HOSPITAL DATABASE

جامعة جدة
University of Jeddah

GROUP MEMBERS

Wateen Abdulkareem Alharthy	2413523
Wareef Abdulelah Fallatah	2411473
Joud Abdullah Alzahrani	2411442
Wasan Fahad Alghamdi	2411315

Course suction | IT2

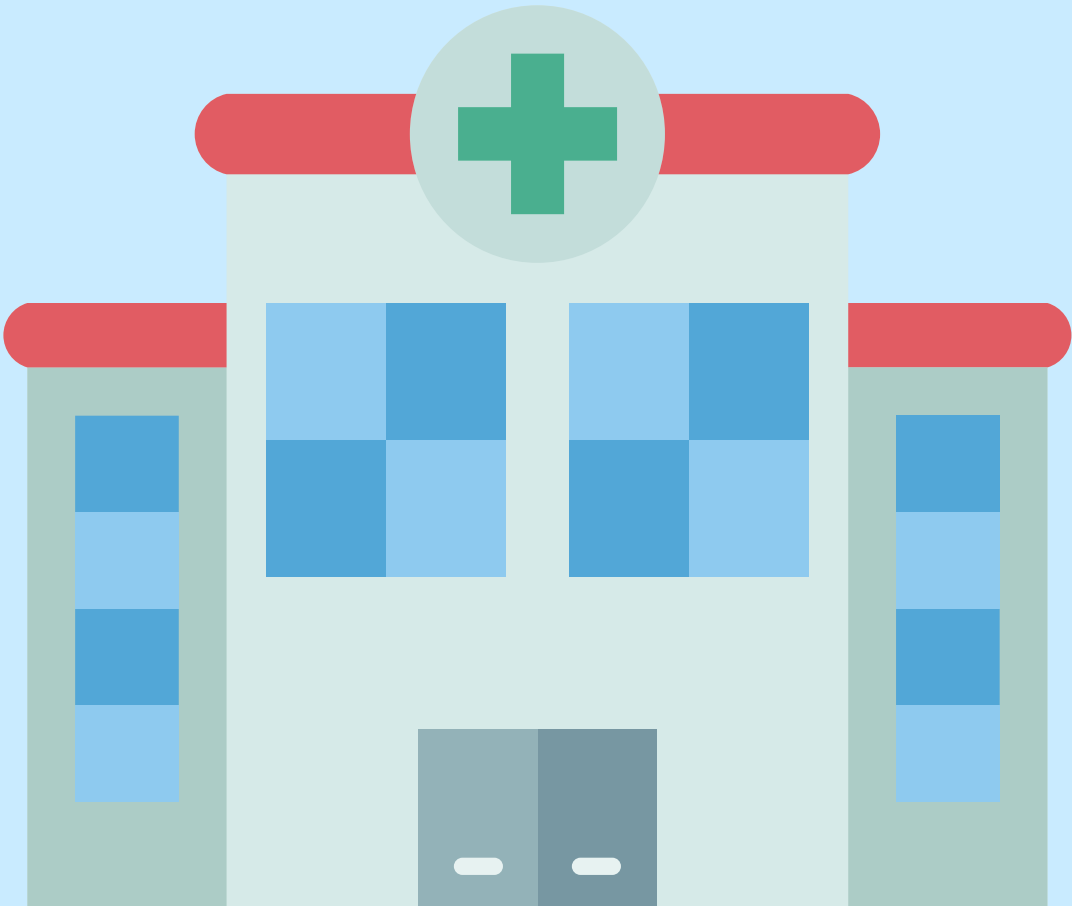
Course instructor | Dr.Mashael Khayyat

CEDS 215

TABLE OF CONTENT

Introduction

Here is an example of a hospital ER diagram. The diagram illustrates the hospital database and its associated fields, including six main entities: Doctors, Tools, Appointments, Departments, Patients, and Suppliers. The diagram specifies their attributes and relationships, aiding the hospital in efficient operations and patient retention.



PROBLEM DESCRIPTION:

HOSPITALS HAVE FACED VARIOUS CHALLENGES IN DEALING WITH PATIENTS AND MEDICAL STAFF. THEREFOR,A DATABASE HAS BEEN CREATED FOR HOSPITALS TO SOLVE THESE PROPLEMS.

SOME OF THESE PROPLEMS INCLUDE:

- Excessive crowding and randomness due to the lack of an appointment system
- Dealing with another patient's information by mistake.
- Legal and Regulatory Compliance: Hospitals are subject to a number of laws and guidelines pertaining to healthcare. Noncompliance may result in penalties, legal troubles, and reputational harm to the organization.
- System Downtime: Unplanned downtime has the potential to interfere with hospital operations and impact patient care. It is imperative to put backup system and rapid recovery procedures into place.

PHASE 1

INFORMATION IDENTIFICATION:

SOME OF THE INFORMATION THAT WOULD HELP SOLVE THE PROBLEM ARE:

1. *PATIENTS:*
- NAME
- NUMBER
- ADDRESS
- CONTACT DETAILS

2. *DOCTORS:*
- NAME
- CONTACT DETAILS
- SPECIALIZATION

3. *APPOINTMENTS:*
- APPOINTMENT NUMBER
- PATIENT ASSOCIATED WITH THE APPOINTMENT
- DOCTOR ASSOCIATED WITH THE APPOINTMENT
- DATE AND TIME OF THE APPOINTMENT

4. *DEPARTMENTS:*
- DEPARTMENT NAME
- DEPARTMENT ID

5. *SUPPLIERS:*
- SUPPLIER NAME
- CONTACT DETAILS
- PRODUCTS OFFERED
- PRICING

6. *TOOLS:*
- NAME
- QUANTITY
-SUPPLIER_ID

BY CONSIDERING THESE ENTITIES AND THEIR ATTRIBUTES, WE CAN CREATE AN ENTITY-RELATIONSHIP DIAGRAM (ERD) FOR THE HOSPITAL SYSTEM, DETAILING THEIR RELATIONSHIPS AND CONNECTIONS WITHIN THE SYSTEM.

INITIAL ENTITIES:

Patients

- Appointment ID
- First Name
- Last Name
- Date Of Birth
- Gender
- Phone Number
- Adress

