**Soha yehia 231001231**

**Omar Hisham 231001872**

**Nour said 231001824**

**Haneen mohamed 231001726**

# Medical Laboratory System – Database Project Report

## 1. Project Overview

This project focuses on building a database system for a medical laboratory using MySQL.  
The system is designed to manage laboratorians, patients, medical tests, test components, and test results.  
It includes a conceptual design (ERD), UML class diagram, and all SQL statements used for implementation and querying.

## 2. Conceptual Design (ER Diagram)

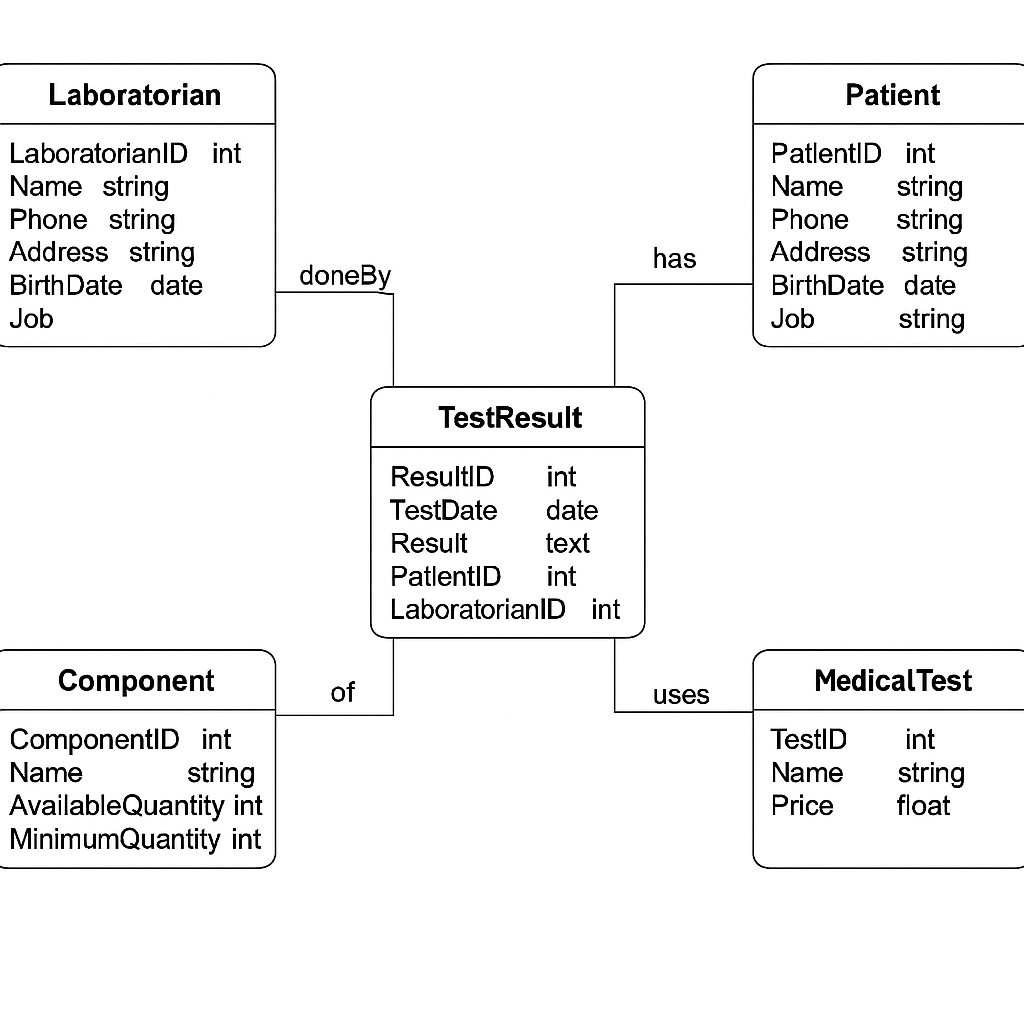
The ER diagram identifies six main entities: Patient, Laboratorian, MedicalTest, Component,  
TestComponent (relationship table), and TestResult. Relationships include:  
- A Patient can have many TestResults.  
- A Laboratorian can be responsible for many TestResults.  
- A MedicalTest can be associated with many Components and vice versa (many-to-many via TestComponent).

Note: ERD was generated using MySQL Workbench Reverse Engineer feature.

## 3. UML Class Diagram

The UML Class Diagram represents the structure of the system from an object-oriented perspective.  
Each class represents a table in the database, with attributes corresponding to table columns and associations representing relationships.

Note: UML diagram included as image or created using diagramming tool such as draw.io



.

