# MRZ

## PHARMACY DATABASE SYSTEM
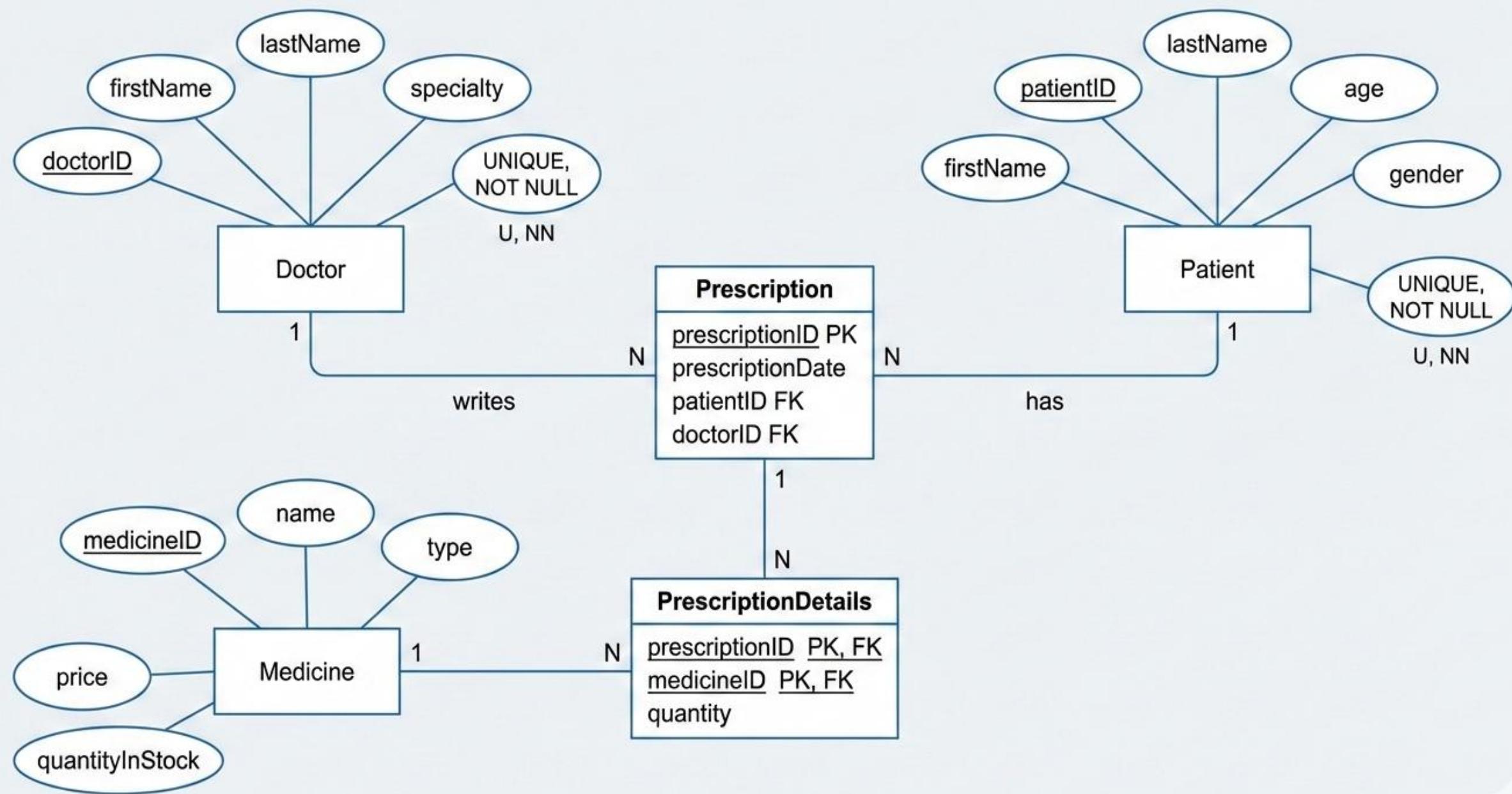
- MOHAMMAD
