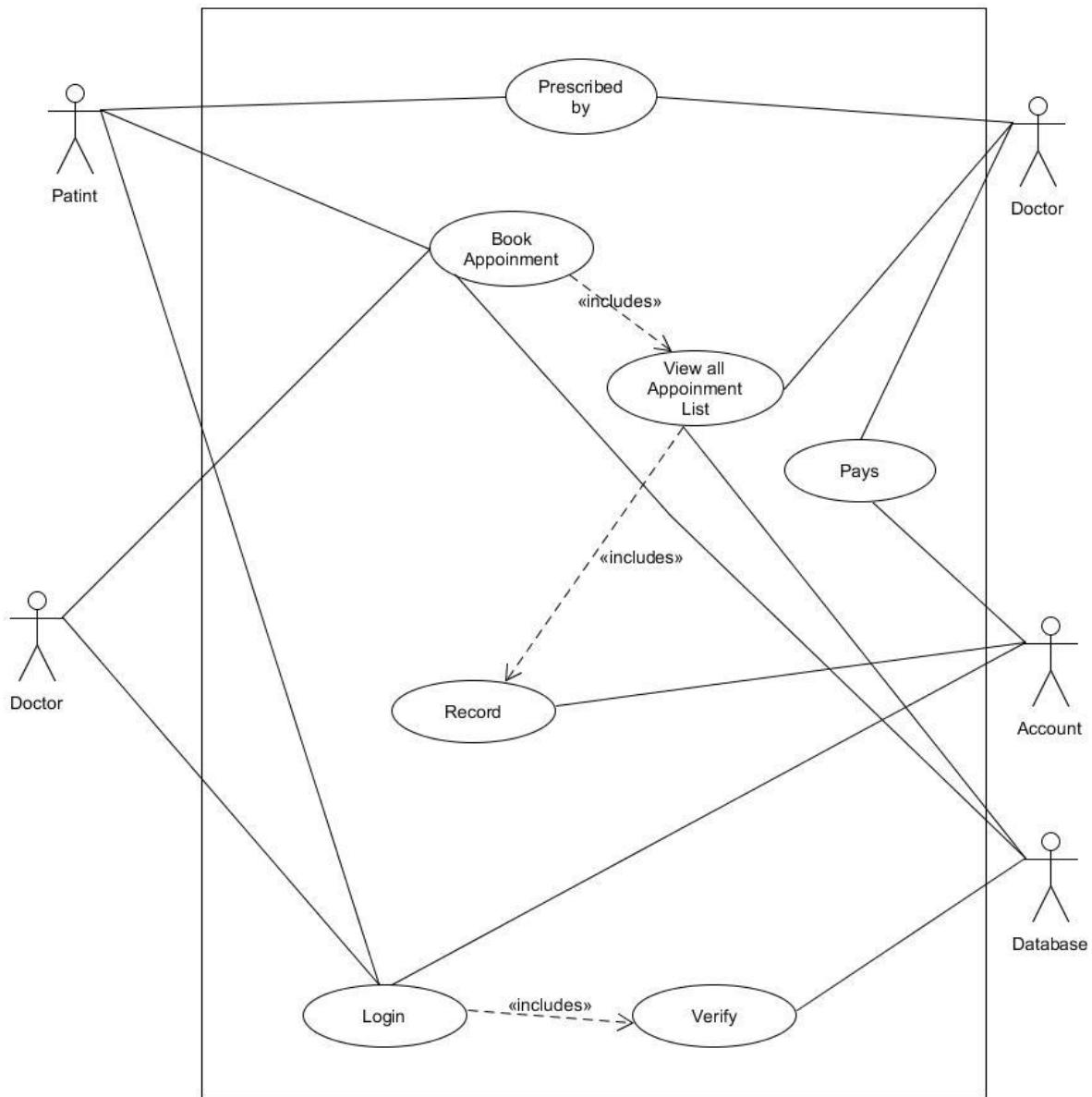
The proposed Doctor Appointment Management System aims to provide an efficient platform for patients to book appointments, doctors to manage appointments, and accountants to maintain financial records. The system includes modules for patient registration, appointment management, prescription and medical records, accountant management, and prioritizes security measures. The system will use MySQL, HTML, CSS and is scalable for future expansion.

