**Fundamentals of Databases (CBIO204)**

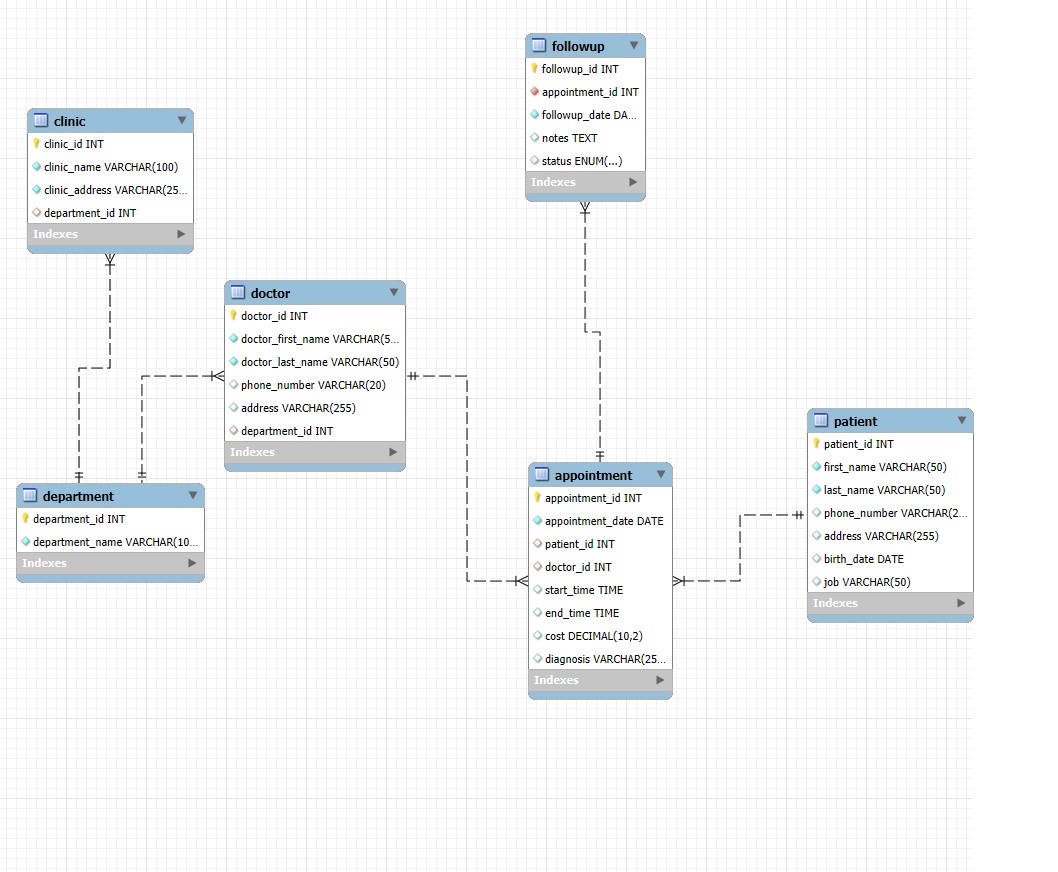
**clinic\_management\_system**

**Project: 03**

**Supervision: Dr Muhammed elsayeh**

|  |  |
| --- | --- |
| Student Name | ID |
| Nada essam saber | 231001261 |
| Jana sherif elsisy | 231002223 |
| Youssef ahmed elshorbagy | 231001220 |

ERD photo



**2. ER Diagram with Cardinality & Participation We’ll have:**

Department ↔ Clinic → 1: N (One department has many clinics; each clinic belongs to one department) Department ↔ Doctor → 1: N (One department has many doctors; each doctor works in one department) Doctor ↔ Appointment → 1: N (One doctor has many appointments; each appointment has one doctor) Patient ↔ Appointment → 1: N (One patient can have many appointments; each appointment is for one patient) Appointment ↔ FollowUp → 1: N (One appointment can have many follow-ups; each follow-up is for one appointment)

**SQL Table Creation**

CREATE TABLE Department ( department\_id INT PRIMARY KEY, department\_name VARCHAR(100) NOT NULL );

CREATE TABLE Clinic ( clinic\_id INT PRIMARY KEY, clinic\_name VARCHAR(100) NOT NULL, clinic\_address VARCHAR(255), department\_id INT, FOREIGN KEY (department\_id) REFERENCES Department(department\_id) );

CREATE TABLE Doctor ( doctor\_id INT PRIMARY KEY, doctor\_name VARCHAR(100) NOT

NULL, doctor\_phone VARCHAR(20), doctor\_address VARCHAR(255), department\_id INT, FOREIGN KEY (department\_id) REFERENCES Department(department\_id) );

CREATE TABLE Patient ( patient\_id INT PRIMARY KEY, first\_name VARCHAR(50), last\_name VARCHAR(50), phone\_number VARCHAR(20), address VARCHAR(255), birth\_date DATE, job VARCHAR(100) );