Supplier

- Supplier ID
- Name

Appointment

- Appointment ID
- Patient ID
- Doctor ID
- Appointment Date

Tools

- Tool ID
- Name
- Quantity
- Supplier ID

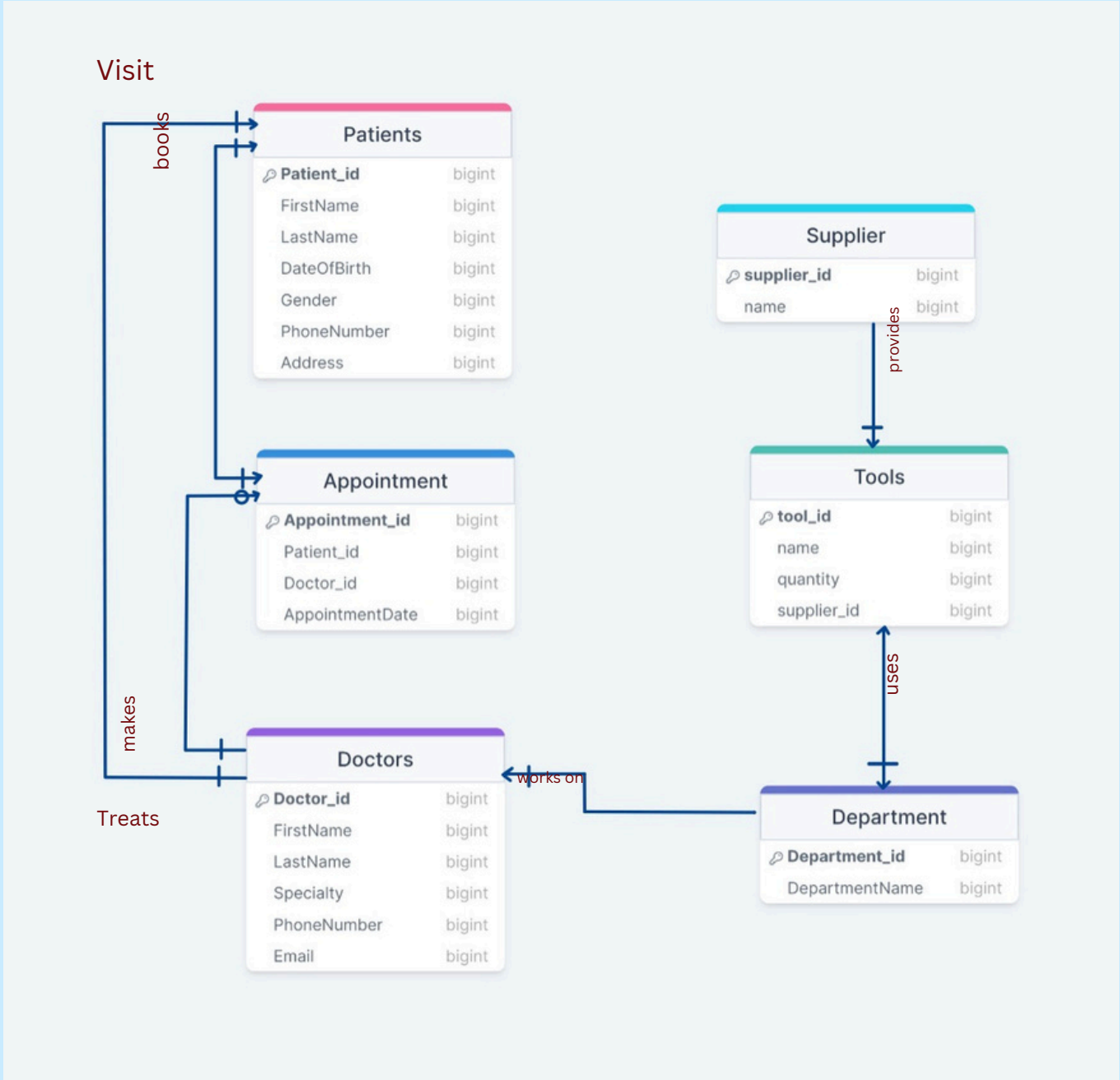
Doctors

- Doctor ID
- First Name
- Last Name
- Specialty
- Phone Number
- Email

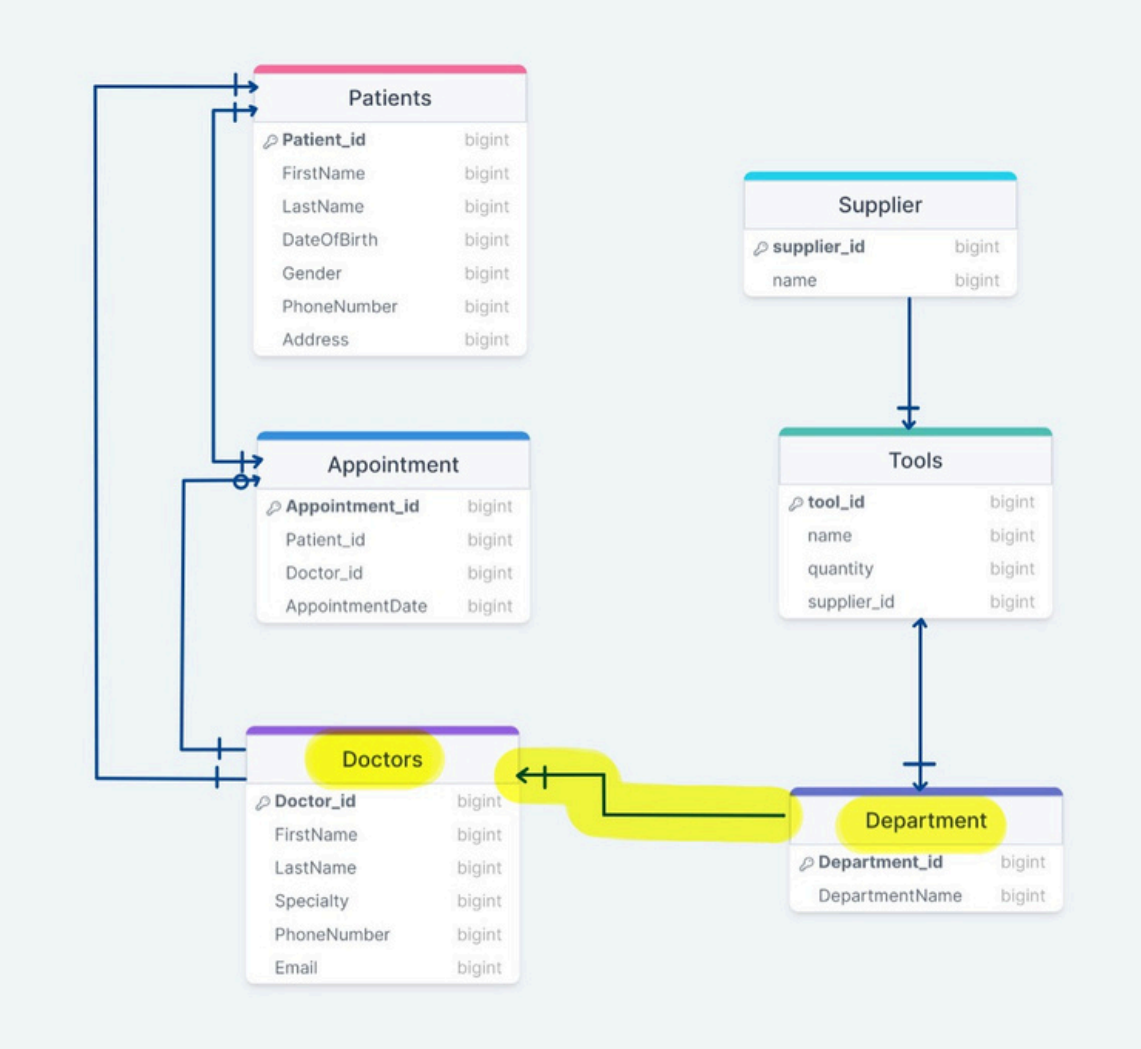
Department

- Department ID
- Department Name

E-R DIAGRAM :


Visit
books
Patients
Patient_id bigint
FirstName bigint
LastName bigint
DateOfBirth bigint
Gender bigint
PhoneNumber bigint
Address bigint
Supplier
supplier_id bigint
name bigint
provides
Appointment
Appointment_id bigint
Patient_id bigint
Doctor_id bigint
AppointmentDate bigint
Tools
tool_id bigint
name bigint
quantity bigint
supplier_id bigint
uses
makes
Doctors
works on
Doctor_id bigint
FirstName bigint
LastName bigint
Specialty bigint
PhoneNumber bigint
Email bigint
Department
Department_id bigint
DepartmentName bigint
Treats

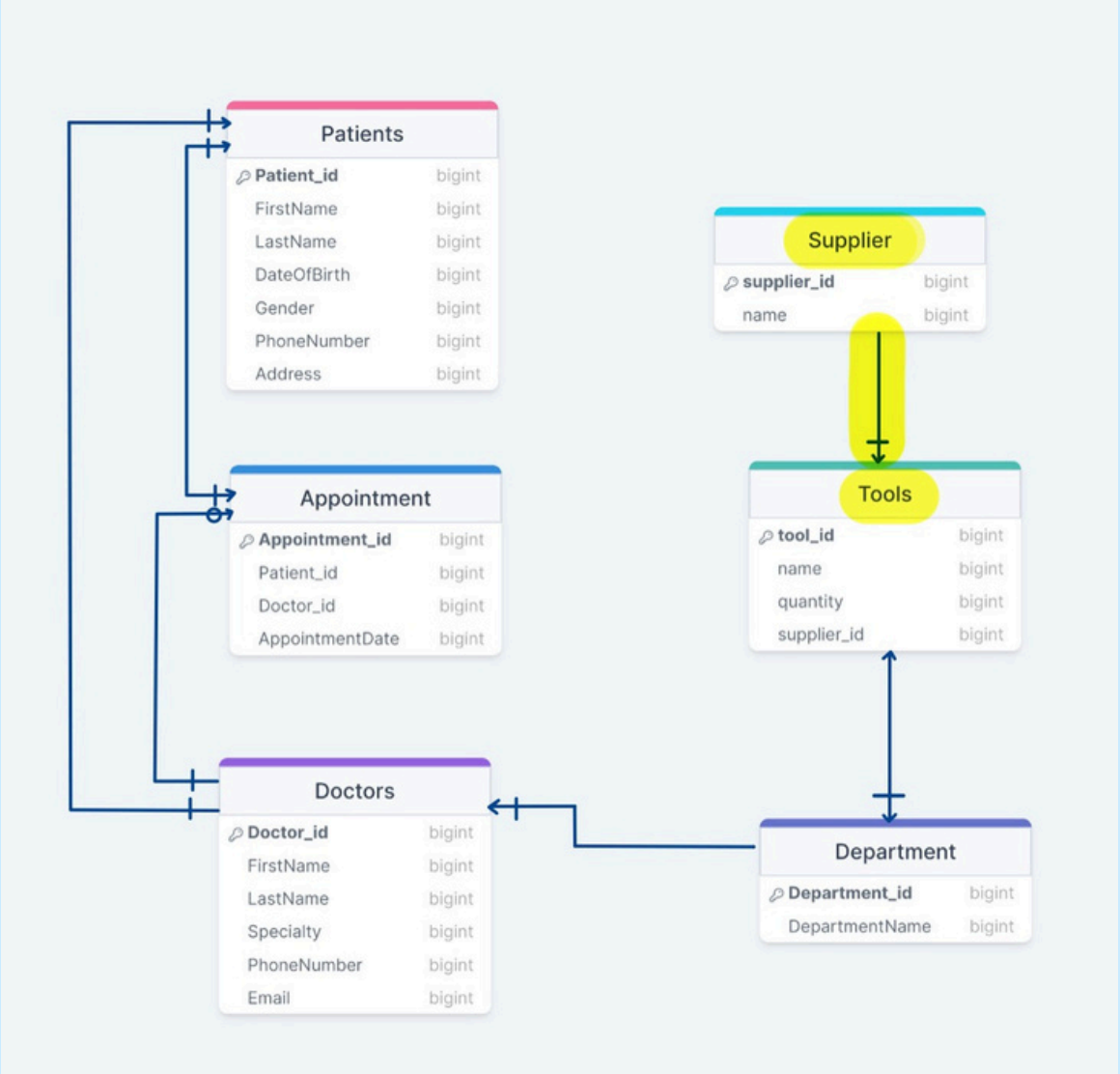
MORE DETAILS ABOUT RELATIONSHIPS :