**Diagram:**

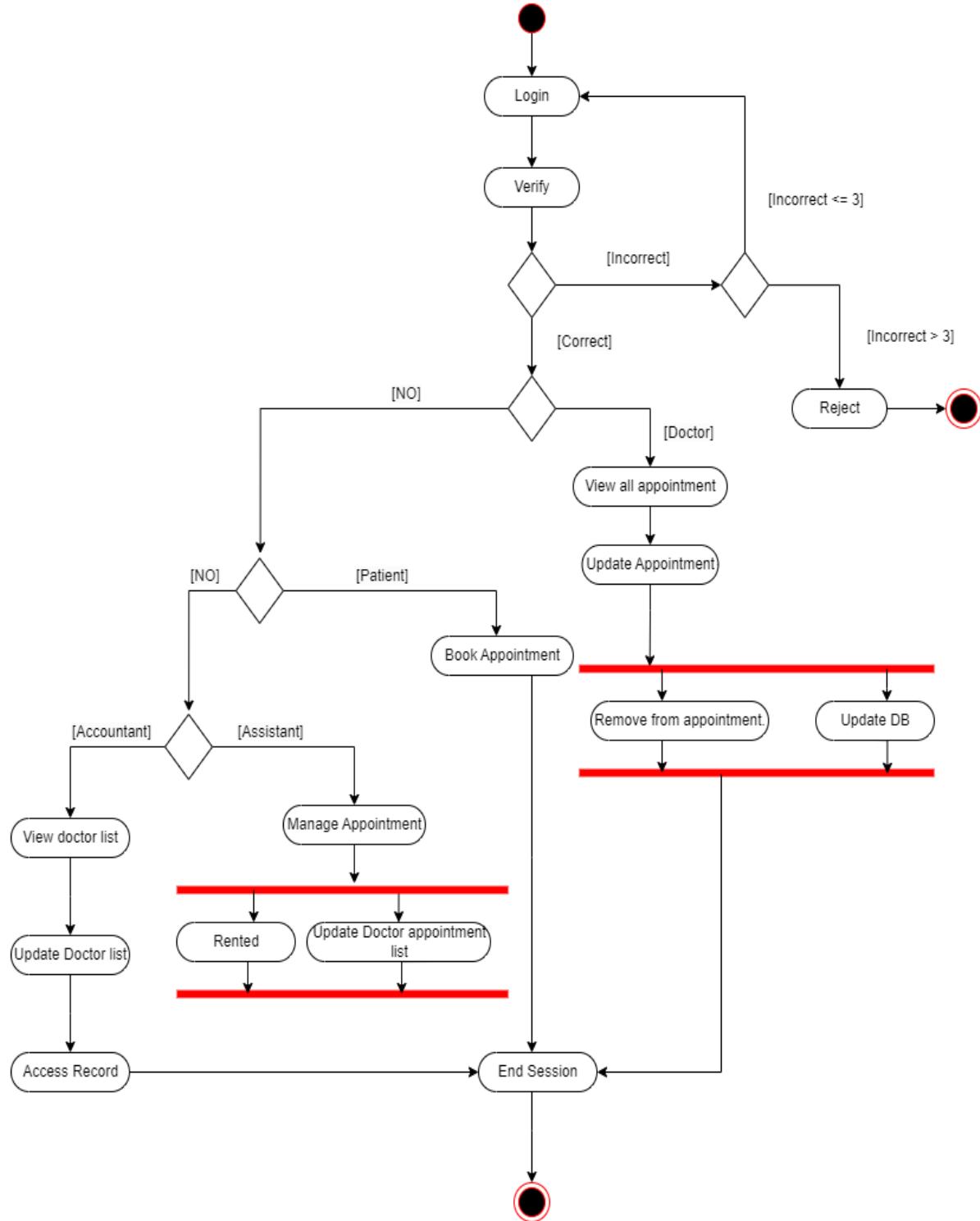
**Class Diagram:**



### User Case Diagram:



### Activity Diagram:



## User Interface:

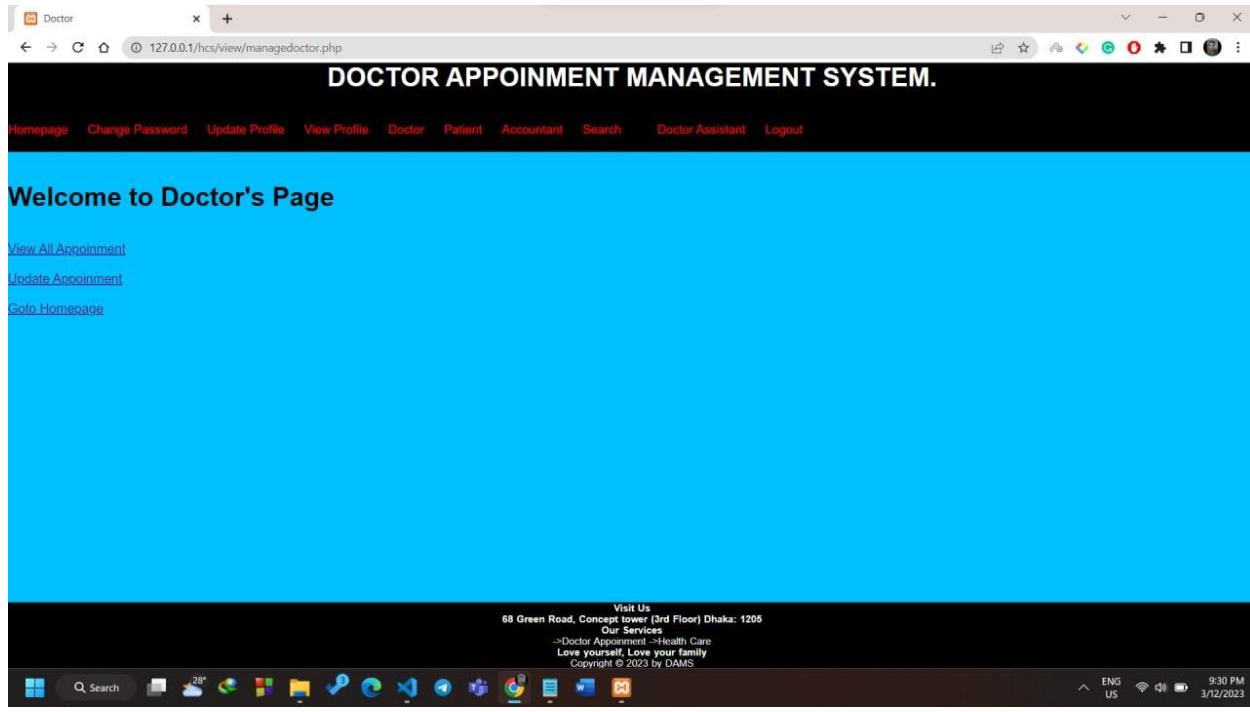
The image displays two screenshots of a web-based Doctor Appointment Management System (DAMS) running on a Windows operating system.

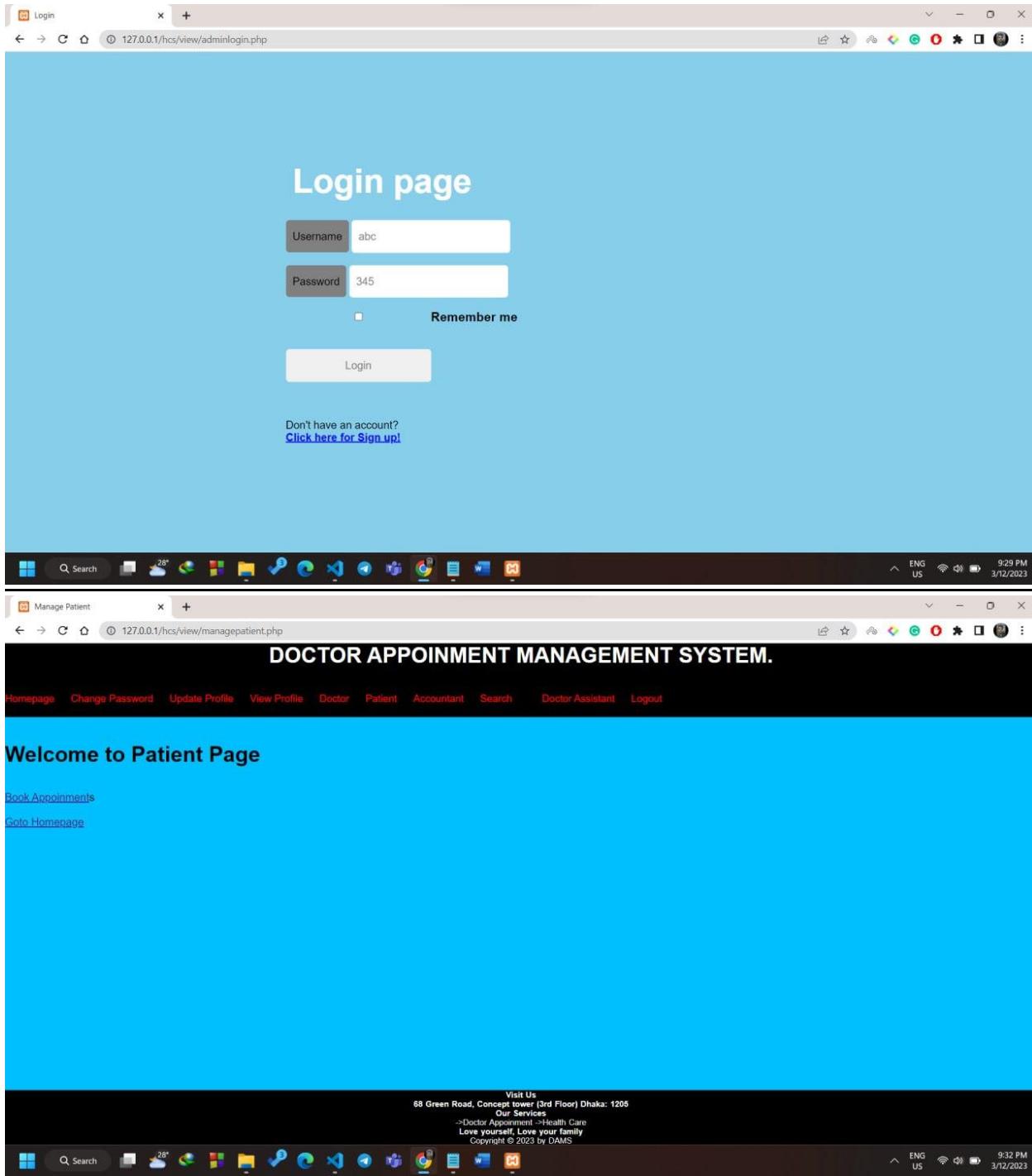
**Screenshot 1: Accountant Page**

The title bar of the browser window says "Manage Nurse". The URL in the address bar is "127.0.0.1/hcs/view/manageacc.php". The main content area displays the heading "DOCTOR APPOINTMENT MANAGEMENT SYSTEM." and "Welcome to Accountant Page". Below this, there are several hyperlinks: "View All Doctor", "Update Doctor List", "Access record", and "Goto Homepage". At the bottom of the page, there is a "Visit Us" section with the address "68 Green Road, Concept tower (3rd Floor) Dhaka: 1205", "Our Services" (listing "Doctor Appointment", "Health Care", "Love yourself, Love your family"), and a copyright notice "Copyright © 2023 by DAMS". The system tray at the bottom right shows "ENG US" and the date "3/12/2023" at 9:33 PM.

**Screenshot 2: Doctor Assistant Page**

The title bar of the browser window says "Doctor". The URL in the address bar is "127.0.0.1/hcs/view/assistant.php". The main content area displays the heading "DOCTOR APPOINTMENT MANAGEMENT SYSTEM." and "Welcome to Doctor Assistant Page". Below this, there are two hyperlinks: "Manage Appointment" and "Goto Homepage". At the bottom of the page, there is a "Visit Us" section with the same information as the first screenshot, and the system tray at the bottom right shows "ENG US" and the date "3/12/2023" at 9:33 PM.



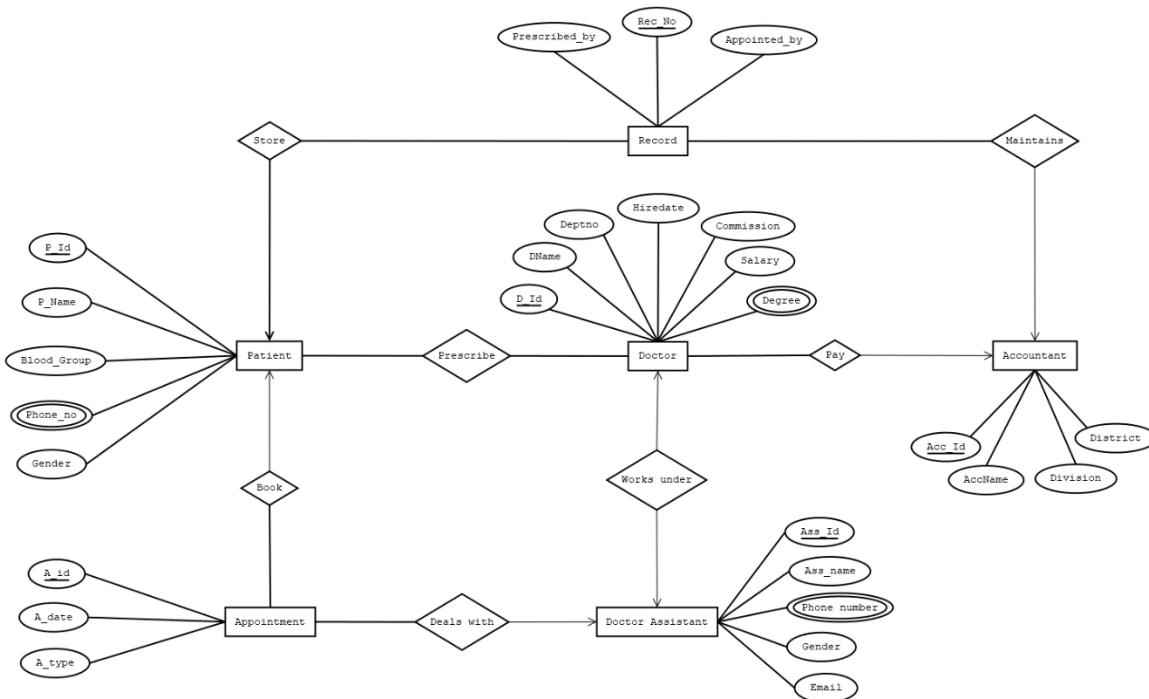


### **Scenario Description:**

In this database management system, we can handle the information of a “DOCTOR APPOINTMENT MANAGEMENT SYSTEM”. In this system, Patient can book appointment. A patient can book many appointments. A patient is identified by P\_id. A patient also has Pname,

Blood\_Group, Phone\_no, Gender. A patient can have multiple phone numbers. An appointment is identified by appointment\_id. It also has appointment\_date, appointment\_type. An assistant deal with appointments. An assistant has Ass\_id, Ass\_name, gender, phone\_number, email. An assistant works under a doctor. A doctor is identified by D\_id, and have Dname, Hiredate, Salary, Degree, Department, Commission. A doctor can have multiple degrees. Patient prescribed by doctor. A patient can prescribe by many doctors. A doctor can prescribe many patients. An accountant pays multiple doctors. An accountant has unique Acc\_id, and have Accname, Division, District. Accountant maintains records. One patient can have many records. A record is identified by Rec\_no and have Prescribed\_by, Appointed\_by.