CREATE TABLE Appointment ( appointment\_id INT PRIMARY KEY, appointment\_date DATE, patient\_id INT, doctor\_id INT, start\_time TIME, end\_time TIME, cost DECIMAL(10,2), status ENUM('scheduled','in progress','postponed'), diagnosis VARCHAR(255), FOREIGN KEY

(patient\_id) REFERENCES Patient(patient\_id), FOREIGN KEY (doctor\_id) REFERENCES Doctor(doctor\_id) );

CREATE TABLE FollowUp ( followup\_id INT PRIMARY KEY, appointment\_id INT, followup\_date DATE, notes TEXT, FOREIGN KEY (appointment\_id) REFERENCES Appointment(appointment\_id) );

**Relational Schema for Medical Clinic Database:**

department ( department\_id INT PRIMARY KEY, department\_name

VARCHAR(100) NOT NULL )

patient ( patient\_id INT PRIMARY KEY, first\_name VARCHAR(50) NOT NULL, last\_name VARCHAR(50) NOT NULL, phone\_number VARCHAR(25) NOT NULL, address VARCHAR(255) NOT NULL, birth\_date DATE NOT NULL, job VARCHAR(50)

)

doctor ( doctor\_id INT PRIMARY KEY, doctor\_first\_name VARCHAR(50) NOT NULL, doctor\_last\_name VARCHAR(50) NOT NULL, phone\_number VARCHAR(20) NOT NULL, address VARCHAR(255) NOT NULL, department\_id INT NOT NULL,

FOREIGN KEY (department\_id) REFERENCES department(department\_id)

)

clinic ( clinic\_id INT PRIMARY KEY, clinic\_name VARCHAR(100) NOT NULL, clinic\_address VARCHAR(255) NOT NULL, department\_id INT NOT NULL,

FOREIGN KEY (department\_id) REFERENCES department(department\_id)

)

appointment (

appointment\_id INT PRIMARY KEY, appointment\_date DATE NOT NULL, patient\_id INT NOT NULL, doctor\_id INT NOT NULL, start\_time TIME NOT NULL, end\_time TIME NOT NULL, cost DECIMAL(10,2) NOT NULL, diagnosis VARCHAR(255),

FOREIGN KEY (patient\_id) REFERENCES patient(patient\_id),

FOREIGN KEY (doctor\_id) REFERENCES doctor(doctor\_id)

)

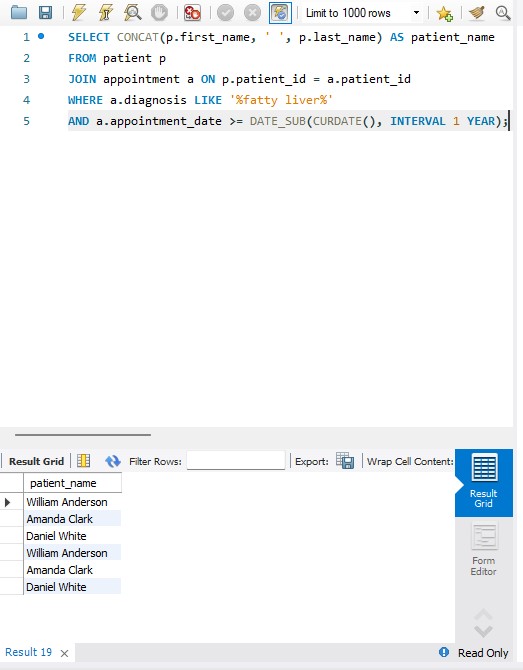
followup ( followup\_id INT PRIMARY KEY, appointment\_id INT NOT NULL, followup\_date DATE NOT NULL, notes TEXT,

status ENUM('scheduled', 'pending', 'completed', 'cancelled') NOT NULL,

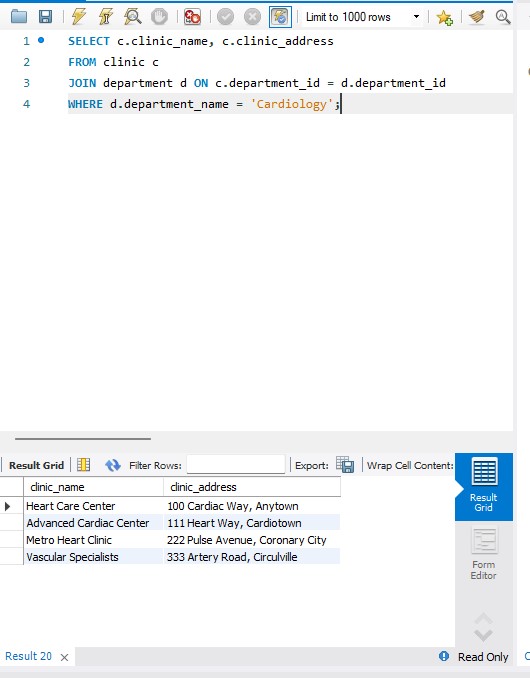
FOREIGN KEY (appointment\_id) REFERENCES appointment(appointment\_id)

)

1• List the name of patients who were diagnosed with fatty liver in the last year.



2• List the addresses of cardiology clinics.



3• List the total money paid by a patient whose ID is 12527 in the last three years.

