

**Project 4: Medical Laboratory System Database Design and Implementation**

**Name : sama saeed moahmed**

**Id :221002117**

**Dr. Mohamed ELSayeh**

**Introduction**

This project aims to provide a comprehensive database system for a medical laboratory. It oversees critical entities such as laboratory employees, patients, medical tests, test outcomes, and test components. The system improves laboratory operations by arranging test supplies, assuring correct recording of findings, and allowing quick access to patient test data. Additionally, a user-friendly desktop application was created using Python and Tkinter, allowing users to interact with the system without the need for sophisticated instructions.

**Entities and Attributes**

* Laboratorian with attributes Laboratorian\_ID (Primary Key), Name, Phone\_Number, and Address.
* Patient with attributes Patient\_ID (Primary Key), Name, Phone\_Number, Address, Birth\_Date, and Job.
* Component with attributes Component\_ID (Primary Key), Name, Available\_Quantity, and Minimum\_Quantity.
* Medical\_Test with attributes Test\_ID (Primary Key), Name, and Price.
* There is a many-to-many relationship between Medical\_Test and Component, represented by the associative entity Test\_Component. It has a composite primary key consisting of Test\_ID and Component\_ID, both of which are foreign keys referencing Medical\_Test and Component respectively.
* The Test\_Result entity stores test results with attributes Result\_ID (Primary Key), Test\_ID (Foreign Key referencing Medical\_Test), Patient\_ID (Foreign Key referencing Patient), Laboratorian\_ID (Foreign Key referencing Laboratorian), Date, and Result\_Description.

**Relational Schema**

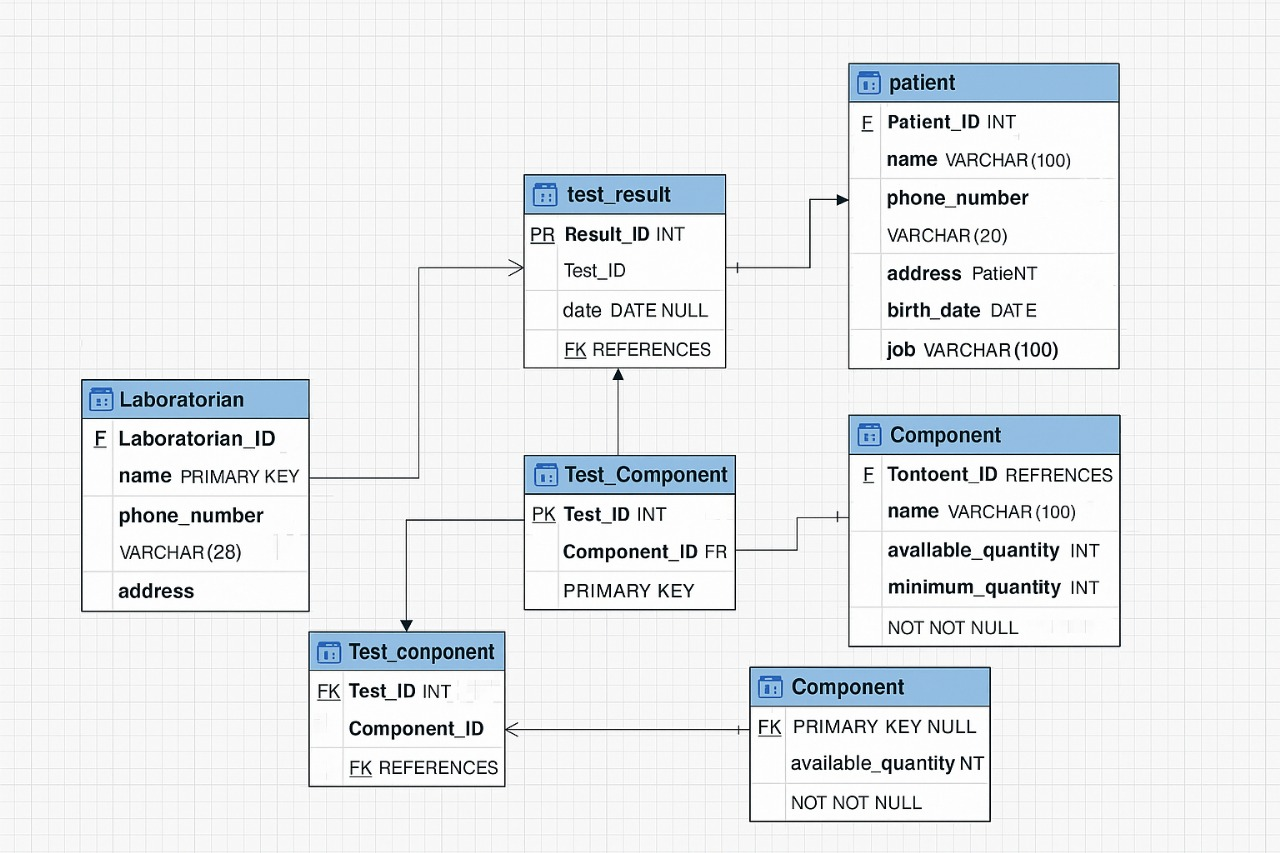
|  |  |  |
| --- | --- | --- |
| **Table** | **Attributes** | **Keys** |
| **Laboratorian** | Laboratorian\_ID, Name, Phone\_Number, Address | PK: Laboratorian\_ID |
| **Patient** | Patient\_ID, Name, Phone\_Number, Address, Birth\_Date, Job | PK: Patient\_ID |
| **Component** | Component\_ID, Name, Available Quantity, Minimum\_Quantity | PK: Component\_ID |
| **Medical\_Test** | Test\_ID, Name, Price | PK: Test\_ID |
| **Test\_Component** | Test\_ID, Component\_ID | PK: (Test\_ID, Component\_ID) FK: to Medical\_Test and Component |
| **Test\_Result** | Result\_ID, Test\_ID, Date, Patient\_ID, Laboratorian\_ID, Result | PK: Result\_ID FK: to related tables |