## 4. Database Schema (SQL Code)

CREATE DATABASE IF NOT EXISTS MedicalLabDB;  
USE MedicalLabDB;  
  
CREATE TABLE IF NOT EXISTS Laboratorian (  
 LaboratorianID INT PRIMARY KEY AUTO\_INCREMENT,  
 Name VARCHAR(100) NOT NULL,  
 Phone VARCHAR(20),  
 Address VARCHAR(200)  
);  
  
CREATE TABLE IF NOT EXISTS Patient (  
 PatientID INT PRIMARY KEY AUTO\_INCREMENT,  
 Name VARCHAR(100) NOT NULL,  
 Phone VARCHAR(20),  
 Address VARCHAR(200),  
 BirthDate DATE,  
 Job VARCHAR(100)  
);  
  
CREATE TABLE IF NOT EXISTS Component (  
 ComponentID INT PRIMARY KEY AUTO\_INCREMENT,  
 Name VARCHAR(100) NOT NULL,  
 AvailableQuantity INT NOT NULL,  
 MinimumQuantity INT NOT NULL  
);  
  
CREATE TABLE IF NOT EXISTS MedicalTest (  
 TestID INT PRIMARY KEY AUTO\_INCREMENT,  
 Name VARCHAR(100) NOT NULL,  
 Price FLOAT NOT NULL  
);  
  
CREATE TABLE IF NOT EXISTS TestComponent (  
 TestID INT,  
 ComponentID INT,  
 PRIMARY KEY (TestID, ComponentID),  
 FOREIGN KEY (TestID) REFERENCES MedicalTest(TestID) ON DELETE CASCADE ON UPDATE CASCADE,  
 FOREIGN KEY (ComponentID) REFERENCES Component(ComponentID) ON DELETE CASCADE ON UPDATE CASCADE  
);  
  
CREATE TABLE IF NOT EXISTS TestResult (  
 ResultID INT PRIMARY KEY AUTO\_INCREMENT,  
 TestID INT,  
 PatientID INT,  
 LaboratorianID INT,  
 TestDate DATE NOT NULL,  
 Result TEXT,  
 FOREIGN KEY (TestID) REFERENCES MedicalTest(TestID) ON DELETE CASCADE ON UPDATE CASCADE,  
 FOREIGN KEY (PatientID) REFERENCES Patient(PatientID) ON DELETE CASCADE ON UPDATE CASCADE,  
 FOREIGN KEY (LaboratorianID) REFERENCES Laboratorian(LaboratorianID) ON DELETE CASCADE ON UPDATE CASCADE  
);

## 5. Inserted Sample Data

INSERT INTO Laboratorian (Name, Phone, Address) VALUES   
('Dr. Ayman', '0123456789', 'Cairo'),  
('Dr. Sara', '0111111111', 'Alexandria');  
  
INSERT INTO Patient (Name, Phone, Address, BirthDate, Job) VALUES   
('Sohaila', '0100000000', 'Giza', '2001-03-10', 'Engineer'),  
('Ahmed Tarek', '0122222222', 'Tanta', '1995-06-15', 'Teacher');  
  
INSERT INTO Component (Name, AvailableQuantity, MinimumQuantity) VALUES   
('Hemoglobin', 50, 20),  
('WBC', 5, 10),  
('Glucose', 30, 25);  
  
INSERT INTO MedicalTest (Name, Price) VALUES   
('CBC', 200),  
('Blood Sugar', 150);  
  
INSERT INTO TestComponent (TestID, ComponentID) VALUES   
(1, 1),  
(1, 2),  
(2, 3);  
  
INSERT INTO TestResult (TestID, PatientID, LaboratorianID, TestDate, Result) VALUES   
(1, 1, 1, '2024-06-01', 'Normal'),  
(2, 1, 2, '2023-05-15', 'High Sugar'),  
(1, 2, 1, '2023-10-20', 'Low Hemoglobin');

## 6. Queries Executed

-- Patients who performed CBC test in the last year  
SELECT DISTINCT P.Name AS PatientName  
FROM Patient P  
JOIN TestResult R ON P.PatientID = R.PatientID  
JOIN MedicalTest T ON R.TestID = T.TestID  
WHERE T.Name = 'CBC' AND R.TestDate >= DATE\_SUB(CURDATE(), INTERVAL 1 YEAR);  
  
-- Components with quantity less than minimum  
SELECT Name AS ComponentName  
FROM Component  
WHERE AvailableQuantity < MinimumQuantity;  
  
-- Total money paid by patient with ID = 1 in last 3 years  
SELECT SUM(T.Price) AS TotalPaid  
FROM TestResult R  
JOIN MedicalTest T ON R.TestID = T.TestID  
WHERE R.PatientID = 1 AND R.TestDate >= DATE\_SUB(CURDATE(), INTERVAL 3 YEAR);

## 7. Execution Result

All SQL statements executed successfully in MySQL Workbench.

Example result: Patient with ID = 1 paid a total of 360 in the last 3 years.

## 8. Diagram Note

The ER and UML diagrams can be created using MySQL Workbench (Reverse Engineer) and diagramming tools like draw.io.