### ER diagram:



### Normalization:

Prescribe- (**D\_id**, DName, Hiredate, Degree, Salary, Deptno, Comm, **P\_id**, PName, Blood\_Group, Phone\_no, Gender)

1NF: Degree and phone\_no are multivalued attributes.

2NF: **D\_id**, DName, Hiredate, Degree, Salary, Deptno, Comm

**P\_id**, PName, Blood\_Group, Phone\_no, Gender

3NF: No, transitive dependency found

D\_id, DName, Hiredate, Degree, Salary, Deptno, Comm

P\_id, PName, Blood\_Group, Phone\_no, Gender

Tables from Prescribe,

1. D\_id, DName, Hiredate, Salary, Deptno, Comm
2. D\_id, Degree -> Composite pk
3. P\_id, PName, Blood\_Group, Gender
4. P\_id, Phone\_no -> Composite pk
5. DP\_id, D\_id, P\_id

Book-( P\_id, PName, Blood\_Group, Phone\_no, Gender, A\_id, A\_date, A\_type)

1NF: phone\_no are multivalued attributes.

2NF: P\_id, PName, Blood\_Group, Phone\_no, Gender

A\_id, A\_date, A\_type

3NF: No, transitive dependency found

P\_id, PName, Blood\_Group, Phone\_no, Gender

A\_id, A\_date, A\_type

Table from Book

6. P\_id, PName, Blood\_Group, Phone\_no, Gender
7. P\_id, Phone\_no -> Composite pk
8. A\_id, A\_date, A\_type, P\_id

Deals with-( A\_id, A\_date, A\_type, Ass\_id, phone number, gender, email)

1NF: Phone\_no is a multivalued attribute.

2NF: A\_id, A\_date, A\_type

Ass\_id, phone number, gender, email

3NF: No transitive dependency found

A\_id, A\_date, A\_type

Ass\_id, phone number, gender, email

Tables from Deals with,

9. A\_id, A\_date, A\_type, Ass\_id
10. Ass\_id, phone number, gender, email

11. Ass\_id, phone number -> Composite pk

Works Under – (Ass\_id, phone number, gender, email, D\_id, DName, Hiredate, Degree, Salary, Deptno, Comm)

1NF: Degree and phone\_no are multivalued attributes.

2NF: Ass\_id, phone number, gender, email

D\_id, DName, Hiredate, Degree, Salary , Deptno, Comm

3NF: No transitive dependency found

Ass\_id, phone number, gender, email

D\_id, DName, Hiredate, Degree, Salary , Deptno, Comm

Tables from works under,

12. Ass\_id, gender, email, D\_id

13. Ass\_id, phone number -> Composite

14. D\_id, DName, Hiredate, Salary, deptno, Comm

15. D\_id, Degree -> Composite pk

Pays- (Acc\_id, Accname, Division, District, D\_id, DName, Hiredate, Degree, Salary, Deptno, Comm)

1NF: Degree is a multivalued attribute.

2NF: Acc\_id, Accname, Division, District,

D\_id, DName, Hiredate, Degree, Salary, Deptno, Comm

3NF: Acc\_id, Accname

Div\_id, Division, District

D\_id, DName, Hiredate, Degree, Salary, Deptno, Comm

Tables from Payment

16. Acc\_id, Accname, Div\_id

17. Div\_id, Division, District,

18. D\_id, DName, Hiredate, Salary, Deptno, Comm Acc\_id

19. D\_id, Degree ->composite pk

Maintain- (**Acc\_id**, Accname, Division, District, **Rec\_no**, Prescribed by, Appointed\_by)

1NF: No multivalued attribute.

2NF: **Acc\_id**, Accname, Division, District,

**Rec\_no**, Appointed\_by, Prescribed\_by

3NF: **Acc\_id**, Accname

**Dis\_id**, Division, District,

**Rec\_no**, Appointed\_by, Prescribed\_by

Tables from maintains

20. **Acc\_id**, Accname, **Dis\_id**

21. **Dis\_id**, Division, District

22. **Rec\_no**, Prescribed\_by, Appointed\_by, **Acc\_id**

Store- (**Rec\_no**, Prescribedby, Appointed\_by, **P\_id**, PName, Blood\_Group, Phone\_no, Gender)

1NF: Phone\_no is a multivalued attribute

2NF: **Rec\_no**, Appointed\_by, Prescribedby

**P\_id**, PName, Blood\_Group, Phone\_no, Gender

3NF: No transitive dependency found

**Rec\_no**, Appointed\_by, Prescribedby

**P\_id**, PName, Blood\_Group, Phone\_no, Gender

Tables from Store

23. **Rec\_no**, Prescribed\_by, Appointed\_by, **P\_id**

24. **P\_id**, PName, Blood\_Group, Gender

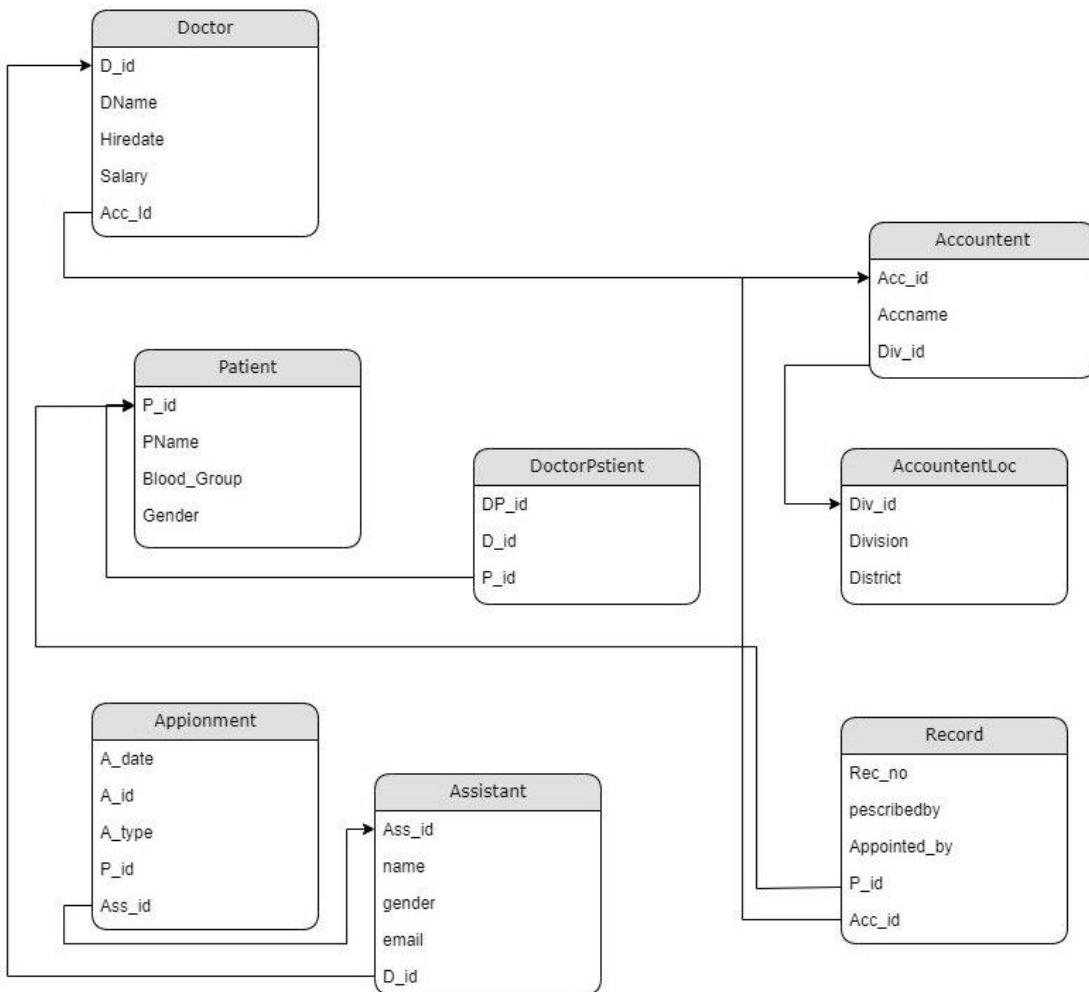
25. **P\_id,Phone\_no ->composite pk**

1. ~~D\_id, DName, Hiredate, Salary, deptno, comm~~
2. ~~D\_id, Degree -> Composite pk~~
3. ~~P\_id, PName, Blood\_Group, Gender~~
4. ~~P\_id, Phone\_no -> Composite pk~~
5. ~~DP\_id, D\_id, P\_id~~
6. ~~P\_id, PName, Blood\_Group, Gender~~
7. ~~P\_id, Phone\_no -> Composite pk~~
8. ~~A\_id, A\_date, A\_type, P\_id~~
9. ~~A\_id, A\_date, A\_type, Ass\_id~~
10. ~~Ass\_id, name, gender, email~~
11. ~~Ass\_id, phone number -> Composite pk~~
12. ~~Ass\_id, name, gender, email, D\_id~~
13. ~~Ass\_id, phone number -> Composite~~
14. ~~D\_id, DName, Hiredate, Salary, deptno, comm~~
15. ~~D\_id, Degree -> Composite pk~~
16. ~~Acc\_id, Accname, Div\_id~~
17. ~~Div\_id, Division, District~~
18. ~~D\_id, DName, Hiredate, Salary, deptno, comm, Acc\_id~~
19. ~~D\_id, Degree -> composite\_pk~~
20. ~~Acc\_id, Accname, Div\_id~~
21. ~~Div\_id, Division, District~~
22. ~~Rec\_no, Prescribedby, Appointed\_by, Acc\_id~~
23. ~~Rec\_no, Prescribedby, Appointed\_by, P\_id~~
24. ~~P\_id, PName, Blood\_Group, Gender~~
25. ~~P\_id, Phone\_no -> composite\_pk~~

