

Event Management System

Presented By

Saad Ajmal

Shehryar Chaudhary

Overview

- The Event Management System is designed to manage and organize events efficiently.
- It stores structured data about users, events, venues, bookings, tickets, payments, staff, food, decorations, and sponsors.
- Admin can monitor and control overall system operations.
- Event organizers can create, update, and manage events.
- Customers can book events and purchase tickets.
- We also did normalization to remove data redundancy

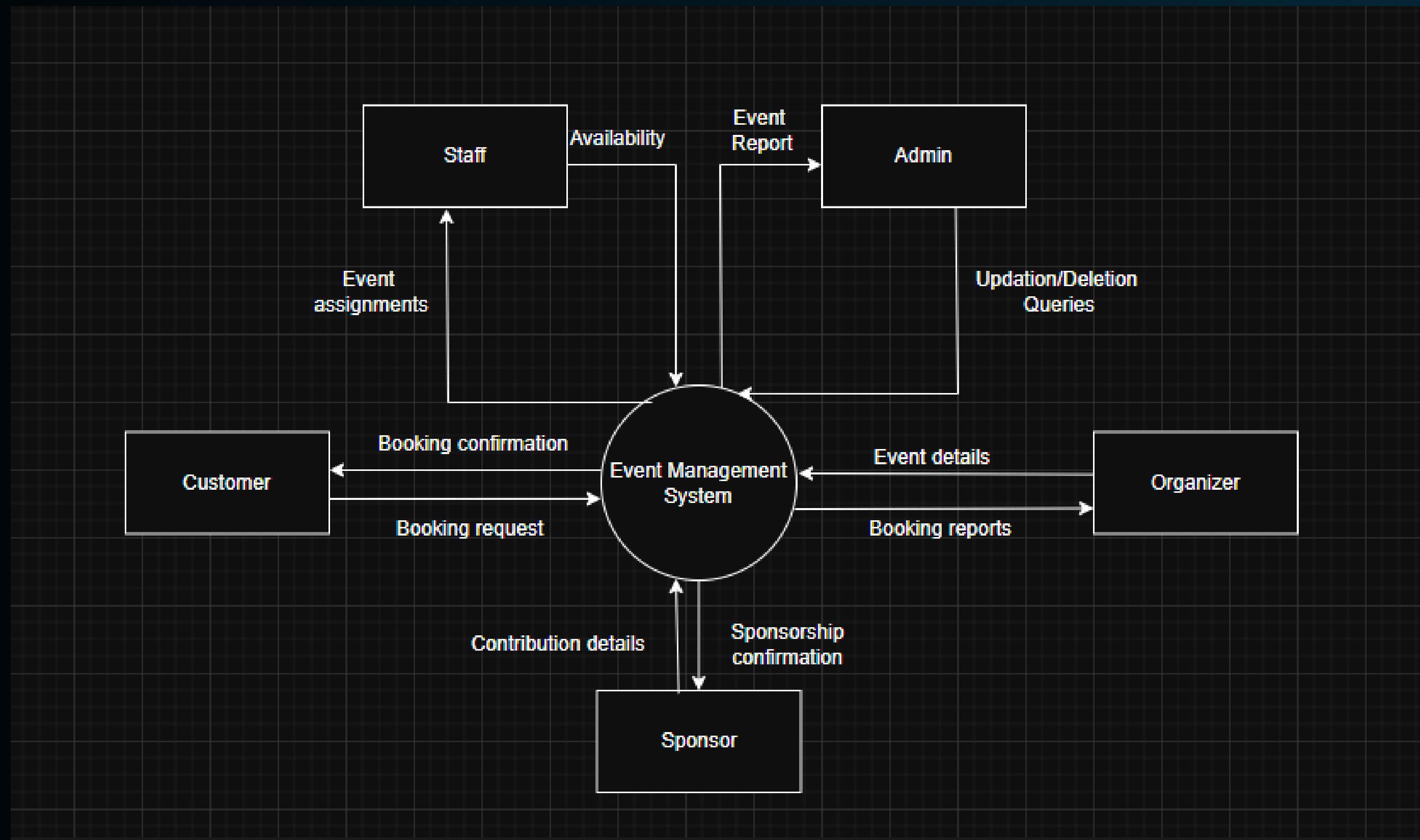
System Users

- Admin : Manages and controls the system
- Organizer : Creates and manages events
- Customer : Registers, books events, and buys tickets
- Staff : Supports event operations