**Test\_Result Table**

* Result\_ID (INT) — Primary Key
* Test\_ID (INT) — Foreign Key: references Medical\_Test(Test\_ID)
* Patient\_ID (INT) — Foreign Key: references Patient (Patient\_ID)
* Laboratorian\_ID (INT) — Foreign Key: references Laboratorian (Laboratorian\_ID) Date (DATE)
* **Explanation:**

The schema structures the data into tables with well-defined relationships. Each table includes a primary key to uniquely identify records and foreign keys to connect related tables. The Test\_Component table employs a composite key to manage the many-to-many relationship between tests and components. This design ensures accurate storage, updating, and retrieval of lab data while maintaining clear and consistent relationships.

**ER Diagram**

The ER diagram shows relationships in the system.

* A one-to-many relationship between Laboratorian and Test Result (a laboratorian can perform many test results).
* A one-to-many relationship between Patient and Test Result (a patient can have multiple test results).
* A many-to-many relationship between Medical Test and Component, managed through the Test Component associative entity.
* Test Result requires total participation from Patient and Laboratorian. Components have partial participation with Medical Test.

**1)SQL Statements for Table Creation**

create database laborATION;

CREATE TABLE Laboratorian (

Laboratorian\_ID INT PRIMARY KEY,

name VARCHAR (100) NOT NULL,

Phone\_number VARCHAR (20),

address VARCHAR (255)

);

CREATE TABLE Patient (

Patient\_ID INT PRIMARY KEY,

name VARCHAR(100) NOT NULL,

phone\_number VARCHAR(20),

address VARCHAR(255),

birth\_date DATE,

job VARCHAR (100)

);