#### Tables after normalization:

1. ~~D\_id, DName, Hiredate, Salary, deptno, comm, Acc\_id --Doctor~~
2. ~~D\_id, Degree -> Composite pk --DoctorDegree~~
3. ~~P\_id, PName, Blood\_Group, Gender --Patient~~
4. ~~P\_id, Phone\_no -> Composite pk --patientContact~~
5. ~~DP\_id, D\_id, P\_id --DoctorPatient~~
6. ~~A\_id, A\_date, A\_type, P\_id, Ass\_id --Appointment~~
7. ~~Ass\_id, name, gender, email, D\_id --Assistant~~
8. ~~Ass\_id, phone number -> Composite pk --AssitantContact~~
9. ~~Acc\_id, Accname, Div\_id --Accountant~~
10. ~~Div\_id, Division, District --AccountantLoc~~
11. ~~Rec\_no, Prescribedby, Appointed\_by, P\_id, Acc\_id --Record~~

### Schema Diagram:



## Table Creation:

**ORACLE Database Express Edition**

User LION

Home > SQL > SQL Commands

```
//Doctor
create table Doctor(
Did int,
dname varchar2(50),
hiredate date,
salary float,
accid int)

alter table doctor add constraint d_pk primary key(did)
alter table doctor add constraint d_fk foreign key(accid) references accountant(accid)

desc doctor
```

Save Run 22

Results Explain Describe Saved SQL History

Object Type TABLE Object DOCTOR

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
DOCTOR	DID	Number	-	-	0	1	-	-	
	DNAME	Varchar2	50	-	-	-	✓	-	
	HIREDATE	Date	7	-	-	-	✓	-	
	SALARY	Float	22	126	-	-	✓	-	
	ACCID	Number	-	-	0	-	✓	-	

1 - 5

Language: en-us Application Express 2.1.0.0.39 Copyright © 1999, 2006, Oracle. All rights reserved.

12:31 AM 3/9/2023

**ORACLE Database Express Edition**

User LION

Home > SQL > SQL Commands

```
//DoctorDegree
create table Doctordegree(
Did int,
Degree varchar2(50))

alter table Doctordegree add constraint dd_pk primary key(did,degree)

desc Doctordegree
```

Save Run 2

Results Explain Describe Saved SQL History

Object Type TABLE Object DOCTORDEGREE

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
DOCTORDEGREE	DID	Number	-	-	0	1	-	-	
	DEGREE	Varchar2	50	-	-	2	-	-	

1 - 2

Language: en-us Application Express 2.1.0.0.39 Copyright © 1999, 2006, Oracle. All rights reserved.

10:07 PM 3/11/2023

SQL Commands (4) Tumse Mohabbat Hai | +

ORACLE Database Express Edition

User LION

Home > SQL > SQL Commands

Autocommit

```
//Patient
create table patient(
pid int,
pname varchar2(50),
BloodGroup varchar2(10),
Gender varchar2(15))

alter table patient add constraint p_pk primary key(pid)

desc patient
```

Results Explain Describe Saved SQL History

Object Type TABLE Object PATIENT

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PATIENT	PID	Number	-	-	0	1	-	-	-
	PNAME	Varchar2	50	-	-	-	✓	-	-
	BLOODGROUP	Varchar2	10	-	-	-	✓	-	-
	GENDER	Varchar2	15	-	-	-	✓	-	-

1 - 4

Application Express 2.1.0.00.39  
Copyright © 1999, 2006, Oracle. All rights reserved.

Language: en-us

12:32 AM 3/9/2023

SQL Commands (4) Tumse Mohabbat Hai | +

ORACLE Database Express Edition

User LION

Home > SQL > SQL Commands

Autocommit

```
//PatientContact
create table patientContact(
pid int,
phoneno varchar2(15))

alter table patientContact add constraint pc_pk primary key(pid,phoneno)

desc patientcontact
```

Results Explain Describe Saved SQL History

Object Type TABLE Object PATIENTCONTACT

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PATIENTCONTACT	PID	Number	-	-	0	1	-	-	-
	PHONENO	Varchar2	15	-	-	2	-	-	-

1 - 2

Application Express 2.1.0.00.39  
Copyright © 1999, 2006, Oracle. All rights reserved.

Language: en-us

12:33 AM 3/9/2023

SQL Commands (4) Tumse Mohabbat Hai | +

ORACLE Database Express Edition

User LION

Home > SQL > SQL Commands

Autocommit Display 10

```
//DoctorPatient
Create table DoctorPatient(
    dpid int,
    did int,
    pid int)

alter table Doctorpatient add constraint dp_pk primary key(dpid)
alter table Doctorpatient add constraint dp_fk_d foreign key(did) references doctor(did)
alter table Doctorpatient add constraint dp_fk_p foreign key(pid) references patient(pid)

desc doctorpatient
```

23

Results Explain Describe Saved SQL History

Object Type TABLE Object DOCTORPATIENT

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
DOCTORPATIENT	DPID	Number	-	-	0	1	-	-	-
	Did	Number	-	-	0	-	✓	-	-
	PID	Number	-	-	0	-	✓	-	-

1 - 3

Application Express 2.1.0.0.39  
Copyright © 1999, 2006, Oracle. All rights reserved.

Language: en-us

12:33 AM 3/9/2023

SQL Commands (4) Tumse Mohabbat Hai | +

ORACLE Database Express Edition

User LION

Home > SQL > SQL Commands

Autocommit Display 10

```
//Appointment
Create table Appointment(
    Appid int,
    Appdate date,
    Apptype varchar2(50),
    pid int,
    Assid int)

alter table appointment add constraint app_pk primary key(appid)
alter table appointment add constraint app_fk_p foreign key(pid) references patient(pid)
alter table appointment add constraint app_fk_ass foreign key(assid) references assistant(assid)

desc appointment
```

23

Results Explain Describe Saved SQL History

Object Type TABLE Object APPOINTMENT

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
APPOINTMENT	APPID	Number	-	-	0	1	-	-	-
	APPDATE	Date	7	-	-	-	✓	-	-
	APPTYPE	Varchar2	50	-	-	-	✓	-	-
	PID	Number	-	-	0	-	✓	-	-
	ASSID	Number	-	-	0	-	✓	-	-

1 - 5

Application Express 2.1.0.0.39  
Copyright © 1999, 2006, Oracle. All rights reserved.

Language: en-us

12:33 AM 3/9/2023

SQL Commands x (4) Tumse Mohabbat Hai S x | +

ORACLE Database Express Edition

User LION

Home > SQL > SQL Commands

Autocommit Display 10

```
//Assistant
Create table assistant(
assid int,
assname varchar2(50),
gender varchar2(15),
email varchar2(50),
did int)

alter table Assistant add constraint ass_pk primary key(assid)
alter table assistant add constraint ass_fk foreign key(did) references doctor(did)

desc assistant
```

Results Explain Describe Saved SQL History

Object Type TABLE Object ASSISTANT

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ASSISTANT	ASSID	Number	-	-	0	1	-	-	-
	ASSNAME	Varchar2	50	-	-	-	✓	-	-
	GENDER	Varchar2	15	-	-	-	✓	-	-
	EMAIL	Varchar2	50	-	-	-	✓	-	-
	DID	Number	-	-	0	-	✓	-	-

1 - 5

Language: en-us Copyright © 1999, 2006, Oracle. All rights reserved. Application Express 2.1.0.0.39 12:34 AM 3/9/2023

SQL Commands x (4) Tumse Mohabbat Hai S x | +

ORACLE Database Express Edition

User LION

Home > SQL > SQL Commands

Autocommit Display 10

```
//AssistantContact
Create table assistantcontact(
Assid int,
phoneno varchar2(15))

alter table Assistantcontact add constraint assCon_pk primary key(assid,phoneno)

desc assistantcontact
```

Results Explain Describe Saved SQL History

Object Type TABLE Object ASSISTANTCONTACT

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ASSISTANTCONTACT	ASSID	Number	-	-	0	1	-	-	-
	PHONENO	Varchar2	15	-	-	2	-	-	-

1 - 2

Language: en-us Copyright © 1999, 2006, Oracle. All rights reserved. Application Express 2.1.0.0.39 12:34 AM 3/9/2023

SQL Commands (4) Tumse Mohabbat Hai | +

ORACLE Database Express Edition

User LION

Home > SQL > SQL Commands

Autocommit Display 10

```
//Accountant
Create table accountant(
accid int,
accname varchar2(50),
DivId int)

alter table Accountant add constraint acc_pk primary key(accid)
alter table Accountant add constraint acc_fk foreign key(divid) references AccountantLoc(divid)

desc accountant
```

Results Explain Describe Saved SQL History

Object Type TABLE Object ACCOUNTANT

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ACCOUNTANT	ACCID	Number	-	-	0	1	-	-	
	ACCNAME	Varchar2	50	-	-	-	✓	-	-
	DIVID	Number	-	-	0	-	✓	-	-
									1 - 3

Application Express 2.1.0.0.39  
Copyright © 1999, 2006, Oracle. All rights reserved.