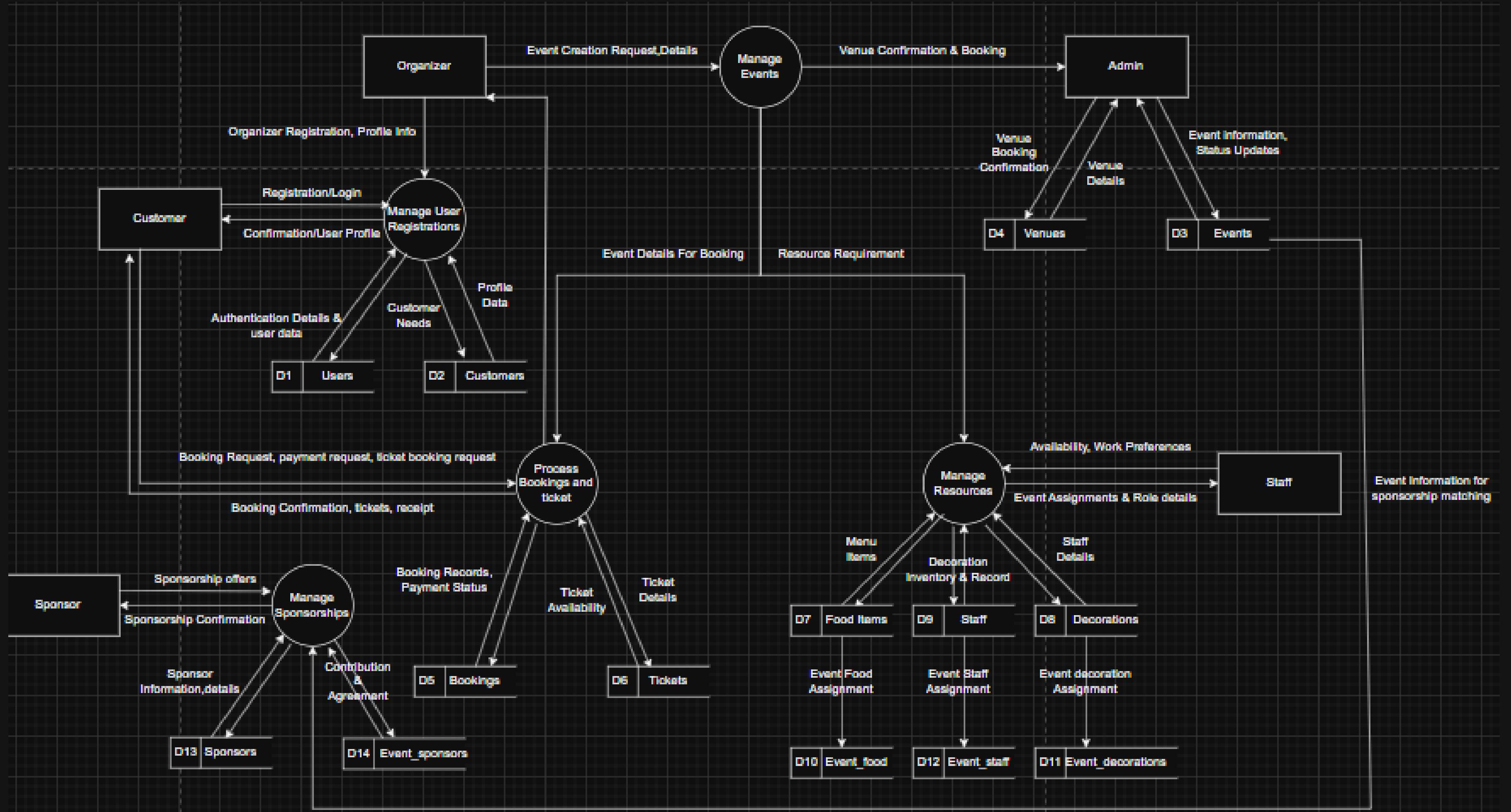
Major Entities Used

- Users
- Customers
- Staff
- Events
- Sponsors
- Payment Modes
- Food Items
- Decorations
- Venues
- Bookings
- Tickets