RELATIONSHIPS:

1. DOCTOR AND DEPARTMENT: EACH DOCTOR WORKS IN EXACTLY ONE DEPARTMENT. (MANDATORY)

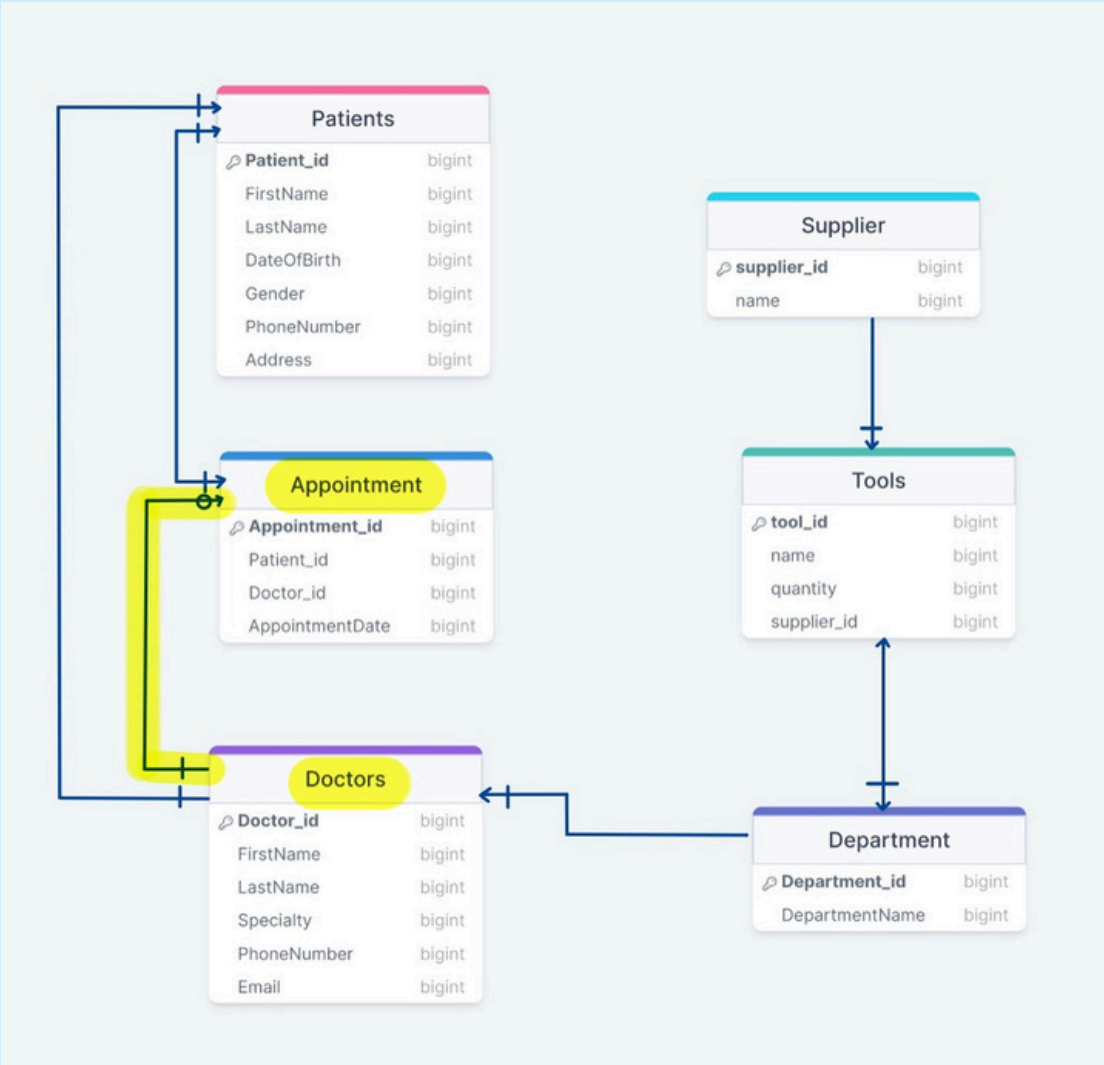
MORE DETAILS ABOUT RELATIONSHIPS :



RELATIONSHIPS:

2.TOOL AND SUPPLIER: EACH TOOL IS SUPPLIED BY EXACTLY ONE SUPPLIER. (MANDATORY)

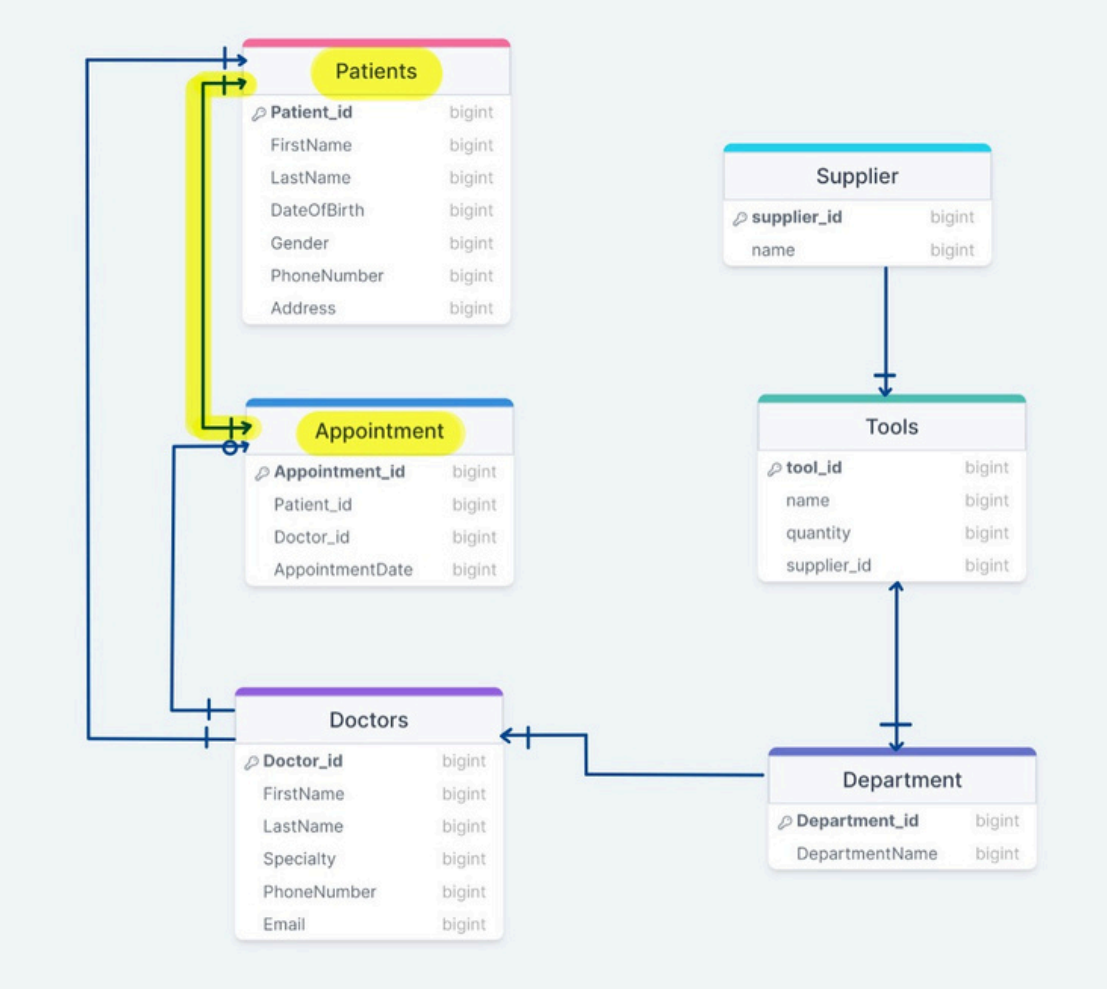
MORE DETAILS ABOUT RELATIONSHIPS :



