



**CHRIST**  
(DEEMED TO BE UNIVERSITY)  
B E N G A L U R U • I N D I A

# **"DBMS Project on Indian Judicial System"**

by

Kirat Singh Chhabra  
(2341343)

**Department of Statistics and Data Science**

**School of Sciences**  
**CHRIST (Deemed to be University)**

# 1. ER Diagram for Indian Judicial System

The Indian Judicial System consists of entities like **Courts, Judges, Cases, Lawyers, and Citizens**. Below is the logical structure for your ER diagram:

## Entities and Relationships

### 1. **Citizen:**

- Attributes: Citizen\_ID (PK), Name, Age, Gender, Address

### 2. **Case:**

- Attributes: Case\_ID (PK), Case\_Type, Case\_Date, Status, Verdict\_Date

### 3. **Court:**

- Attributes: Court\_ID (PK), Court\_Name, Location, Type (Supreme, High, District)

### 4. **Judge:**

- Attributes: Judge\_ID (PK), Name, Specialization, Experience

### 5. **Lawyer:**

- Attributes: Lawyer\_ID (PK), Name, Expertise, Contact

### 6. **Relationships:**

- Citizen** files **Case**: One-to-Many (A citizen can file multiple cases).
- Case** is heard in **Court**: Many-to-One (Many cases in one court).
- Judge** oversees **Case**: Many-to-Many (Multiple judges may oversee multiple cases).
- Lawyer** represents **Citizen**: Many-to-Many (Lawyers can represent multiple citizens).

- **Lawyer** handles **Case**: Many-to-Many (Lawyers can handle multiple cases).

## 2. Database Design (DDL)

### SQL Code for Creating Tables:

-- Citizen Table

```
CREATE TABLE Citizen (  
  
    Citizen_ID INT PRIMARY KEY,  
  
    Name VARCHAR(50),  
  
    Age INT,  
  
    Gender VARCHAR(10),  
  
    Address VARCHAR(100)  
  
);
```

-- Case Table

```
CREATE TABLE Case (  
  
    Case_ID INT PRIMARY KEY,  
  
    Case_Type VARCHAR(30),  
  
    Case_Date DATE,  
  
    Status VARCHAR(20),  
  
    Verdict_Date DATE,  
  
    Citizen_ID INT,  
  
    Court_ID INT,  
  
    FOREIGN KEY (Citizen_ID) REFERENCES Citizen(Citizen_ID),  
  
    FOREIGN KEY (Court_ID) REFERENCES Court(Court_ID)  
  
);
```

-- Court Table

```
CREATE TABLE Court (  
  
    Court_ID INT PRIMARY KEY,  
  
    Court_Name VARCHAR(50),
```

```
        Location VARCHAR(50),

        Type VARCHAR(20)

    );

-- Judge Table

CREATE TABLE Judge (

    Judge_ID INT PRIMARY KEY,

    Name VARCHAR(50),

    Specialization VARCHAR(30),

    Experience INT

);
```

```
-- Lawyer Table

CREATE TABLE Lawyer (

    Lawyer_ID INT PRIMARY KEY,

    Name VARCHAR(50),
```

```
Expertise VARCHAR(30),  
  
Contact VARCHAR(20)  
  
);
```

```
-- Intermediate Table: Judge_Case
```

```
CREATE TABLE Judge_Case (  
  
    Judge_ID INT,  
  
    Case_ID INT,  
  
    FOREIGN KEY (Judge_ID) REFERENCES Judge(Judge_ID),  
  
    FOREIGN KEY (Case_ID) REFERENCES Case(Case_ID),  
  
    PRIMARY KEY (Judge_ID, Case_ID)  
  
);
```

```
-- Intermediate Table: Lawyer_Citizen
```

```
CREATE TABLE Lawyer_Citizen (  
  
    Lawyer_ID INT,
```

```
Citizen_ID INT,  
  
FOREIGN KEY (Lawyer_ID) REFERENCES Lawyer(Lawyer_ID),  
  
FOREIGN KEY (Citizen_ID) REFERENCES Citizen(Citizen_ID),  
  
PRIMARY KEY (Lawyer_ID, Citizen_ID)  
  
);
```

-- Intermediate Table: Lawyer\_Case

```
CREATE TABLE Lawyer_Case (  
  
    Lawyer_ID INT,  
  
    Case_ID INT,  
  
    FOREIGN KEY (Lawyer_ID) REFERENCES Lawyer(Lawyer_ID),  
  
    FOREIGN KEY (Case_ID) REFERENCES Case(Case_ID),  
  
    PRIMARY KEY (Lawyer_ID, Case_ID)  
  
);
```

### **3. SQL Programs (DML/TCL Examples)**

#### **1. Inserting Records (DML)**

```
-- Insert Citizens
```

```
INSERT INTO Citizen VALUES (1, 'Rajesh Kumar', 45, 'Male',  
'Delhi');
```

```
INSERT INTO Citizen VALUES (2, 'Sneha Kapoor', 35, 'Female',  
'Mumbai');
```

```
-- Insert Courts
```

```
INSERT INTO Court VALUES (1, 'Supreme Court', 'Delhi',  
'Supreme');
```

```
INSERT INTO Court VALUES (2, 'High Court Mumbai', 'Mumbai',  
'High');
```

```
-- Insert Cases
```

```
INSERT INTO Case VALUES (101, 'Civil', '2024-01-15', 'Pending',  
NULL, 1, 2);
```

## **2. Updating Records (DML)**

```
-- Update Case Status
```

```
UPDATE Case
```

```
SET Status = 'Closed', Verdict_Date = '2024-10-12'
```



```
WHERE Case_ID = 101;
```

### **3. Deleting Records (DML)**

```
-- Delete a Case
```

```
DELETE FROM Case WHERE Case_ID = 101;
```

### **4. Join Queries**

```
-- Inner Join to display Case details with Court name
```

```
SELECT C.Case_ID, C.Case_Type, Crt.Court_Name
```

```
FROM Case C
```

```
INNER JOIN Court Crt ON C.Court_ID = Crt.Court_ID;
```

### **5. Aggregate Functions**

```
-- Count the total number of cases
```

```
SELECT COUNT(*) AS Total_Cases FROM Case;
```

## 6. Transaction Control (TCL)

-- Start a Transaction

BEGIN TRANSACTION;

-- Insert and Update Example

INSERT INTO Case VALUES (102, 'Criminal', '2024-10-01', 'Open',  
NULL, 2, 1);

UPDATE Case SET Status = 'Closed' WHERE Case\_ID = 102;

-- Commit the Transaction

COMMIT;

## 4. Explanation of SQL Concepts with Example and Output

### 1. DDL (Data Definition Language):

- Used to define and modify database structures.
- Example: CREATE TABLE, ALTER TABLE, DROP TABLE.

## 2. DML (Data Manipulation Language):

- Used to manipulate the data inside tables.
- Example: INSERT, UPDATE, DELETE.

## 3. TCL (Transaction Control Language):

- Manages transactions in the database.
- Example: COMMIT, ROLLBACK.

## 4. Joins:

- Used to combine rows from two or more tables.
- Types: Inner Join, Left Join, Right Join, Full Outer Join.
- Example: The **Inner Join** shown earlier displays case details along with court names.