CREATE TABLE Component (

Component\_ID INT PRIMARY KEY,

name VARCHAR(100) NOT NULL,

available\_quantity INT NOT NULL,

minimum\_quantity INT NOT NULL

);

CREATE TABLE Medical\_Test (

Test\_ID INT PRIMARY KEY,

name VARCHAR(100) NOT NULL,

price DECIMAL(10,2) NOT NULL

);

CREATE TABLE Test\_Component (

Test\_ID INT,

Component\_ID INT,

PRIMARY KEY (Test\_ID, Component\_ID),

FOREIGN KEY (Test\_ID) REFERENCES Medical\_Test(Test\_ID),

FOREIGN KEY (Component\_ID) REFERENCES Component(Component\_ID)

);

**2)CREATE TABLE Test\_Result (**

Result\_ID INT PRIMARY KEY,

Test\_ID INT,

date DATE NOT NULL,

Patient\_ID INT,

Laboratorian\_ID INT,

result VARCHAR (255), FOREIGN KEY (Test\_ID) REFERENCES Medical\_Test(Test\_ID), FOREIGN KEY (Patient\_ID) REFERENCES Patient (Patient\_ID),

FOREIGN KEY (Laboratorian\_ID) REFERENCES Laboratorian (Laboratorian\_ID)

);

**Explanation:**

The SQL statements create tables for the medical lab system, with each table having a primary key to uniquely identify records and maintain organization. The Test\_Component table uses a composite primary key consisting of Test\_ID and Component\_ID to represent the many-to-many relationship between medical tests and components. Foreign keys are implemented to connect related tables and ensure referential integrity. Additionally, important fields are defined as NOT NULL to guarantee that essential data is always provided.

**3).Sample Data Insertion**

INSERT INTO Laboratorian VALUES

(1, 'Yousef Kamal', '0155555555', 'Cairo'),

(2, 'Lina Farouk', '0123456789', 'Alexandria'),

(3, 'Samir Nabil', '0112345678', 'Giza'),

(4, 'Dina Youssef', '0109876543', 'Cairo'),

(5, 'Hassan Adel', '0133333333', 'Tanta'),

(6, 'Mona Sherif', '0144444444', 'Mansoura'),

(7, 'Omar Khaled', '0156666666', 'Cairo'),

(8, 'Rania Ibrahim', '0167777777', 'Alexandria'),

(9, 'Tariq Sami', '0178888888', 'Cairo'),

(10, 'Nahla Samir', '0189999999', 'Giza');

INSERT INTO Patient VALUES

(1001, 'Sara Ibrahim', '0191111111', 'Cairo', '1983-02-14', 'Architect'),

(1002, 'Ahmed Mostafa', '0112222222', 'Alexandria', '1989-07-21', 'Engineer'),

(1003, 'Mona Khalil', '0123333333', 'Giza', '1979-12-01', 'Teacher'),

(1004, 'Hany Nasser', '0134444444', 'Cairo', '1985-04-25', 'Pharmacist'),

(1005, 'Laila Farid', '0145555555', 'Tanta', '1993-11-30', 'Nurse'),

(1006, 'Khaled Salah', '0156666666', 'Mansoura', '1982-09-10', 'Lawyer'),

(1007, 'Nada Samir', '0167777777', 'Cairo', '1986-03-05', 'Accountant'),

(1008, 'Tamer Maher', '0178888888', 'Alexandria', '1994-06-18', 'Designer'),

(1009, 'Amira Adel', '0189999999', 'Giza', '1977-08-22', 'Doctor'),

(1010, 'Omar Khaled', '0190000000', 'Cairo', '1991-01-30', 'Student');

INSERT INTO Component VALUES

(1, 'Serum Separator Tube', 60, 15),

(2, 'Blood Glucose Reagent', 25, 20),

(3, 'Hemoglobin A1c Reagent', 50, 30),

(4, 'Glass Microscope Slides', 55, 20),

(5, 'Sterile Urine Cups', 40, 20),

(6, 'Chemical Reagent X', 18, 12),

(7, 'Chemical Reagent Y', 10, 15),

(8, 'Plastic Test Tubes', 45, 20),

(9, 'Sterile Alcohol Swabs', 75, 40),

(10, 'Disposable Gloves', 110, 60);