RELATIONSHIPS:

3.APPOINTMENT AND DOCTOR: EACH APPOINTMENT IS ASSIGNED TO EXACTLY ONE DOCTOR. (MANDATORY)

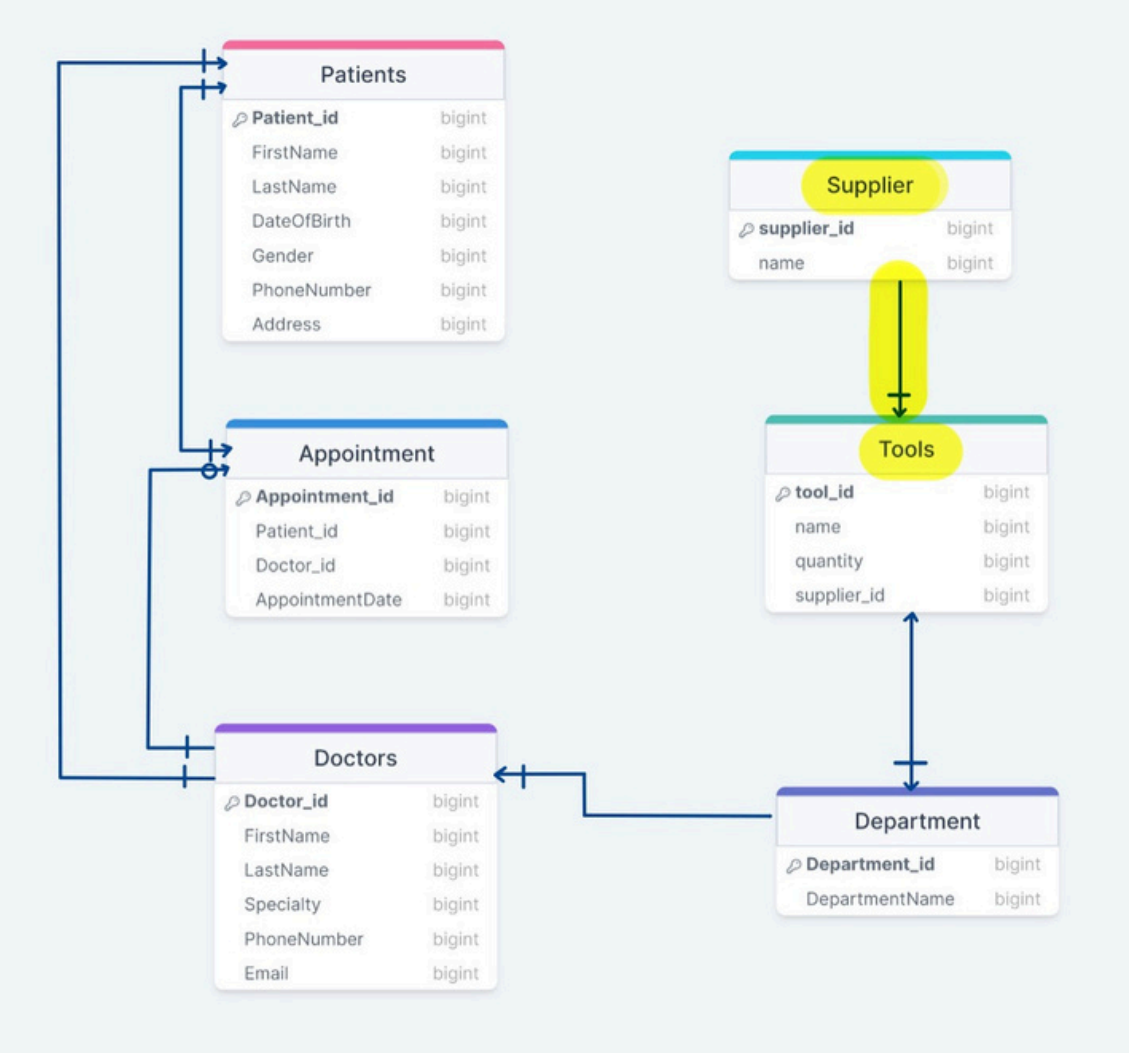
MORE DETAILS ABOUT RELATIONSHIPS :



RELATIONSHIPS:

4.APPOINTMENT AND PATIENT: EACH APPOINTMENT IS MADE BY EXACTLY ONE PATIENT. (MANDATORY)

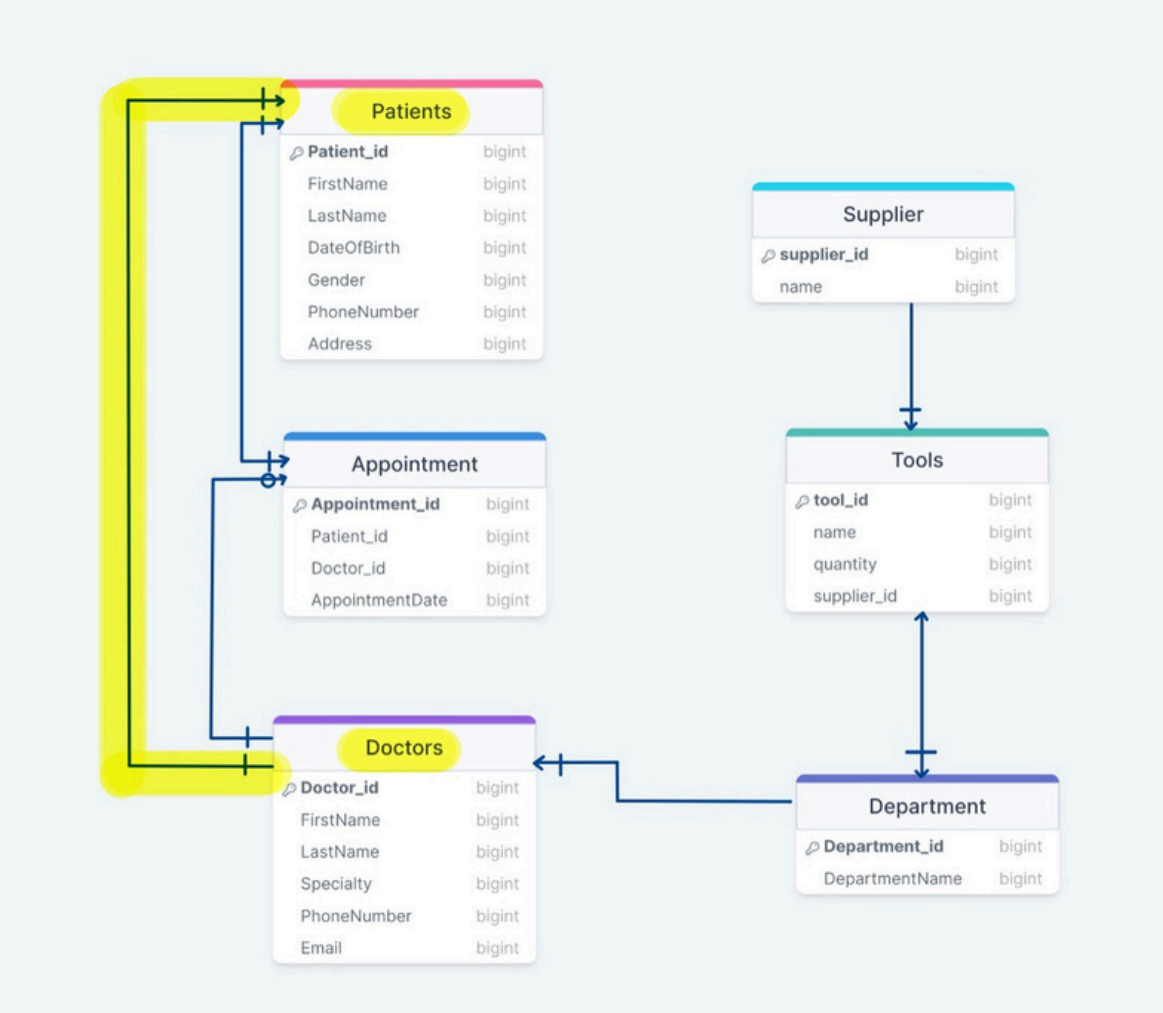
MORE DETAILS ABOUT RELATIONSHIPS :