Language: en-us

SQL Commands (4) Tomar Jonno Nilche Tare | +

ORACLE Database Express Edition

User LION

Home > SQL > SQL Commands

Autocommit Display 10

```
//AccountantLoc
Create table AccountantLoc(
divid int,
division varchar2(50),
district varchar2(50))

alter table AccountantLoc add constraint accloc_pk primary key(divid)

desc AccountantLoc
```

Results Explain Describe Saved SQL History

Object Type TABLE Object ACCOUNTANTLOC

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ACCOUNTANTLOC	DIVID	Number	-	-	0	1	-	-	
	DIVISION	Varchar2	50	-	-	-	✓	-	-
	DISTRICT	Varchar2	50	-	-	-	✓	-	-
									1 - 3

Application Express 2.1.0.0.39  
Copyright © 1999, 2006, Oracle. All rights reserved.

Language: en-us

The screenshot shows the Oracle Database Express Edition interface. In the top navigation bar, there are tabs for 'SQL Commands', 'Index', and 'ProjectGuidelinesMidtermADBM'. The main area displays an SQL script for creating a table 'record' with constraints and a corresponding object type 'RECORD'.

```
//record
create table record(
    Recno int,
    prescribedby varchar2(50),
    appointedby varchar2(50),
    Did int,
    accid int)

alter table record add constraint r_pk primary key(recno)
alter table record add constraint r_fk_p foreign key(pid) references patient(pid)
alter table record add constraint r_fk_acc foreign key(accid) references accountant(accid)

desc record
```

Below the SQL editor, the 'Object Type' dropdown is set to 'TABLE Object RECORD'. A table titled 'RECORD' is displayed:

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
RECORD	RECNO	Number	-	-	0	1	-	-	-
	PRESCRIBEDBY	VARCHAR2	50	-	-	-	✓	-	-
	APPOINTEDBY	VARCHAR2	50	-	-	-	✓	-	-
	PID	Number	-	-	0	-	✓	-	-
	ACCID	Number	-	-	0	-	✓	-	-

At the bottom of the interface, it shows 'Language: en-us', the system status bar with icons, and the footer 'Application Express 2.1.0.0.39 Copyright © 1999, 2006, Oracle. All rights reserved.' and the date '3/11/2023'.

### Queries for table creation:

//Doctor

create table Doctor(

Did int,

dname varchar2(50),

hiredate date,

salary float,

accid int)

alter table doctor add constraint d\_pk primary key(did)

alter table doctor add constraint d\_fk foreign key(accid) references accountant(accid)

//DoctorDegree

create table Doctordegree(

```
Did int,  
Degree varchar2(50))
```

```
alter table Doctordegree add constraint dd_pk primary key(did,degree)
```

```
//Patient  
create table patient(  
pid int,  
pname varchar2(50),  
BloodGroup varchar2(10),  
Gender varchar2(15))
```

```
alter table patient add constraint p_pk primary key(pid)
```

```
//PatientContact  
create table patientContact(  
pid int,  
phoneno varchar2(15))
```

```
alter table patientContact add constraint pc_pk primary key(pid,phoneno)
```

```
//DoctorPatient  
Create table DoctorPatient(  
dpid int,  
did int,  
pid int)
```

```
alter table Doctorpatient add constraint dp_pk primary key(dpid)
alter table Doctorpatient add constraint dp_fk_d foreign key(did) references doctor(did)
alter table Doctorpatient add constraint dp_fk_p foreign key(pid) references patient(pid)
```

//Appointment

```
Create table Appointment(
Appid int,
Appdate date,
Apptype varchar2(50),
pid int,
Assid int)
```

```
alter table appointment add constraint app_pk primary key(appid)
alter table appointment add constraint app_fk_p foreign key(pid) references patient(pid)
alter table appointment add constraint app_fk_ass foreign key(assid) references assistant(assid)
```

//Assistant

```
Create table assistant(
assid int,
assname varchar2(50),
gender varchar2(15),
email varchar2(50),
did int)
```

```
alter table Assistant add constraint ass_pk primary key(assid)
alter table assistant add constraint ass_fk foreign key(did) references doctor(did)
```

//AsstContact

Create table assistantcontact(

Assid int,

phoneno varchar2(15))

alter table Assistantcontact add constraint assCon\_pk primary key(assid,phoneno)

//Accountant

Create table accountant(

accid int,

accname varchar2(50),

Divid int)

alter table Accountant add constraint acc\_pk primary key(accid)

alter table Accountant add constraint acc\_fk foreign key(divid) references AccountantLoc(divid)

//AccountantLoc

Create table AccountantLoc(

divid int,

division varchar2(50),

district varchar2(50))

alter table AccountantLoc add constraint accloc\_pk primary key(divid)

//record

create table record(

```
Recno int,  
prescribedby varchar2(50),  
appointedby varchar2(50),  
Pid int,  
accid int)  
  
alter table record add constraint r_pk primary key(recno)  
alter table record add constraint r_fk_p foreign key(pid) references patient(pid)  
alter table record add constraint r_fk_acc foreign key(accid) references accountant(accid)
```

#### **Queries For Sequences:**

```
//Doctor(DOCSEQ_101)  
create sequence docseq  
increment by 1  
start with 101  
maxvalue 200  
nocache  
nocycle
```

```
//Patient(PATSEQ_5001)  
create sequence patseq  
increment by 1  
start with 5001  
maxvalue 6000  
nocache  
nocycle
```

```
//DoctorPatient(DPSEQ_6001)
```

```
create sequence DPseq
```

```
increment by 1
```

```
start with 6001
```

```
maxvalue 7000
```

```
nocache
```

```
necycle
```

```
//Appointment(APPSEQ_7001)
```

```
create sequence APPseq
```

```
increment by 1
```

```
start with 7001
```

```
maxvalue 8000
```

```
nocache
```

```
necycle
```

```
//Assistant(ASTSEQ_301)
```

```
create sequence ASTseq
```

```
increment by 1
```

```
start with 301
```

```
maxvalue 400
```

```
nocache
```

```
necycle
```

```
//Accountant(ACCSEQ_1)
```

```
create sequence ACCseq
```

```
increment by 5
```

```
start with 5
```

```
maxvalue 100
```

```
nocache
```

```
nocycle
```

```
//AccountantLoc(ACCLSEQ_1201)
```

```
create sequence ACCLseq
```

```
increment by 1
```

```
start with 1201
```

```
maxvalue 1300
```

```
nocache
```

```
nocycle
```

```
//Record(RECSEQ_8001)
```

```
create sequence RECseq
```

```
increment by 1
```

```
start with 8001
```

```
maxvalue 9000
```

```
nocache
```

```
nocycle
```

### **Index Creation:**

```
//doctor
```

```
Create index doctor_idx
```

```
on doctor(dname,salary)
```

```
//patient  
Create index patient_idx  
on patient(pname,bloodgroup)
```

```
//appointment  
create index appointment_idx  
on appointment(pid,appdate,apptype)
```

```
//assistant  
Create index assistant_idx  
on assistant(assname,email)
```

```
//accountant  
Create index accountant_idx  
on accountant(accid,accname)
```

## Data insertion:

**ORACLE Database Express Edition**

User LION

Home > SQL > SQL Commands

Autocommit

```
insert into doctor values(docseq.nextval,'Nibir', TO_DATE('22-JUN-2022', 'DD-MON-YYYY'),200000,15)
insert into doctor values(docseq.nextval,'Shadhin',TO_DATE('16-DEC-2020', 'DD-MON-YYYY'),300000,5)
insert into doctor values(docseq.nextval,'Foyosal',TO_DATE('31-DEC-2019', 'DD-MON-YYYY'),400000,5)
insert into doctor values(docseq.nextval,'Tuhin',TO_DATE('20-NOV-2019', 'DD-MON-YYYY'),500000,20)
insert into doctor values(docseq.nextval,'Nitul',TO_DATE('31-DEC-2019', 'DD-MON-YYYY'),600000,20)

select * from doctor;
```

Results Explain Describe Saved SQL History

DID	DNAME	HIREDATE	SALARY	ACCID
101	Nibir	22-JUN-22	200000	15
102	Shadhin	16-DEC-20	300000	5
103	Foyosal	31-DEC-19	400000	5
104	Tuhin	20-NOV-19	500000	20
105	Nitul	31-DEC-19	600000	20

5 rows returned in 0.00 seconds [CSV Export](#)

Language: en-us Application Express 2.1.0.00.39 Copyright © 1999, 2006, Oracle. All rights reserved.