INSERT INTO Medical\_Test VALUES

(101, 'Complete Blood Count', 180.00),

(102, 'Fasting Blood Sugar', 130.00),

(103, 'Urine Routine Test', 110.00),

(104, 'Liver Enzymes Test', 210.00),

(105, 'Kidney Profile', 195.00),

(106, 'Cholesterol Test', 170.00),

(107, 'Thyroid Panel', 175.00),

(108, 'Vitamin B12 Test', 165.00),

(109, 'COVID-19 Antigen', 320.00),

(110, 'Electrolytes Test', 150.00);

INSERT INTO Test\_Component VALUES

(101, 1),

(101, 3),

(102, 2),

(103, 5),

(104, 6),

(105, 7),

(106, 8),

(107, 9),

(108, 10),

(109, 4),

(110, 3);

INSERT INTO Test\_Result VALUES

(1, 101, '2025-01-15', 1001, 2, 'CBC within normal range'),

(2, 102, '2025-02-10', 1002, 3, 'Slightly elevated blood sugar'),

(3, 103, '2025-03-05', 1003, 4, 'Urine test normal'),

(4, 104, '2025-01-25', 1004, 5, 'Mildly elevated liver enzymes'),

(5, 105, '2025-02-15', 1005, 6, 'Kidney function normal'),

(6, 106, '2025-03-20', 1006, 7, 'Cholesterol high'),

(7, 107, '2025-01-30', 1007, 8, 'Normal thyroid levels'),

(8, 108, '2025-02-28', 1008, 9, 'Vitamin B12 deficiency'),

(9, 109, '2025-03-15', 1009, 10, 'COVID-19 antigen negative'),

(10, 110, '2025-01-10', 1010, 1, 'Electrolyte levels normal');

**Explanation:**

These SQL insertion commands populate each table with realistic sample data, enabling effective testing of the database. Including at least ten records per table provides sufficient data diversity to thoroughly evaluate queries and system functionality.

**4).Example Queries**

SELECT DISTINCT p.name

FROM Patient p

JOIN Test\_Result tr ON p.Patient\_ID = tr.Patient\_ID

JOIN Medical\_Test mt ON tr.Test\_ID = mt.Test\_ID

WHERE mt.name = 'Complete Blood Count'

AND tr.date >= DATE\_SUB(CURDATE(), INTERVAL 1 YEAR);

SELECT name, available\_quantity, minimum\_quantity

FROM Component

WHERE available\_quantity < minimum\_quantity;

SELECT p.name, SUM(mt.price) AS total\_paid

FROM Patient p

JOIN Test\_Result tr ON p.Patient\_ID = tr.Patient\_ID

JOIN Medical\_Test mt ON tr.Test\_ID = mt.Test\_ID

WHERE p.Patient\_ID = 1001

AND tr.date >= DATE\_SUB(CURDATE(), INTERVAL 3 YEAR)

GROUP BY p.Patient\_ID;

SELECT p.name, SUM(mt.price) AS total\_paid

FROM Patient p

JOIN Test\_Result tr ON p.Patient\_ID = tr.Patient\_ID

JOIN Medical\_Test mt ON tr.Test\_ID = mt.Test\_ID

WHERE p.Patient\_ID = 12527

AND tr.date >= DATE\_SUB(CURDATE(), INTERVAL 3 YEAR)

GROUP BY p.Patient\_ID;

**Conclusion**

In conclusion, designing a well-structured database with clear relationships, proper keys, and realistic sample data is essential for building a reliable and efficient medical lab system. This foundation ensures data integrity, supports accurate testing, and enables smooth operation of the system’s functionalities.