RELATIONSHIPS:

5.SUPPLIER AND TOOL: EACH SUPPLIER PROVIDES TOOLS, AND EACH TOOL IS PROVIDED BY ONE SUPPLIER. (MANDATORY)

MORE DETAILS ABOUT RELATIONSHIPS :

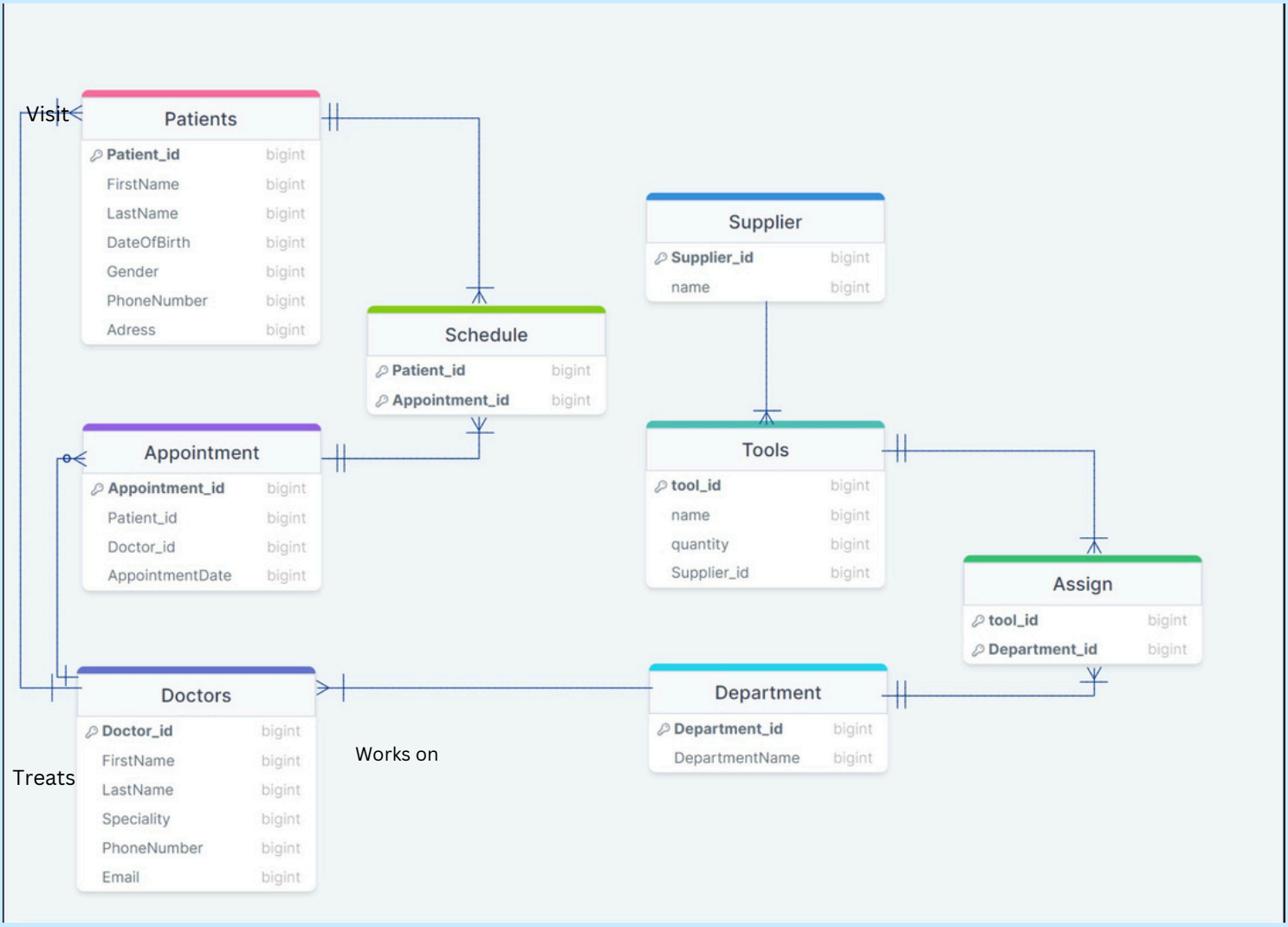


RELATIONSHIPS:

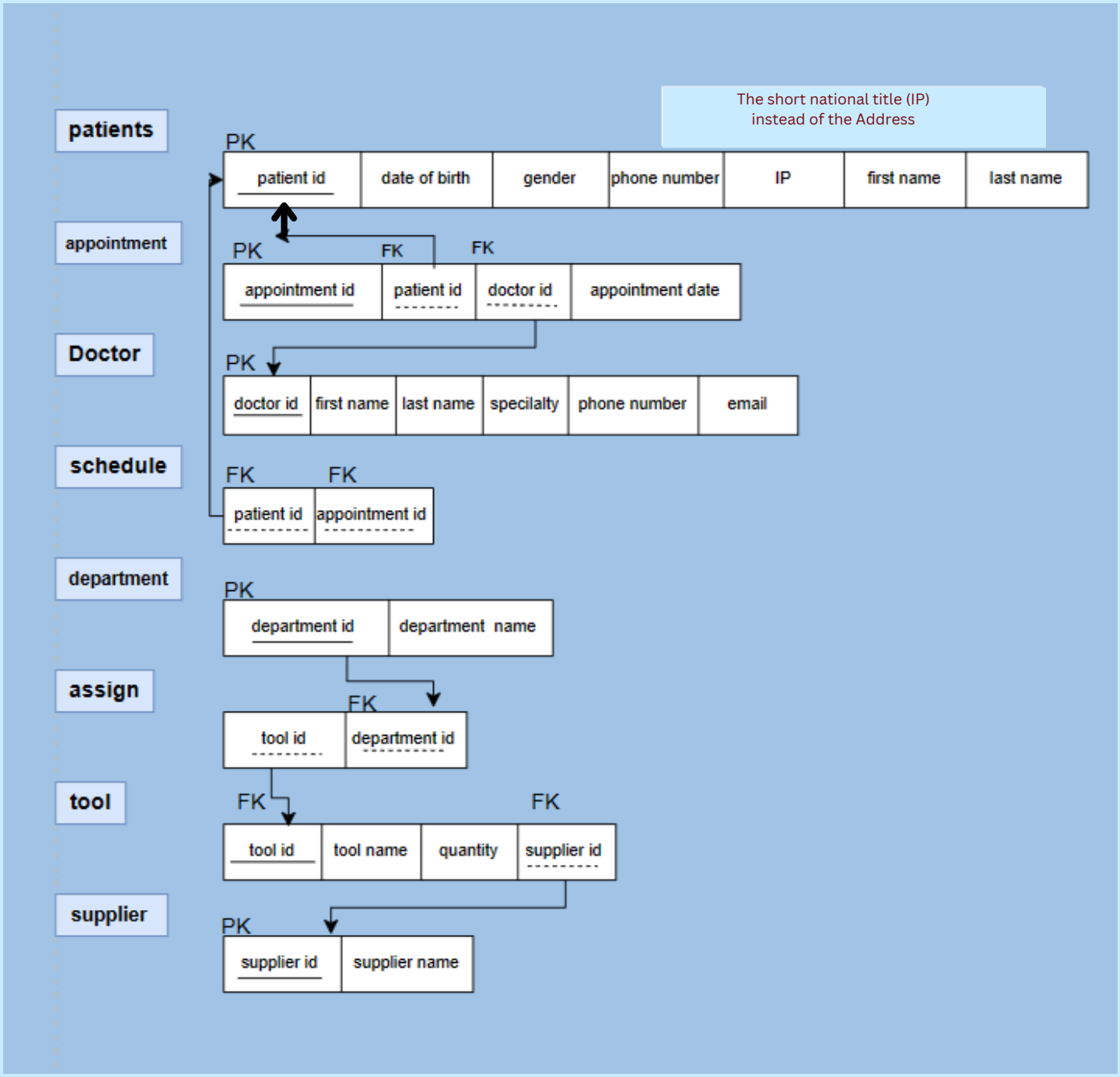
6. DOCTORS AND PATIENTS : EACH DOCTOR CAN HAVE MANY PATIENTS AND EACH PATIENT CAN BE TRATED BY ONLY ONE DOCTOR .

