# BookMyShow Ticketing Platform Database Design

## Part 1: Entities, Attributes, and Table Structures

The database design follows **1NF, 2NF, 3NF, and BCNF** to ensure normalization.

### 1. Movies

* **Attributes**:
  + MovieID (PK)
  + Title
  + ReleaseDate
  + DurationMinutes
  + Deleted (BOOLEAN)
  + Disabled (BOOLEAN)

CREATE TABLE Movies (  
 MovieID INT AUTO\_INCREMENT PRIMARY KEY,  
 Title VARCHAR(255) NOT NULL,  
 ReleaseDate DATE,  
 DurationMinutes INT,  
 Deleted BOOLEAN DEFAULT FALSE,  
 Disabled BOOLEAN DEFAULT FALSE  
);

### 2. Languages

* **Attributes**:
  + LanguageID (PK)
  + LanguageName
  + Deleted
  + Disabled

CREATE TABLE Languages (  
 LanguageID INT AUTO\_INCREMENT PRIMARY KEY,  
 LanguageName VARCHAR(100) NOT NULL,  
 Deleted BOOLEAN DEFAULT FALSE,  
 Disabled BOOLEAN DEFAULT FALSE  
);

### 3. Genres

* **Attributes**:
  + GenreID (PK)
  + GenreName
  + Deleted
  + Disabled

CREATE TABLE Genres (  
 GenreID INT AUTO\_INCREMENT PRIMARY KEY,  
 GenreName VARCHAR(100) NOT NULL,  
 Deleted BOOLEAN DEFAULT FALSE,  
 Disabled BOOLEAN DEFAULT FALSE  
);

### 4. MovieLanguages (Many-to-Many)

* **Attributes**:
  + MovieID (FK)
  + LanguageID (FK)

CREATE TABLE MovieLanguages (  
 MovieID INT,  
 LanguageID INT,  
 PRIMARY KEY (MovieID, LanguageID),  
 FOREIGN KEY (MovieID) REFERENCES Movies(MovieID),  
 FOREIGN KEY (LanguageID) REFERENCES Languages(LanguageID)  
);

### 5. MovieGenres (Many-to-Many)

* **Attributes**:
  + MovieID (FK)
  + GenreID (FK)

CREATE TABLE MovieGenres (  
 MovieID INT,  
 GenreID INT,  
 PRIMARY KEY (MovieID, GenreID),  
 FOREIGN KEY (MovieID) REFERENCES Movies(MovieID),  
 FOREIGN KEY (GenreID) REFERENCES Genres(GenreID)  
);

### 6. Locations

* **Attributes**:
  + LocationID (PK)
  + City
  + State
  + Country
  + Deleted
  + Disabled

CREATE TABLE Locations (  
 LocationID INT AUTO\_INCREMENT PRIMARY KEY,  
 City VARCHAR(100) NOT NULL,  
 State VARCHAR(100),  
 Country VARCHAR(100),  
 Deleted BOOLEAN DEFAULT FALSE,  
 Disabled BOOLEAN DEFAULT FALSE  
);

### 7. Cinemas

* **Attributes**:
  + CinemaID (PK)
  + CinemaName
  + LocationID (FK)
  + Deleted
  + Disabled

CREATE TABLE Cinemas (  
 CinemaID INT AUTO\_INCREMENT PRIMARY KEY,  
 CinemaName VARCHAR(255) NOT NULL,  
 LocationID INT,  
 Deleted BOOLEAN DEFAULT FALSE,  
 Disabled BOOLEAN DEFAULT FALSE,  
 FOREIGN KEY (LocationID) REFERENCES Locations(LocationID)  
);

### 8. Shows

* **Attributes**:
  + ShowID (PK)
  + MovieID (FK)
  + CinemaID (FK)
  + ShowDate
  + ShowTime
  + Deleted
  + Disabled

CREATE TABLE Shows (  
 ShowID INT AUTO\_INCREMENT PRIMARY KEY,  
 MovieID INT,  
 CinemaID INT,  
 ShowDate DATE,  
 ShowTime TIME,  
 Deleted BOOLEAN DEFAULT FALSE,  
 Disabled BOOLEAN DEFAULT FALSE,  
 FOREIGN KEY (MovieID) REFERENCES Movies(MovieID),  
 FOREIGN KEY (CinemaID) REFERENCES Cinemas(CinemaID)  
);

## Sample Entries

-- Movies  
INSERT INTO Movies (Title, ReleaseDate, DurationMinutes) VALUES  
('Inception', '2010-07-16', 148),  
('Avengers: Endgame', '2019-04-26', 181);  
  
-- Languages  
INSERT INTO Languages (LanguageName) VALUES ('English'), ('Hindi');  
  
-- Genres  
INSERT INTO Genres (GenreName) VALUES ('Sci-Fi'), ('Action');  
  
-- MovieLanguages  
INSERT INTO MovieLanguages VALUES (1, 1), (2, 2);  
  
-- MovieGenres  
INSERT INTO MovieGenres VALUES (1, 1), (2, 2);  
  
-- Locations  
INSERT INTO Locations (City, State, Country) VALUES  
('Mumbai', 'Maharashtra', 'India'),  
('Delhi', 'Delhi', 'India');  
  
-- Cinemas  
INSERT INTO Cinemas (CinemaName, LocationID) VALUES  
('PVR Cinemas', 1),  
('INOX', 2);  
  
-- Shows  
INSERT INTO Shows (MovieID, CinemaID, ShowDate, ShowTime) VALUES  
(1, 1, '2025-09-24', '18:30:00'),  
(2, 2, '2025-09-24', '21:00:00');

## Part 2: Query

**List down all shows on a given date at a given theatre along with their respective show timings**

SELECT s.ShowID, m.Title AS MovieTitle, s.ShowDate, s.ShowTime  
FROM Shows s  
JOIN Movies m ON s.MovieID = m.MovieID  
WHERE s.CinemaID = ?  
 AND s.ShowDate = ?  
 AND s.Deleted = FALSE  
 AND s.Disabled = FALSE;

* Replace ? with the actual CinemaID and ShowDate.

Example:

SELECT s.ShowID, m.Title, s.ShowDate, s.ShowTime  
FROM Shows s  
JOIN Movies m ON s.MovieID = m.MovieID  
WHERE s.CinemaID = 1 AND s.ShowDate = '2025-09-24';