```
-- Создание таблицы "Event"
CREATE TABLE Event (
    event_id INT AUTO_INCREMENT PRIMARY KEY,
    event_name VARCHAR(255) NOT NULL,
    event_date DATE NOT NULL,
    event_location VARCHAR(255),
    event_description TEXT,
    event_category VARCHAR(255),
    event_image VARCHAR(255)
);

-- Создание таблицы "Ticket"
CREATE TABLE Ticket (
    ticket_id INT AUTO_INCREMENT PRIMARY KEY,
    ticket_type VARCHAR(255) NOT NULL,
    seat_number INT,
    ticket_price DECIMAL(10, 2) NOT NULL,
    status ENUM('reserved', 'sold', 'available') NOT NULL,
    ticket_purchase_date DATE,
    ticket_barcode VARCHAR(255),
    ticket_notes TEXT
);

-- Создание таблицы "User"
CREATE TABLE User (
    user_id INT AUTO_INCREMENT PRIMARY KEY,
    user_name VARCHAR(255) NOT NULL,
    email VARCHAR(255) NOT NULL,
    phone VARCHAR(20),
    user_address VARCHAR(255),
    user_role ENUM('user', 'administrator') NOT NULL,
    user_preferences TEXT
);

-- Создание таблицы "Reservation" с внешними ключами, связанными с "User" и "Event"
CREATE TABLE Reservation (
    reservation_id INT AUTO_INCREMENT PRIMARY KEY,
    user_id INT,
    event_id INT,
    reserved_date DATE NOT NULL,
    reservation_status ENUM('confirmed', 'awaiting payment') NOT NULL,
    FOREIGN KEY (user_id) REFERENCES User(user_id),
    FOREIGN KEY (event_id) REFERENCES Event(event_id)
);

-- Создание таблицы "Payment" с внешним ключом, связанным с "User"
CREATE TABLE Payment (
    payment_id INT AUTO_INCREMENT PRIMARY KEY,
    user_id INT,
    payment_date DATE NOT NULL,
    payment_amount DECIMAL(10, 2) NOT NULL,
    payment_method VARCHAR(255) NOT NULL,
    FOREIGN KEY (user_id) REFERENCES User(user_id)
);

-- Создание таблицы для связи "Ticket" и "Payment" (Many-to-Many)
CREATE TABLE TicketPayment (
    ticket_id INT,
    payment_id INT,
    PRIMARY KEY (ticket_id, payment_id),
    FOREIGN KEY (ticket_id) REFERENCES Ticket(ticket_id),
    FOREIGN KEY (payment_id) REFERENCES Payment(payment_id)
);
```