- ZAID
- REEM
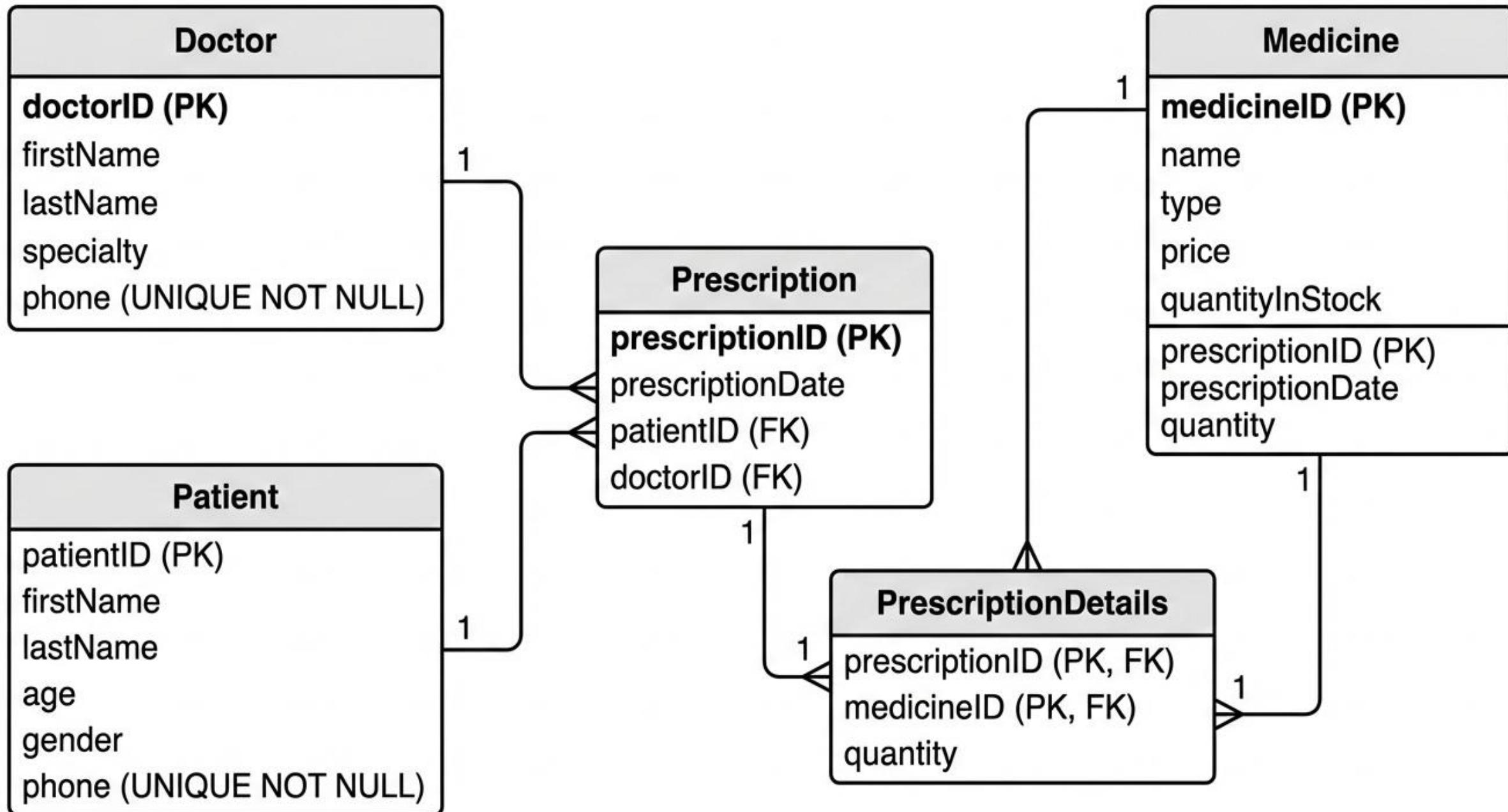
# PROBLEM & OBJECTIVE