PHASE 2

After Normalization :



RELATIONAL SCHEMA:



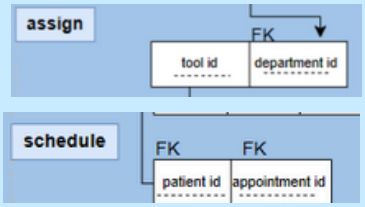
Normalization

1NF:

The condition for first normal form:

- All relation must have a unique primary key.
- Remove any multi value or composite attribute.
- Remove repeating groups.

- The relationship between the **Tool** table and the **Department** Many-to-Many, we dismantled it and created a table **Assign**.
- The relationship between the **Patients** table and the **Appointment** Many-to-Many, we dismantled it and created a table **Schedule**.



2NF:

The condition for second normal form:

- **In** 1NF
- All non key attributes are fully functionally dependent on the primary key in each table.

The schema satisfied to the 2NF rules.

3NF:

The condition foe third normal form:

- Remove transitive dependencies.

All attributes are functionally independent of any other non-primary key attributes.

Functional Dependencies :

FD Patients_id → FirstName, LastName, DateOfBirth, Gender, phoneNumber, (IP)

FD Appointment _id → Patient_id, Doctor_id, AppointmentDate

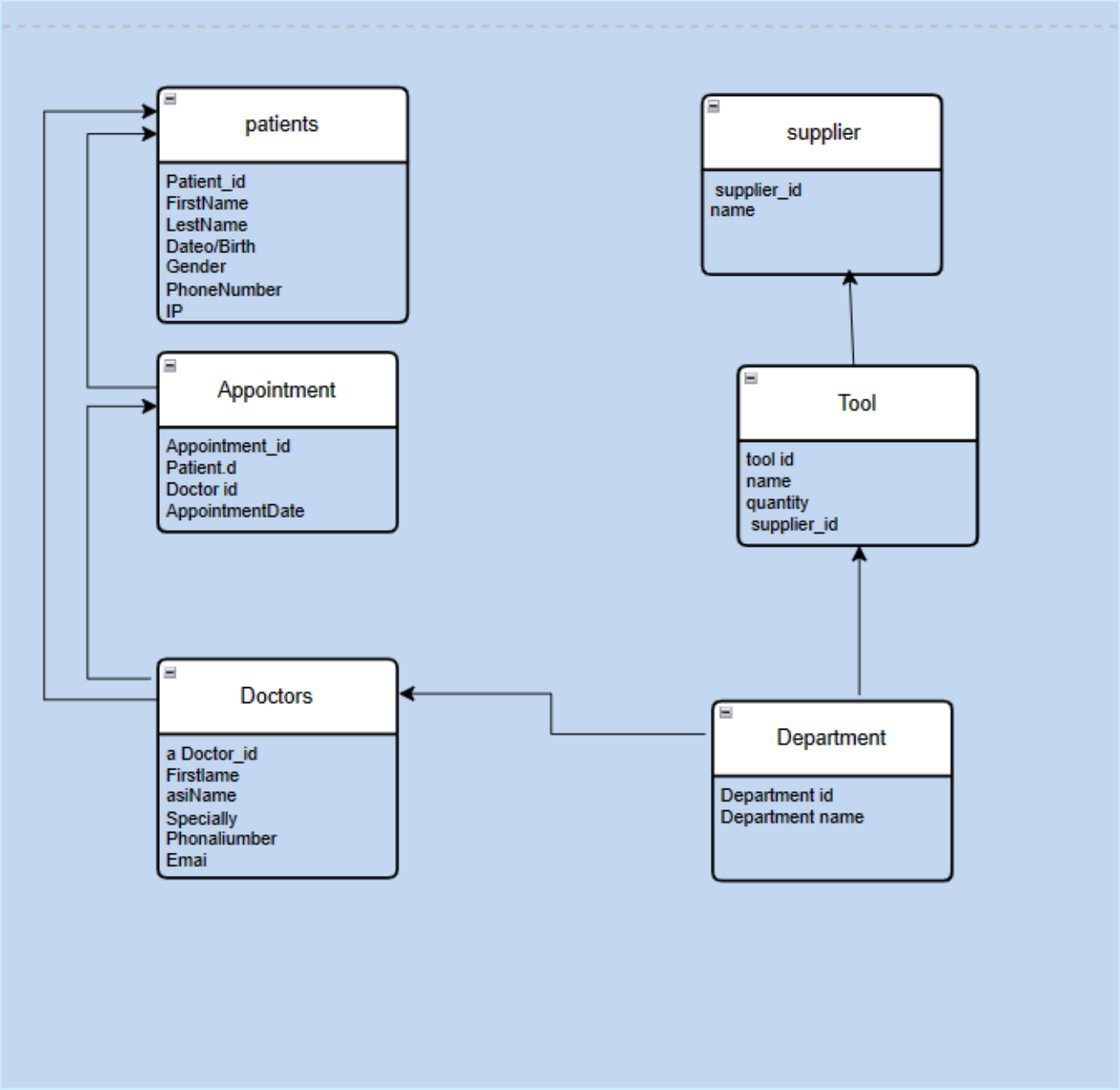
FD Doctors _id → Doctor_id, FirstName, LastName, Speciality, PhoneNumber, Email

FD Department _id → DepartmentName

FD Tool _id → Name, quanttity, Supplier_id

FD Supplier_id → Name

Logical Modeling :



PHASE 3

Normalization Table

Table Name	Attributes	Data Type	Constraints
Patients	Patient_ id	Integer (10)	Primary Key
	First Name	Varchar (100)	
	last Name	Varchar (100)	
	Date Of Birth	Varchar (100)	
	Gender	Varchar (10)	
	Phone Number	Integer (15)	
	address	Varchar (100)	
Appointment	Appointment_ id	Integer (10)	Primary Key
	Patient_ id	Integer (10)	Foreign Key
	Doctor_ id	Integer (10)	Foreign Key
	Appointment Date	Integer (100)	
Doctor	Doctor_ id	Integer (10)	Primary Key
	First Name	Varchar (100)	
	last Name	Varchar (100)	
	specially	Varchar (50)	
	phone Number	Integer (10)	
	email	Varchar (50)	

Supplier	Supplier_ id Name	Integer (10) Varchar (15)	Primary key
Tool	tool_ id Name Quantity Supplier_ id	Integer (10) Varchar (20) Integer (20) Integer (10)	Primary Key Foreign Key
Department	Department _id Department _name	Integer (10) Varchar (30)	Primary Key

TABLES

Patient table :

```
CREATE TABLE Patient (
Patient_id INT PRIMARY KEY,
FirstName VARCHAR (100),
LastName VARCHAR (100) ,
DateOfBirth VARCHAR (100),
Gendre VARCHAR (10),
PhoneNumber varchar (15),
address varchar (100)
)
```

```
Table created.
```

Insert :

```
insert into Patient values(1,'mohammed','alzahrani','22 jan 2001','male','0570732997','jeddah');
insert into Patient values(2, 'waleed','alanzi','04 nov 2000','male','0559232155','riyadh');
insert into Patient values(3,'jana','alotibi','12 sep 2004','female','0563212267','abha');
insert into Patient values(4,'sara ', 'alharbi', '29 may 2004','female','0565439876','dammam');
insert into Patient values(5,'rawan','alharthi', '1 june 2003','female','0543298667','macca');
```

```
1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.
```

Fainal table :

```
select *from patient;
```

PATIENT_ID	FIRSTNAME	LASTNAME	DATEOFBIRTH	GENDRE	PHONENUMBER	ADDRESS
1	mohammed	alzahrani	22 jan 2001	male	0570732997	jeddah
2	waleed	alanzi	04 nov 2000	male	0559232155	riyadh
3	jana	alotibi	12 sep 2004	female	0563212267	abha
4	sara	alharbi	29 may 2004	female	0565439876	dammam
5	rawan	alharthi	1 june 2003	female	0543298667	macca

Download CSV

5 rows selected.