10:11 PM 3/11/2023

**ORACLE Database Express Edition**

User LION

Home > SQL > SQL Commands

Autocommit

```
insert into doctorDEGREE values(101,'MBBS')
insert into doctordegree values(101,'FCPS')
insert into doctorDEGREE values(102,'MBBS')
insert into doctorDEGREE values(103,'MBBS')
insert into doctorDEGREE values(104,'MBBS')
insert into doctorDEGREE values(105,'MBBS')

select * from doctordegree;
```

Results Explain Describe Saved SQL History

ID	DEGREE
101	FCPS
101	MBBS
102	MBBS
103	MBBS
104	MBBS
105	MBBS

6 rows returned in 0.02 seconds [CSV Export](#)

Language: en-us Application Express 2.1.0.00.39 Copyright © 1999, 2006, Oracle. All rights reserved.

10:11 PM 3/11/2023

SQL Commands

User LION

Home > SQL > SQL Commands

Autocommit Display 10

```
insert into patient values(patseq.nextval,'Adnan','A+','Male')
insert into patient values(patseq.nextval,'Abir','O+','Male')
insert into patient values(patseq.nextval,'Farid','B-','Male')
insert into patient values(patseq.nextval,'Akhi','AB+','Female')
insert into patient values(patseq.nextval,'Tanzila','B+','Female')

select* from patient;
```

Results Explain Describe Saved SQL History

PID	PNAME	BLOODGROUP	GENDER
5001	Adnan	A+	Male
5002	Abir	O+	Male
5003	Farid	B-	Male
5004	Akhi	AB+	Female
5005	Tanzila	B+	Female

5 rows returned in 0.00 seconds [CSV Export](#)

Application Express 2.1.0.00.39  
Copyright © 1999, 2006, Oracle. All rights reserved.

Language: en-us

10:12 PM 3/11/2023

SQL Commands

User LION

Home > SQL > SQL Commands

Autocommit Display 10

```
insert into patientcontact values(5001,'+8801954775770')
insert into patientcontact values(5002,'+8801754775771')
insert into patientcontact values(5003,'+8801713575873')
insert into patientcontact values(5004,'+8801973575873')
insert into patientcontact values(5002,'+8801850201030')

select* from patientcontact;
```

Results Explain Describe Saved SQL History

PID	PHONENO
5001	+8801954775770
5002	+8801754775771
5002	+8801850201030
5003	+8801713575873
5004	+8801973575873

5 rows returned in 0.00 seconds [CSV Export](#)

Application Express 2.1.0.00.39  
Copyright © 1999, 2006, Oracle. All rights reserved.

Language: en-us

10:12 PM 3/11/2023

SQL Commands x +

127.0.0.1:8080/apex/f?p=4500:1003:7473850913643872::NO::

ORACLE Database Express Edition

User LION

Home > SQL > SQL Commands

Autocommit

```
insert into DoctorPatient values(DPseq.nextval,101,5001)
insert into DoctorPatient values(DPseq.nextval,101,5002)
insert into DoctorPatient values(DPseq.nextval,102,5003)
insert into DoctorPatient values(DPseq.nextval,103,5004)
insert into DoctorPatient values(DPseq.nextval,104,5005)

select* from DoctorPatient
```

Results Explain Describe Saved SQL History

DPID	DID	PID
6001	101	5001
6002	101	5002
6003	102	5003
6004	103	5004
6005	104	5005

5 rows returned in 0.00 seconds [CSV Export](#)

Application Express 2.1.0.00.39  
Copyright © 1999, 2006, Oracle. All rights reserved.

Language: en-us

SQL Commands x +

127.0.0.1:8080/apex/f?p=4500:1003:7473850913643872::NO::

ORACLE Database Express Edition

User LION

Home > SQL > SQL Commands

Autocommit

```
insert into Appointment values(APPseq.nextval,TO_DATE('22-JUN-2022', 'DD-MON-YYYY'),'Cancer',5001,301)
insert into Appointment values(APPseq.nextval,TO_DATE('25-JUN-2023', 'DD-MON-YYYY'),'Cancer',5001,301)
insert into Appointment values(APPseq.nextval,TO_DATE('21-JUN-2023', 'DD-MON-YYYY'),'Influenza',5002,302)
insert into Appointment values(APPseq.nextval,TO_DATE('26-JUN-2023', 'DD-MON-YYYY'),'Aids',5003,303)
insert into Appointment values(APPseq.nextval,TO_DATE('29-JUN-2023', 'DD-MON-YYYY'),'Common cold',5004,304)

select * from appointment
```

Results Explain Describe Saved SQL History

APPID	APPPDATE	APPTYPE	PID	ASSID
7001	22-JUN-22	Cancer	5001	301
7002	25-JUN-23	Cancer	5002	301
7003	21-JUN-23	Influenza	5003	302
7004	26-JUN-23	Aids	5004	303
7005	29-JUN-23	Common cold	5005	304

5 rows returned in 0.02 seconds [CSV Export](#)

Application Express 2.1.0.00.39  
Copyright © 1999, 2006, Oracle. All rights reserved.

Language: en-us

SQL Commands x +

127.0.0.1:8080/apex/f?p=4500:1003:7473850913643872::NO::

ORACLE Database Express Edition

10:12 PM 3/11/2023

SQL Commands

ORACLE Database Express Edition

User LION

Home > SQL > SQL Commands

Autocommit

```
insert into Assistant values(ASTseq.nextval,'Shyama','Female','shymail@gmail.com',101)
insert into Assistant values(ASTseq.nextval,'Shishir','Female','shishirpharma@gmail.com',102)
insert into Assistant values(ASTseq.nextval,'Abida','Female','abida0@gmail.com',103)
insert into Assistant values(ASTseq.nextval,'Mubin','Male','mubin211@gmail.com',104)
insert into Assistant values(ASTseq.nextval,'Asha','Female','asha89@gmail.com',105)

select* from Assistant
```

Results Explain Describe Saved SQL History

ASSID	ASSNAME	GENDER	EMAIL	DID
301	Shyama	Female	shymail@gmail.com	101
302	Shishir	Female	shishirpharma@gmail.com	102
303	Abida	Female	abida0@gmail.com	103
304	Mubin	Male	mubin211@gmail.com	104
305	Asha	Female	asha89@gmail.com	105

5 rows returned in 0.00 seconds

[CSV Export](#)

Application Express 2.1.0.0.39  
Copyright © 1999, 2006, Oracle. All rights reserved.

Language: en-us

10:13 PM 3/11/2023

SQL Commands

ORACLE Database Express Edition

User LION

Home > SQL > SQL Commands

Autocommit

```
insert into assistantcontact values(301,'+8801647390996')
insert into assistantcontact values(302,'+8801950120661')
insert into assistantcontact values(303,'+8801680706080')
insert into assistantcontact values(304,'+8801445450985')
insert into assistantcontact values(305,'+8801952528457')

select * from assistantcontact
```

Results Explain Describe Saved SQL History

ASSID	PHONENO
301	+8801647390996
302	+8801950120661
303	+8801680706080
304	+8801445450985
305	+8801952528457

5 rows returned in 0.00 seconds

[CSV Export](#)

Application Express 2.1.0.0.39  
Copyright © 1999, 2006, Oracle. All rights reserved.



SQL Commands

User LION

Home > SQL > SQL Commands

Autocommit

```
insert into Accountant values(ACCseq.nextval,'Naim',1201)
insert into Accountant values(ACCseq.nextval,'Anik',1202)
insert into Accountant values(ACCseq.nextval,'Siam',1203)
insert into Accountant values(ACCseq.nextval,'Anika',1204)
insert into Accountant values(ACCseq.nextval,'Anjum',1205)

SELECT * FROM ACCOUNTANT
```

Results Explain Describe Saved SQL History

ACCID	ACCNAME	DIVID
5	Naim	1201
10	Anik	1202
15	Siam	1203
20	Anika	1204
25	Anjum	1205

5 rows returned in 0.00 seconds [CSV Export](#)

Application Express 2.1.0.0.39  
Copyright © 1999, 2006, Oracle. All rights reserved.

Language: en-us

10:16 PM 3/11/2023

SQL Commands

User LION

Home > SQL > SQL Commands

Autocommit

```
insert into AccountantLoc values(ACCLseq.nextval,'DHAKA','TANGAIL')
insert into AccountantLoc values(ACCLseq.nextval,'BARISHAL','BHOLA')
insert into AccountantLoc values(ACCLseq.nextval,'MYMENSINGH','JAMALPUR')
insert into AccountantLoc values(ACCLseq.nextval,'CHATTAGRAM','NOAKHALI')
insert into AccountantLoc values(ACCLseq.nextval,'KHULNA','JESSORE')
insert into AccountantLoc values(ACCLseq.nextval,'MYMENSINGH','SHERPUR')
insert into AccountantLoc values(ACCLseq.nextval,'MYMENSINGH','NETROKONA')
insert into AccountantLoc values(ACCLseq.nextval,'DHAKA','KISARGONJ')

select *
from accountantloc
```

Results Explain Describe Saved SQL History

DIVID	DIVISION	DISTRICT
1201	DHAKA	TANGAIL
1202	BARISHAL	BHOLA
1203	MYMENSINGH	JAMALPUR
1204	CHATTAGRAM	NOAKHALI
1205	KHULNA	JESSORE
1206	MYMENSINGH	SHERPUR
1207	MYMENSINGH	NETROKONA
1208	DHAKA	KISARGONJ

8 rows returned in 0.00 seconds [CSV Export](#)

Application Express 2.1.0.0.39  
Copyright © 1999, 2006, Oracle. All rights reserved.