- Need system to manage pharmacy data

- Manage patients, doctors, medicines, prescriptions

- Ensure data integrity

# USERS & FUNCTIONS
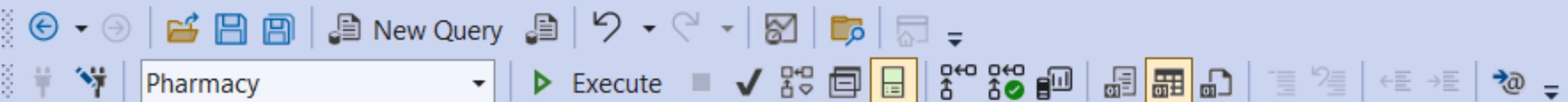
- **Users**

- Pharmacist

- Doctor

- Manager

- **Functions**

- Patient, Doctor, Medicine, Prescription Management

**Doctor**
- **doctorID (PK)**
- firstName
- lastName
- specialty
- phone (UNIQUE NOT NULL)

**Patient**
- patientID (PK)
- firstName
- lastName
- age
- gender
- phone (UNIQUE NOT NULL)

**Prescription**
- **prescriptionID (PK)**
- prescriptionDate
- patientID (FK)
- doctorID (FK)

**Medicine**
- **medicineID (PK)**
- name
- type
- price
- quantityInStock
- prescriptionID (PK)
- prescriptionDate
- quantity

**PrescriptionDetails**
- prescriptionID (PK, FK)
- medicineID (PK, FK)
- quantity

Pharmacy   ▾   ▷ Execute ■ ✓

```sql
1   CREATE TABLE Doctor(
2       doctorID INT NOT NULL PRIMARY KEY,
3       firstName VARCHAR(20) NOT NULL,
4       lastName  VARCHAR(20) NOT NULL,
5       specialty VARCHAR(15) NOT NULL,
6       phone     VARCHAR(10) NOT NULL UNIQUE
7   );
8
```

# KEYS & CONSTRAINTS

**Doctor**

doctorID **INT PK NOT NULL**

firstName **VARCHAR(20) NN**, lastName **VARCHAR(20) NN**, specialty **VARCHAR(15) NN**

phone **VARCHAR(10) NN UNIQUE**

**Patient**

patientID **INT PK NOT NULL**

firstName **VARCHAR(20) NN**, lastName **VARCHAR(20) NN**

age **INT NN CHECK (0–120)**

gender **VARCHAR(6) NN CHECK ('Male','Female')**

phone **VARCHAR(10) NN UNIQUE**

**Medicine**

medicineID **INT PK NOT NULL**

name **VARCHAR(20) NN**, type **VARCHAR(20) NN**

price **MONEY NN CHECK (>0)**

quantityInStock **INT NN CHECK (≥0)**

**Prescription**

prescriptionID **INT PK NOT NULL**

prescriptionDate **DATETIME NN**

patientID **INT NN FK → Patient(patientID)** *(ON UPDATE CASCADE, ON DELETE CASCADE)*

doctorID **INT NN FK → Doctor(doctorID)** *(ON UPDATE CASCADE, ON DELETE NO ACTION)*

**PrescriptionDetails**

(prescriptionID **INT**, medicineID **INT**) **Composite PK**

prescriptionID **FK → Prescription(prescriptionID)**

medicineID **FK → Medicine(medicineID)** , quantity **INT NN CHECK (>0)**

# REFERENTIAL INTEGRITY

1. Prescription.patientID → Patient.patientID

2. Prescription.doctorID → Doctor.doctorID

3. PrescriptionDetails.prescriptionID → Prescription

4. PrescriptionDetails.medicineID → Medicine

5. "Foreign keys ensure data consistency between related tables