Doctor table :

```
CREATE TABLE doctors (
  doctor_id INT PRIMARY KEY,
  FirstName VARCHAR (100),
  LastName VARCHAR (100) ,
  Specialty VARCHAR (50),
  PhoneNumber varchar (10),
  Email varchar (50)
 )
```

```
Table created.
```

Insert :

```
insert into doctors values(1,'norah','alharbi','dintistry','0570733323','norah4545@gmail.com');
insert into doctors values(2, 'sadeem','alanzi','dermatology','0559767432','sadeem1111@gmail.com');
insert into doctors values(3,'abdullah','alotibi','ear,nose and throat','0563212267','abdullah8888@hotmail.com');
insert into doctors values(4,'fawaz ', 'alzahrani', 'general surgery','0565439988','fawaz222@gmail.com');
insert into doctors values(5,'manar','alharthi', 'pediatrics','0565439876','manar999@hmtmail.com');
```

```
1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.
```

Fainal table :

```
select *from doctors;
```

DOCTOR_ID	FIRSTNAME	LASTNAME	SPECIALTY	PHONENUMBER	EMAIL
1	norah	alharbi	dintistry	570733323	norah4545@gmail.com
2	sadeem	alanzi	dermatology	559767432	sadeem1111@gmail.com
3	abdullah	alotibi	ear,nose and throat	563212267	abdullah8888@hotmail.com
4	fawaz	alzahrani	general surgery	565439988	fawaz222@gmail.com
5	manar	alharthi	pediatrics	565439876	manar999@hmtmail.com

Download CSV

5 rows selected.

Appointment table :

```
CREATE TABLE appointment (
appointment_id INT PRIMARY KEY,
patient_id INT ,
doctor_id INT ,
appointmentDate DATE,
CONSTRAINT FK_patient_id FOREIGN KEY (patient_id) REFERENCES patient (patient_id),
CONSTRAINT FK_doctor_id FOREIGN KEY (doctor_id) REFERENCES doctors (doctor_id)
)
```

```
Table created.
```

Insert :

```
insert into appointment values (1, 1,1,TO_DATE('2024-04-24' , 'YYYY-MM-DD'));
insert into appointment values (2, 2,2,TO_DATE('2024-04-27' , 'YYYY-MM-DD'));
insert into appointment values (3, 3,3,TO_DATE('2024-04-28' , 'YYYY-MM-DD'));
insert into appointment values (4, 4,4,TO_DATE('2024-05-01' , 'YYYY-MM-DD'));
insert into appointment values (5, 5,5,TO_DATE('2024-05-02' , 'YYYY-MM-DD'));
```

```
1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.
```

Fainal table :

```
select*from appointment;
```

APPOINTMENT_ID	PATIENT_ID	DOCTOR_ID	APPOINTMENTDATE
1	1	1	24-APR-24
2	2	2	27-APR-24
3	3	3	28-APR-24
4	4	4	01-MAY-24
5	5	5	02-MAY-24

Download CSV

5 rows selected.

Supplier table :

```
CREATE TABLE supplier (
  supplier_id INT PRIMARY KEY,
  name VARCHAR (15)
 )

```

```
Table created.
```

Insert :

```
 insert into supplier values(1234,'company A');
 insert into supplier values(1235, 'company B');
 insert into supplier values(1236,'company C');
 insert into supplier values(1237,'company D' );
 insert into supplier values(1238,'company E');
```

```
1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.
```

Fainal table :

```
select*from supplier;
```

SUPPLIER_ID	NAME
1234	company A
1235	company B
1236	company C
1237	company D
1238	company E

Download CSV

```
5 rows selected.
```

Tools table :

```
CREATE TABLE Tools (
tool_id INT PRIMARY KEY,
supplier_id INT,
name VARCHAR (20),
quantity VARCHAR (20),
CONSTRAINT FK_supplier_id FOREIGN KEY (supplier_id) REFERENCES supplier (supplier_id)
)
```

Table created.

Insert :

```
insert into Tools values(1,1234,'stethoscope','50');
insert into Tools values(2,1235,'thermometer','40');
insert into Tools values(3,1236,'defibrillator','70');
insert into Tools values(4,1237,'eye char ', '20');
insert into Tools values(5,1238,'stretcher','30');
```

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

Fainal table :

```
select*from Tools;
```

TOOL_ID	SUPPLIER_ID	NAME	QUANTITY
1	1234	stethoscope	50
2	1235	thermometer	40
3	1236	defibrillator	70
4	1237	eye char	20
5	1238	stretcher	30

Download CSV

5 rows selected.

Department table :

```
CREATE TABLE department (
  department_id INT PRIMARY KEY,
  departmentName VARCHAR (30)
  )
```

```
Table created.
```

Insert :

```
insert into department values(1,'pediatrics');
insert into department values(2, 'pediatrics');
insert into department values(3,'emergency');
insert into department values(4,'ophthalmology ');
insert into department values(5,'emergency');
```

```
1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.
```

Fainal table :

```
select*from department;
```

DEPARTMENT_ID	DEPARTMENTNAME
1	pediatrics
2	pediatrics
3	emergency
4	ophthalmology
5	emergency

Download CSV

5 rows selected.

QUEIRISE

1-Retrieve patient's full name, date of birth, and address:

```
SELECT FirstName, LastName, DateOfBirth, Address
FROM Patient;
```

FIRSTNAME	LASTNAME	DATEOFBIRTH	ADDRESS
mohammed	alzahrani	22 jan 2001	jeddah
waleed	alanzi	04 nov 2000	riyadh
jana	alotibi	12 sep 2004	abha
sara	alharbi	29 may 2004	dammam
rawan	alharthi	1 june 2003	macca

Download CSV

5 rows selected.

2-Retrieve the full name of patients who have appointments scheduled with doctors in the pediatrics department:

```
SELECT p.FirstName, p.LastName
  FROM Patient p
  JOIN appointment a ON p.Patient_id = a.patient_id
  JOIN doctors d ON a.doctor_id = d.doctor_id
  JOIN department dept ON d.Specialty = dept.departmentName
  WHERE dept.departmentName = 'pediatrics';
```

FIRSTNAME	LASTNAME
rawan	alharthi
rawan	alharthi

Download CSV

2 rows selected.

QUEIRISE

3-Retrieve the names of doctors who have appointments scheduled on or after May 1, 2024:

```
SELECT d.FirstName, d.LastName
FROM doctors d
JOIN appointment a ON d.doctor_id = a.doctor_id
WHERE a.appointmentDate >= TO_DATE('2024-05-01', 'YYYY-MM-DD');
```

FIRSTNAME	LASTNAME
fawaz	alzahrani
manar	alharthi

Download CSV

2 rows selected.

4-Retrieve the names of suppliers who provide tools with quantities greater than 50:

```
SELECT s.name
FROM supplier s
JOIN Tools t ON s.supplier_id = t.supplier_id
WHERE t.quantity > 50;
```

NAME
company C

Download CSV

QUEIRISE

5-Retrieve the full names of patients along with their appointment dates and the specialty of the doctor they're scheduled with:

```
SELECT p.FirstName, p.LastName, a.appointmentDate, d.Specialty
FROM Patient p
JOIN appointment a ON p.Patient_id = a.patient_id
JOIN doctors d ON a.doctor_id = d.doctor_id;
```

FIRSTNAME	LASTNAME	APPOINTMENTDATE	SPECIALTY
mohammed	alzahrani	24-APR-24	dintistry
waleed	alanzi	27-APR-24	dermatology
jana	alotibi	28-APR-24	ear,nose and throat
sara	alharbi	01-MAY-24	general surgery
rawan	alharthi	02-MAY-24	pediatrics

Download CSV

5 rows selected.

6-Retrieve the names of doctors along with the number of appointments they have:

```
SELECT d.FirstName, d.LastName, COUNT(a.appointment_id) AS AppointmentCount
FROM doctors d
LEFT JOIN appointment a ON d.doctor_id = a.doctor_id
GROUP BY d.FirstName, d.LastName;
```

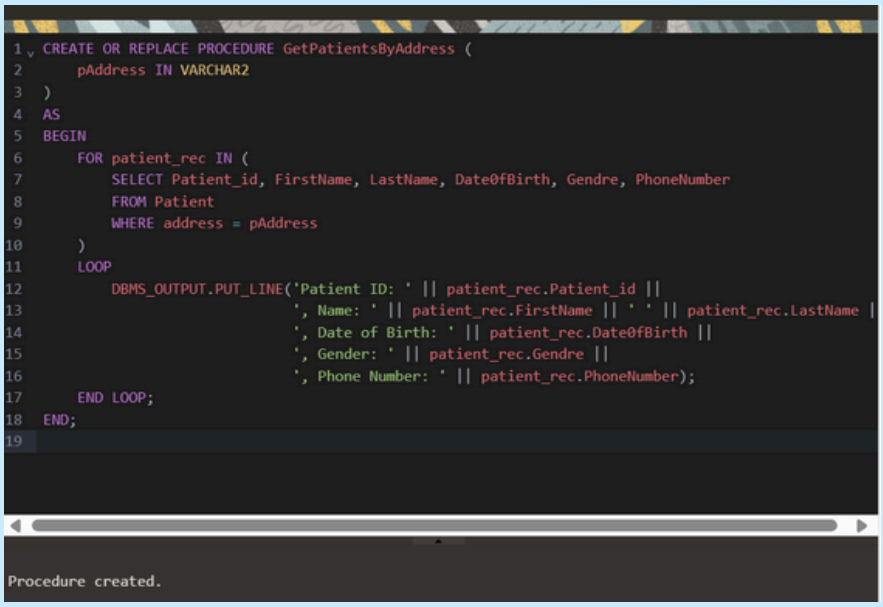
FIRSTNAME	LASTNAME	APPOINTMENTCOUNT
sadeem	alanzi	1
manar	alharthi	1
abdullah	alotibi	1
fawaz	alzahrani	1
norah	alharbi	1

Download CSV

5 rows selected.

