**Box Cricket Database Management**

**SQL to crate tables:**

CREATE TABLE User (

UserID INT PRIMARY KEY,

Username VARCHAR(50) UNIQUE NOT NULL,

Password VARCHAR(100) NOT NULL,

Name VARCHAR(100) NOT NULL,

Email VARCHAR(100) UNIQUE NOT NULL,

MobileNumber VARCHAR(20) UNIQUE NOT NULL

);

CREATE TABLE Field (

FieldID INT PRIMARY KEY,

Name VARCHAR(100) NOT NULL,

UserID INT NOT NULL,

FOREIGN KEY (UserID) REFERENCES User(UserID)

);

CREATE TABLE Slot (

SlotID INT PRIMARY KEY,

FieldID INT NOT NULL,

StartTime TIME NOT NULL,

EndTime TIME NOT NULL,

Day VARCHAR(10) NOT NULL,

isAvailable BOOLEAN NOT NULL,

FOREIGN KEY (FieldID) REFERENCES Field(FieldID)

);

CREATE TABLE Booking (

BookingID INT PRIMARY KEY,

UserID INT NOT NULL,

FieldID INT NOT NULL,

StartTime TIME NOT NULL,

EndTime TIME NOT NULL,

BookingDate DATE NOT NULL,

Payment DECIMAL(10,2) NOT NULL,

PlayerNumber INT NOT NULL,

PlayerEmailID VARCHAR(100) NOT NULL,

Status VARCHAR(20) NOT NULL,

FOREIGN KEY (UserID) REFERENCES User(UserID),

FOREIGN KEY (FieldID) REFERENCES Field(FieldID)

);

**Creating Sequences and Triggers for most Primary Keys:**

**For User Table:**

CREATE SEQUENCE user\_id\_seq

START WITH 1

INCREMENT BY 1;

CREATE OR REPLACE TRIGGER user\_id\_trigger

BEFORE INSERT ON "User"

FOR EACH ROW

BEGIN

:NEW.UserID := user\_id\_seq.NEXTVAL;

END;

**For Field table:**

CREATE SEQUENCE field\_id\_seq

START WITH 1

INCREMENT BY 1;

CREATE OR REPLACE TRIGGER field\_id\_trigger

BEFORE INSERT ON Field

FOR EACH ROW

BEGIN

:NEW.FieldID := field\_id\_seq.NEXTVAL;

END;

**For Slot table:**

CREATE SEQUENCE slot\_id\_seq

START WITH 1

INCREMENT BY 1;

CREATE OR REPLACE TRIGGER slot\_id\_trigger

BEFORE INSERT ON Slot

FOR EACH ROW

BEGIN

:NEW.SlotID := slot\_id\_seq.NEXTVAL;

END;

**For Booking table:**

CREATE SEQUENCE booking\_id\_seq

START WITH 1

INCREMENT BY 1;

CREATE OR REPLACE TRIGGER booking\_id\_trigger

BEFORE INSERT ON Booking

FOR EACH ROW

BEGIN

:NEW.BookingID := booking\_id\_seq.NEXTVAL;

END;

**Trigger to check overlapping time slots for a user booking:**

DELIMITER //

CREATE TRIGGER check\_overlap\_booking

BEFORE INSERT ON Booking

FOR EACH ROW

BEGIN

DECLARE overlap\_count INT;

SELECT COUNT(\*)

INTO overlap\_count

FROM Booking

WHERE FieldID = NEW.FieldID

AND StartTime < NEW.EndTime

AND EndTime > NEW.StartTime

AND BookingID != NEW.BookingID;

IF overlap\_count > 0 THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Cannot book overlapping time slots for the same field.';

END IF;

END //

DELIMITER ;

This trigger will check if a new booking overlaps with any existing bookings for the same field. If an overlap is found, it will raise an error and prevent the new booking from being inserted.

**Trigger to ensure that the user creating a field entry in the Field table is the owner of that field:**

CREATE TRIGGER ensure\_field\_ownership

BEFORE INSERT ON Field

FOR EACH ROW

BEGIN

IF NOT EXISTS (

SELECT 1

FROM User

WHERE UserID = NEW.UserID

) THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'User does not exist.';

END IF;

END;

**Trigger to ensure that a slot can only be booked if it is marked as available:**

CREATE TRIGGER ensure\_slot\_availability

BEFORE INSERT ON Booking

FOR EACH ROW

BEGIN

DECLARE is\_slot\_available INT;

SELECT isAvailable

INTO is\_slot\_available

FROM Slot

WHERE SlotID = NEW.SlotID;

IF is\_slot\_available = 0 THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Slot is not available for booking.';

END IF;

END;