10:17 PM 3/11/2023

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The SQL code entered is:

```

insert into record values(recseq.nextval,'Nibir','Shyama',5001,15)
insert into record values(recseq.nextval,'Nibir','Shyama',5002,15)
insert into record values(recseq.nextval,'Shadin','Shishir',5003,5)
insert into record values(recseq.nextval,'Foysal','Abida',5004,5)
insert into record values(recseq.nextval,'Tuhin','Hubin',5005,20)
select * from record

```

The results section displays the following table:

RECNNO	PREScribedBY	APPOINTEDBY	PID	ACCID
8001	Nibir	Shyama	5001	15
8002	Nibir	Shyama	5002	15
8003	Shadin	Shishir	5003	5
8004	Foysal	Abida	5004	5
8005	Tuhin	Hubin	5005	20

5 rows returned in 0.00 seconds [CSV Export](#)

Language: en-us Application Express 2.1 0.00.39 Copyright © 1999, 2006, Oracle. All rights reserved.

### Queries for data insertion:

//Doctor

```
insert into doctor values(docseq.nextval,'Nibir', TO_DATE('22-JUN-2022', 'DD-MON-YYYY'),200000,15)
```

```
insert into doctor values(docseq.nextval,'Shadhin',TO_DATE('16-DEC-2020', 'DD-MON-YYYY'),300000,5)
```

```
insert into doctor values(docseq.nextval,'Foysal',TO_DATE('31-DEC-2019', 'DD-MON-YYYY'),400000,5)
```

```
insert into doctor values(docseq.nextval,'Tuhin',TO_DATE('20-NOV-2019', 'DD-MON-YYYY'),500000,20)
```

```
insert into doctor values(docseq.nextval,'Nitu',TO_DATE('31-DEC-2019', 'DD-MON-YYYY'),600000,20)
```

//Doctordegree

```
insert into doctorDEGREE values(101,'MBBS')
```

```
insert into doctordegree values(101,'FCPS')
```

```
insert into doctorDEGREE values(102,'MBBS')
insert into doctorDEGREE values(103,'MBBS')
insert into doctorDEGREE values(104,'MBBS')
insert into doctorDEGREE values(105,'MBBS')

//patient
insert into patient values(patseq.nextval,'Adnan','A+','Male')
insert into patient values(patseq.nextval,'Abir','O+','Male')
insert into patient values(patseq.nextval,'Farid','B-','Male')
insert into patient values(patseq.nextval,'Akhi','AB+','Female')
insert into patient values(patseq.nextval,'Tanzila','B+','Female')

//patientContact
insert into patientcontact values(5001,'+8801954775770')
insert into patientcontact values(5002,'+8801754775771')
insert into patientcontact values(5003,'+8801713575873')
insert into patientcontact values(5004,'+8801973575873')
insert into patientcontact values(5002,'+8801850201030')

//DoctorPatient
insert into DoctorPatient values(DPseq.nextval,101,5001)
insert into DoctorPatient values(DPseq.nextval,101,5002)
insert into DoctorPatient values(DPseq.nextval,102,5003)
insert into DoctorPatient values(DPseq.nextval,103,5004)
insert into DoctorPatient values(DPseq.nextval,104,5005)

//Appointment
```

```
insert into Appointment values(APPseq.nextval,TO_DATE('22-JUN-2022', 'DD-MON-YYYY'),'Cancer',5001,301)
```

```
insert into Appointment values(APPseq.nextval,TO_DATE('25-JUN-2023', 'DD-MON-YYYY'),'Cancer',5001,301)
```

```
insert into Appointment values(APPseq.nextval,TO_DATE('21-JUN-2023', 'DD-MON-YYYY'),'Influenza',5002,302)
```

```
insert into Appointment values(APPseq.nextval,TO_DATE('26-JUN-2023', 'DD-MON-YYYY'),'Aids',5003,303)
```

```
insert into Appointment values(APPseq.nextval,TO_DATE('29-JUN-2023', 'DD-MON-YYYY'),'Common cold',5004,304)
```

```
//Assistant
```

```
insert into Assistant values(ASTseq.nextval,'Shyama','Female','shyma11@gmail.com',101)
```

```
insert into Assistant values(ASTseq.nextval,'Shishir','Female','shishirpharma@gmail.com',102)
```

```
insert into Assistant values(ASTseq.nextval,'Abida','Female','abida0@gmail.com',103)
```

```
insert into Assistant values(ASTseq.nextval,'Mubin','Male','mubin211@gmail.com',104)
```

```
insert into Assistant values(ASTseq.nextval,'Asha','Female','asha89@gmail.com',105)
```

```
//AssistantContact
```

```
insert into assistantcontact values(301,'+8801647390996')
```

```
insert into assistantcontact values(302,'+8801950120661')
```

```
insert into assistantcontact values(303,'+8801680706080')
```

```
insert into assistantcontact values(304,'+8801445450985')
```

```
insert into assistantcontact values(305,'+8801952528457')
```

```
//Accountant
```

```
insert into Accountant values(ACCseq.nextval,'Naim',1201)
```

```
insert into Accountant values(ACCseq.nextval,'Anik',1202)
insert into Accountant values(ACCseq.nextval,'Siam',1203)
insert into Accountant values(ACCseq.nextval,'Anika',1204)
insert into Accountant values(ACCseq.nextval,'Anjum',1205)

//AccountantLoc
insert into AccountantLoc values(ACCLseq.nextval,'DHAKA','TANGAIL')
insert into AccountantLoc values(ACCLseq.nextval,'BARISHAL','BHOLA')
insert into AccountantLoc values(ACCLseq.nextval,'MYMENSINGH','JAMALPUR')
insert into AccountantLoc values(ACCLseq.nextval,'CHATTAGRAM','NOAKHALI')
insert into AccountantLoc values(ACCLseq.nextval,'KHULNA','JESSORE')
insert into AccountantLoc values(ACCLseq.nextval,'MYMENSINGH','SHERPUR')
insert into AccountantLoc values(ACCLseq.nextval,'MYMENSINGH','NETROKONA')
insert into AccountantLoc values(ACCLseq.nextval,'DHAKA','KISARGONJ')

//Record
insert into record values(recseq.nextval,'Nibir','Shyama',5001,15)
insert into record values(recseq.nextval,'Nibir','Shyama',5002,15)
insert into record values(recseq.nextval,'Shadin','Shishir',5003,5)
insert into record values(recseq.nextval,'Foysal','Abida',5004,5)
insert into record values(recseq.nextval,'Tuhin','Mubin',5005,20)
```

### Create user, role and assigning role:

- Create a role administrator. The role administrator has access to all the object privileges. Ekramul (password: administrator) is the administrator.

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The SQL editor contains the following commands:

```
CREATE ROLE administrator
GRANT ALL PRIVILEGES ON accountant TO administrator
CREATE USER Ekramul IDENTIFIED BY administrator
GRANT administrator TO Ekramul
```

Below the SQL editor, the message "Statement processed." is displayed, along with a timestamp of "0.00 seconds". At the bottom of the interface, the Application Express version "2.1 0.00.39" and copyright information "Copyright © 1999, 2006, Oracle. All rights reserved." are visible.

## Query Writing:

### Single row function:

1. Write a query to display the information of the doctor named foysal.

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The SQL command window contains the following code:

```
//Case conversion
select *
from doctor
where lower(dname) = 'foysal'.
```

The results pane shows a single row of data:

DID	DNAME	HIREDATE	SALARY	ACCID
103	Foysal	31-DEC-19	400000	5

1 rows returned in 0.00 seconds

At the bottom right of the interface, there is a red circle with the number 2.

2. Display the name, salary, comm and mod of comm and salary whose department no is 10

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The SQL command window contains the following code:

```
select dname, salary, comm, MOD(salary,comm)
from doctor
where deptno = 10;
```

The results pane shows two rows of data:

DNAME	SALARY	COMM	MOD(SALARY,COMM)
Nibir	200000	5000	0
Tuhin	500000	-	-

2 rows returned in 0.00 seconds

At the bottom right of the interface, there is a red circle with the number 3.

3. Display the doctor's name and the total number of weeks have they been attending whose department no is 10.

## Group function:

1. Display department wise Maximum salary, whose salary is higher than 300000.

SQL Commands

127.0.0.1:8080/apex/f?p=4500:1003:7473850913643872::NO::

ORACLE Database Express Edition

User LION

Home > SQL > SQL Commands

Autocommit

```
select deptno, max(salary)
from doctor
group by deptno
having max(salary)>30000
```

1

Results Explain Describe Saved SQL History

DEPTNO	MAX(SALARY)
30	600000
20	400000
10	500000

3 rows returned in 0.00 seconds [CSV Export](#)

Application Express 2.1.0.0.39  
Copyright © 1999, 2006, Oracle. All rights reserved.

Language: en-us

10:45 PM 3/11/2023

2. Count the number of doctors who works under department 20.

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The SQL command entered is:

```
select count(*)
from doctor
where deptno = 20;
```

The results section shows a single row with the value 2, indicating there are 2 doctors in department 20. The status bar at the bottom right shows the date and time as 3/11/2023 10:48 PM.

3. Display the maximum average salary

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The SQL command entered is:

```
select max(avg(comm))
from doctor
group by deptno;
```

The results section shows a single row with the value 5000, indicating the maximum average salary is 5000. The status bar at the bottom right shows the date and time as 3/11/2023 10:58 PM.