The procedure :

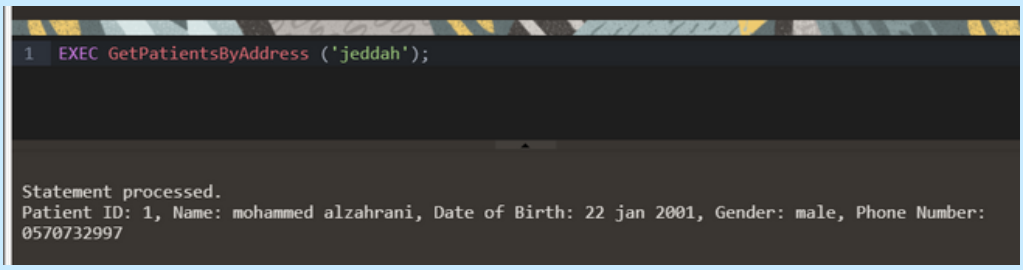
This procedure retrieves information about patients with the specified address and prints out their details, such as ID, name, date of birth, gender, and phone number. You can execute this procedure by passing the desired address as a parameter.



```
CREATE OR REPLACE PROCEDURE GetPatientsByAddress (
    pAddress IN VARCHAR2
)
AS
BEGIN
    FOR patient_rec IN (
        SELECT Patient_id, FirstName, LastName, Date0fBirth, Gendre, PhoneNumber
        FROM Patient
        WHERE address = pAddress
    )
    LOOP
        DBMS_OUTPUT.PUT_LINE('Patient ID: ' || patient_rec.Patient_id ||
                             ', Name: ' || patient_rec.FirstName || ' ' || patient_rec.LastName |
                             ', Date of Birth: ' || patient_rec.Date0fBirth ||
                             ', Gender: ' || patient_rec.Gendre ||
                             ', Phone Number: ' || patient_rec.PhoneNumber);
    END LOOP;
END;

Procedure created.
```

Call procedure :

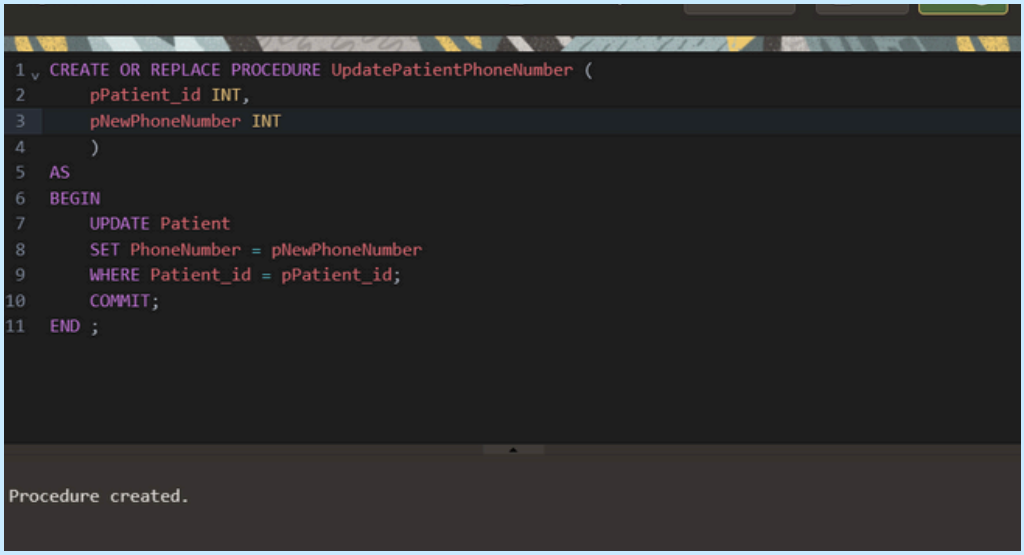


```
EXEC GetPatientsByAddress ('jeddah');

Statement processed.
Patient ID: 1, Name: mohammed alzahrani, Date of Birth: 22 jan 2001, Gender: male, Phone Number:
0570732997
```

The procedure :

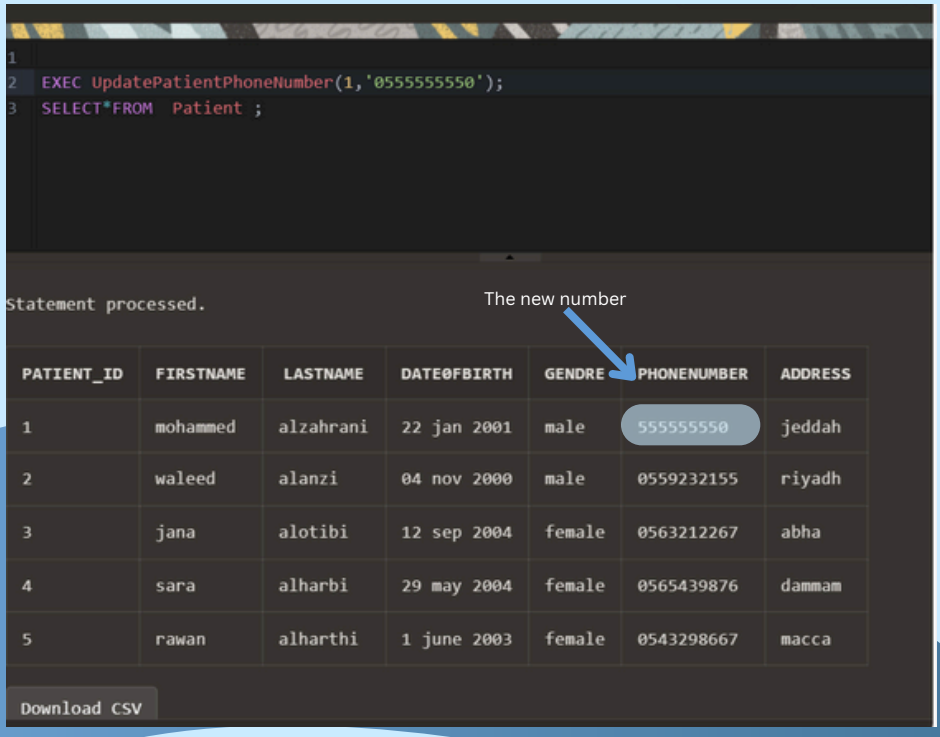
This procedure updet patient number. You can execute this procedure by passing patient id and the new number as a parameter.



```
CREATE OR REPLACE PROCEDURE UpdatePatientPhoneNumber (
    pPatient_id INT,
    pNewPhoneNumber INT
    )
AS
BEGIN
    UPDATE Patient
    SET PhoneNumber = pNewPhoneNumber
    WHERE Patient_id = pPatient_id;
    COMMIT;
END ;
```

```
Procedure created.
```

Call procedure :



```
EXEC UpdatePatientPhoneNumber(1,'0555555550');
SELECT*FROM  Patient ;
```

```
Statement processed.
```

The new number

PATIENT_ID	FIRSTNAME	LASTNAME	DATE0FBIRTH	GENDRE	PHONENUMBER	ADDRESS
1	mohammed	alzahrani	22 jan 2001	male	555555550	jeddah
2	waleed	alanzi	04 nov 2000	male	0559232155	riyadh
3	jana	alotibi	12 sep 2004	female	0563212267	abha
4	sara	alharbi	29 may 2004	female	0565439876	dammam
5	rawan	alharthi	1 june 2003	female	0543298667	macca

Download CSV

DISTRIBUTION OF TASKS

WATEEN	Cover page indicating the title of the project. Full names of the group members (with student ID). ER-Diagram . Logical model and relations. Functional dependencies . Populate tables with 5 rows . Table of content .
JOUD	A narrative description of the problem. Normalization. Normalization table.
WAREEF	Identification of the information needs . Mapping. Design two stored procedures.
WASAN	Initial list of entities (tables) that have been identified. Distribution of tasks. The course number and course section. ER diagram after Normalization . Design and implement 6 queries.