

### **Question 1:**

Create a database named SchoolDB and design a table named Students with the following columns:

StudentID (Primary Key, Auto-increment)

FirstName

LastName

Age (Must be greater than or equal to 18)

Gender (Must be 'M' or 'F')

Insert data for at least three students.

Display:

Show the structure of the Students table.

Note:

Ensure that the Age column has a check constraint for values greater than or equal to 18

### **-- Question 2:**

-- Create a database named LibraryDB and design two tables: Books and Authors.

-- The Books table should have the following columns:

-- - BookID (Primary Key, Auto-increment)

-- - Title

-- - AuthorID (Foreign Key referencing Authors table)

-- - PublishedYear

-- The Authors table should have the following columns:

-- - AuthorID (Primary Key, Auto-increment)

-- - AuthorName

-- Insert data for at least three authors and five books.

-- Display:

-- Show the structure of both Books and Authors tables.

-- Note:

-- Use foreign key constraints appropriately, linking Books.AuthorID to Authors.AuthorID.

### **-- Question 3:**

-- Create a database named EmployeeDB and design a table named Employees with the following columns:

- - EmployeeID (Primary Key, Auto-increment)
- - FirstName
- - LastName
- - Department (Default value should be 'Unknown')
- - Salary (Must be greater than or equal to 0)
- Insert data for at least five employees.

-- Display:

-- Show the structure of the Employees table.

-- Note:

-- Ensure that the Department column has a default value of 'Unknown'.

### **-- Question 4:**

-- Create a database named InventoryDB and design two tables: Products and Categories.

-- The Products table should have the following columns:

- - ProductID (Primary Key, Auto-increment)
- - ProductName
- - CategoryID (Foreign Key referencing Categories table)
- - QuantityInStock (Must be greater than or equal to 0)

-- The Categories table should have the following columns:

- - CategoryID (Primary Key, Auto-increment)
- - CategoryName

-- Insert data for at least three categories and five products.

-- Display:

-- Show the structure of both Products and Categories tables.

-- Note:

-- Use foreign key constraints appropriately, linking Products.CategoryID to Categories.CategoryID.

### **-- Question 5:**

-- Create a database named OrdersDB and design three tables: Customers, Orders, and OrderDetails.

-- The Customers table should have the following columns:

-- - CustomerID (Primary Key, Auto-increment)

-- - FirstName

-- - LastName

-- - Email (Must be unique)

-- The Orders table should have the following columns:

-- - OrderID (Primary Key, Auto-increment)

-- - CustomerID (Foreign Key referencing Customers table)

-- - OrderDate

-- The OrderDetails table should have the following columns:

-- - OrderDetailID (Primary Key, Auto-increment)

-- - OrderID (Foreign Key referencing Orders table)

-- - Product

-- - Quantity (Must be greater than 0)

-- Insert data for at least three customers, three orders, and five order details.

-- Display:

-- Show the structure of Customers, Orders, and OrderDetails tables.

-- Note:

-- Use foreign key constraints appropriately, linking Orders.CustomerID to Customers.CustomerID

-- and OrderDetails.OrderID to Orders.OrderID.

### **-- Question 6:**

-- Create a database named HospitalDB and design three tables: Doctors, Patients, and Appointments.

-- The Doctors table should have the following columns:

-- - DoctorID (Primary Key, Auto-increment)

-- - FirstName

-- - LastName

-- - Specialty

-- The Patients table should have the following columns:

- - PatientID (Primary Key, Auto-increment)
- - FirstName
- - LastName
- - DateOfBirth
- - Gender (Must be 'M' or 'F')
- The Appointments table should have the following columns:
  - - AppointmentID (Primary Key, Auto-increment)
  - - DoctorID (Foreign Key referencing Doctors table)
  - - PatientID (Foreign Key referencing Patients table)
  - - AppointmentDate
- Insert data for at least three doctors, three patients, and five appointments.
  
- Display:
  - Show the structure of Doctors, Patients, and Appointments tables.
  
- Note:
  - Use foreign key constraints appropriately, linking Appointments.DoctorID to Doctors.DoctorID
  - and Appointments.PatientID to Patients.PatientID.