## Subquery:

1. Display the doctor name and hiredate who joined before Shadhin.

The screenshot shows the Oracle Database Express Edition interface. The SQL Commands window contains the following query:

```
select dname,hiredate
from doctor
where hiredate < (select hiredate
from doctor
where dname = 'Shadhin')
```

The results show three rows:

DNAME	HIREDATE
Foyal	31-DEC-19
Tuhin	20-NOV-19
Nitu	31-DEC-19

3 rows returned in 0.00 seconds

Language: en-us Application Express 2.1.0.0.39 Copyright © 1999, 2006, Oracle. All rights reserved. 11:06 PM 3/11/2023

2. Display the doctor name who get the lowest salary department wise.

The screenshot shows the Oracle Database Express Edition interface. The SQL Commands window contains the following query:

```
select dname,salary
from doctor
where salary in (
select min(salary)
from doctor
group by deptno)
```

The results show four rows:

DNAME	SALARY
Nitu	60000
Shadhin	300000
Nibir	200000

3 rows returned in 0.00 seconds

Language: en-us Application Express 2.1.0.0.39 Copyright © 1999, 2006, Oracle. All rights reserved. 11:12 PM 3/11/2023

3. What is the list of doctors' department names, salaries, and department numbers, where the salary is higher than any doctor's salary in department 20, and the department number is not 20?

The screenshot shows the Oracle Database Express Edition interface. The SQL command entered is:

```
select dname,salary,deptno
from doctor
where salary > any(select salary
from doctor
where deptno = 20) and deptno<>20;
```

The results table is:

DNAME	SALARY	DEPTNO
Tuhin	500000	10
Nitu	600000	30

2 rows returned in 0.00 seconds

CSV Export

Language: en-us Application Express 2.1 0.00.39 Copyright © 1999, 2006, Oracle. All rights reserved.

## Joining:

1. Display the name of doctor who lives in dhaka.

The screenshot shows the Oracle Database Express Edition interface. The SQL command entered is:

```
//select the name of doctor who lives in dhaka
select accid,accname
from accountant,accountantloc
where division = 'DHAKA' and accountant.divid = accountantloc.divid
```

The results table is:

ACCID	ACCNAME
5	Naim

1 rows returned in 0.00 seconds

CSV Export

Language: en-us Application Express 2.1 0.00.39 Copyright © 1999, 2006, Oracle. All rights reserved.

2. What is the list of doctors' names and degrees using outer join.

The screenshot shows the Oracle Database Express Edition interface. The SQL Commands window displays the following code:

```
//Display doctor's name and their degree using outer join
select dname, degree
from doctor, doctordegree
where doctor.did (+)= doctordegree.did
```

The results table shows the following data:

DNAME	DEGREE
Nibir	FCPS
Nibir	MBBS
Shadhin	MBBS
Foysal	MBBS
Tuhin	MBBS
Nitu	MBBS

6 rows returned in 0.00 seconds

CSV Export

Language: en-us Application Express 2.1.0.0.39 Copyright © 1999, 2006, Oracle. All rights reserved. 11:28 PM 3/11/2023

3. Display the Doctor's name and their degree who in not FCPS using Natural join

The screenshot shows the Oracle Database Express Edition interface. The SQL Commands window displays the following code:

```
//Display doctor's name and their degree who is not FCPS using Natural join
select dname, degree
from doctor natural join doctordegree
where degree <> 'FCPS'
```

The results table shows the following data:

DNAME	DEGREE
Nibir	MBBS
Shadhin	MBBS
Foysal	MBBS
Tuhin	MBBS
Nitu	MBBS

5 rows returned in 0.00 seconds

CSV Export

Language: en-us Application Express 2.1.0.0.39 Copyright © 1999, 2006, Oracle. All rights reserved. 11:35 PM 3/11/2023

## View:

1. Create a view to display doctor name, hiredate and salary who works under deptno 10.

The screenshot shows the Oracle Database Express Edition interface. In the SQL Commands window, the following SQL code is entered and executed:

```
//Create view to display doctor name, hiredate and salary who works under deptno 10

create view doctor_view
as select dname, hiredate, salary
from doctor
where deptno = 10

describe doctor_view

select * from doctor_view
```

The results show a table with three columns: DNAME, HIREDATE, and SALARY. Two rows are returned:

DNAME	HIREDATE	SALARY
Nibir	22-JUN-22	200000
Tuhin	20-NOV-19	500000

2 rows returned in 0.00 seconds

CSV Export

Application Express 2.1.0.0.39  
Copyright © 1999, 2006, Oracle. All rights reserved.  
Language: en-us

2. Create a complex view to display Accountant's name and their division accordingly.

The screenshot shows the Oracle Database Express Edition interface. In the SQL Commands window, the following SQL code is entered and executed:

```
//Create a complex view to display Accountant's name and their division name

CREATE VIEW accountant_view AS
SELECT a.accname, al.division
FROM accountant a , accountantloc al
where a.divid = al.divid

describe accountant_view

select * from accountant_view
```

The results show a table with two columns: ACCNAME and DIVISION. Five rows are returned:

ACCCNAME	DIVISION
Naim	DHAKA
Anik	BARISHAL
Siam	MYMENSINGH
Anika	CHATTAGRAM
Anjum	KHULNA

5 rows returned in 0.00 seconds

CSV Export

Application Express 2.1.0.0.39  
Copyright © 1999, 2006, Oracle. All rights reserved.  
Language: en-us

3. Create a view to display patient id, name and their blood group; Do not allow DML operation for the view.

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The URL is 127.0.0.1:8080/apex/f?p=4500:1003:7473850913643872::NO:::1. The SQL command entered is:

```
//create view to display patient id, name, blood group; do not allow DML operation
to the view.

create view patient_view
as select pid,pname,bloodgroup
from patient
with read only

select * from patient_view
```

The results section shows a table with the following data:

PID	PNAME	BLOODGROUP
5001	Adnan	A+
5002	Abir	O+
5003	Farid	B-
5004	Akhi	AB+
5005	Tanzila	B+

5 rows returned in 0.00 seconds. A CSV Export link is available.

Language: en-us Application Express 2.1.0.0.39 Copyright © 1999, 2006, Oracle. All rights reserved.

### Synonym:

1. Create a synonym for patient\_view view.

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The URL is 127.0.0.1:8080/apex/f?p=4500:1003:7473850913643872::NO:::1. The SQL command entered is:

```
//Create synonym for Patient_view View

Create synonym p_view
for patient_view
```

The results section displays the message: "Synonym created." and "0.00 seconds".

Language: en-us Application Express 2.1.0.0.39 Copyright © 1999, 2006, Oracle. All rights reserved.

2. Create synonym for doctor table.

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The title bar reads "SQL Commands" and "ORACLE Database Express Edition". The URL is "127.0.0.1:8080/apex/f?p=4500:1003:7473850913643872::NO::". The main area contains the following SQL code:

```
//Create synonym for doctor table
create synonym doc.
for doctor;
```

Below the code, the message "Synonym created." is displayed. The status bar at the bottom indicates "0.00 seconds". The bottom right corner shows the Windows taskbar with the date and time as "12:24 AM 3/12/2023".

3. Delete the created synonym for doctor table.

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The title bar reads "SQL Commands" and "ORACLE Database Express Edition". The URL is "127.0.0.1:8080/apex/f?p=4500:1003:7473850913643872::NO::". The main area contains the following SQL code:

```
//drop the synonym for doctor table
drop synonym doc;
```

Below the code, the message "Synonym dropped." is displayed. The status bar at the bottom indicates "0.02 seconds". The bottom right corner shows the Windows taskbar with the date and time as "12:25 AM 3/12/2023".

### **Relational Algebra:**

#### 1. Project Operation( $\pi$ ):

->To project the DoctorID and Name attributes from the Doctor table :

$$\pi_{did, dname}(\text{Doctor})$$

->To project only the Phone number attributes from the Patient Contact table:

$$\pi_{Phoneno}(\text{PatientContact})$$

#### 2. Selection Operation( $\sigma$ ):

->To select all the Accountants who lives in Dhaka Division:

$$\sigma \text{Division}='DHAKA'(\text{Accountant})$$

->To select all Patients name whose PID = 5002 :

$$\pi_{Name}(\sigma \text{PID}=5002(\text{Patient}))$$

#### 3. Cartesian Product Operation:

->To find all possible Appointments for each doctor :

$$\text{DoctorAppointment} <- \text{Doctor} \times \text{Appointment}$$

#### 4. Union Operation( $\cup$ ) :

->To combine all Appointments for June 21th,2023 and June 25<sup>th</sup>,2023 :

$$\text{Appointment\_1} <- \sigma \text{Date}='2023-06-21'(\text{Appointment})$$
$$\text{Appointment\_2} <- \sigma \text{Date}='2023-06-25'(\text{Appointment})$$
$$\text{Appointment\_1} \cup \text{Appointment\_2}$$

#### 5. Rename Operation ( $\rho$ ) :

->To rename the ACCNAME attribute in the Accountant table to Accountant Name

$$\rho \text{ Accountant Name/ACCNAME}(\text{Accountant})$$

**Conclusion:**

The Doctor Appointment Management System presented in this case study is an efficient and comprehensive system for managing patient appointments, medical records. The system also maintains medical records, identifies patients prescribed by doctors. The system provides a seamless approach to patient care and management, resulting in a more efficient and effective doctor appointment system. To improve this existing project, we can add several features, such as automated reminder notifications for patients about their appointments, online payment options for patients, and the ability for doctors to access medical records remotely. Finally, we can make the system scalable and adaptable to accommodate the growth of the healthcare system and future technological advancements.