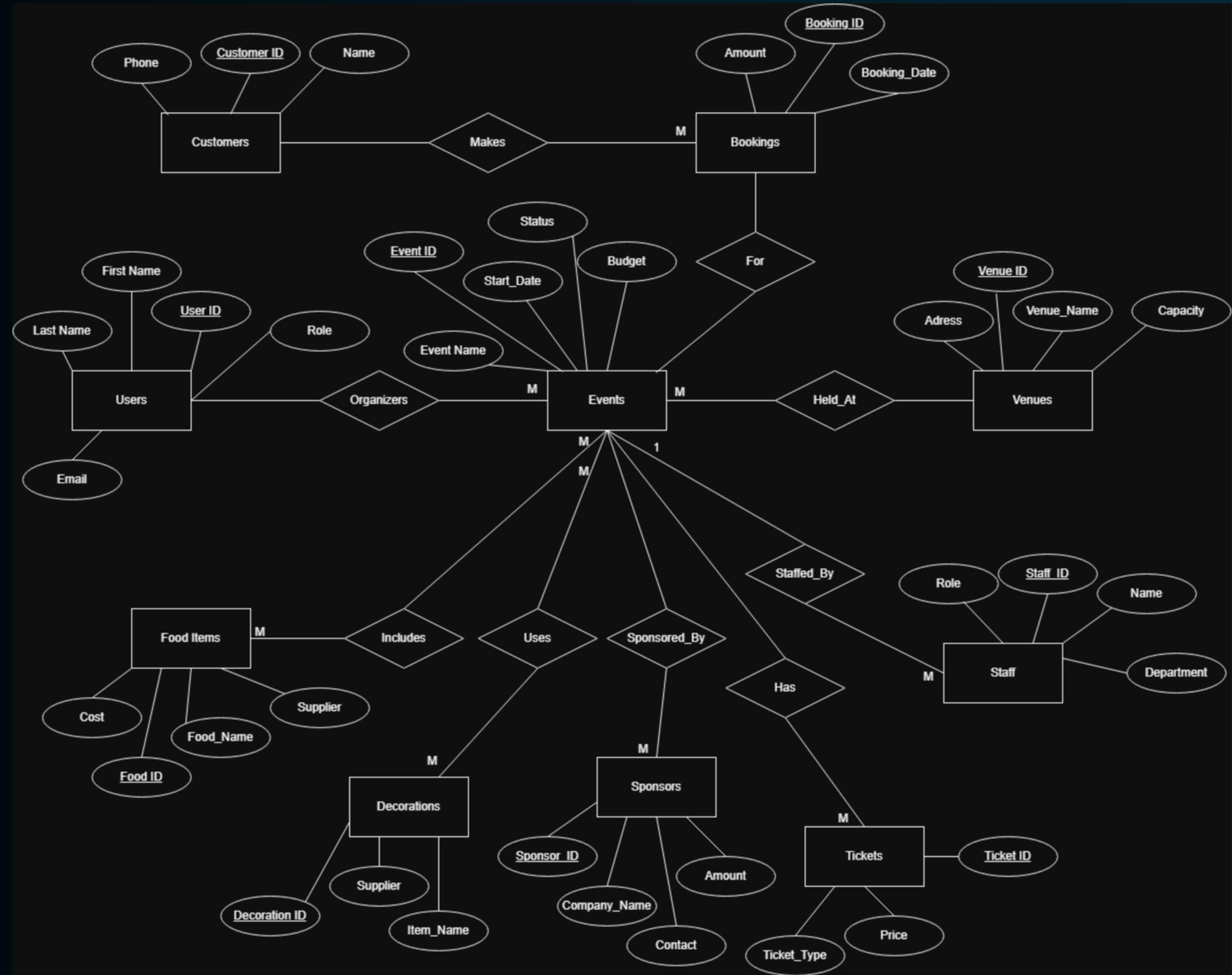
DFD - level 0



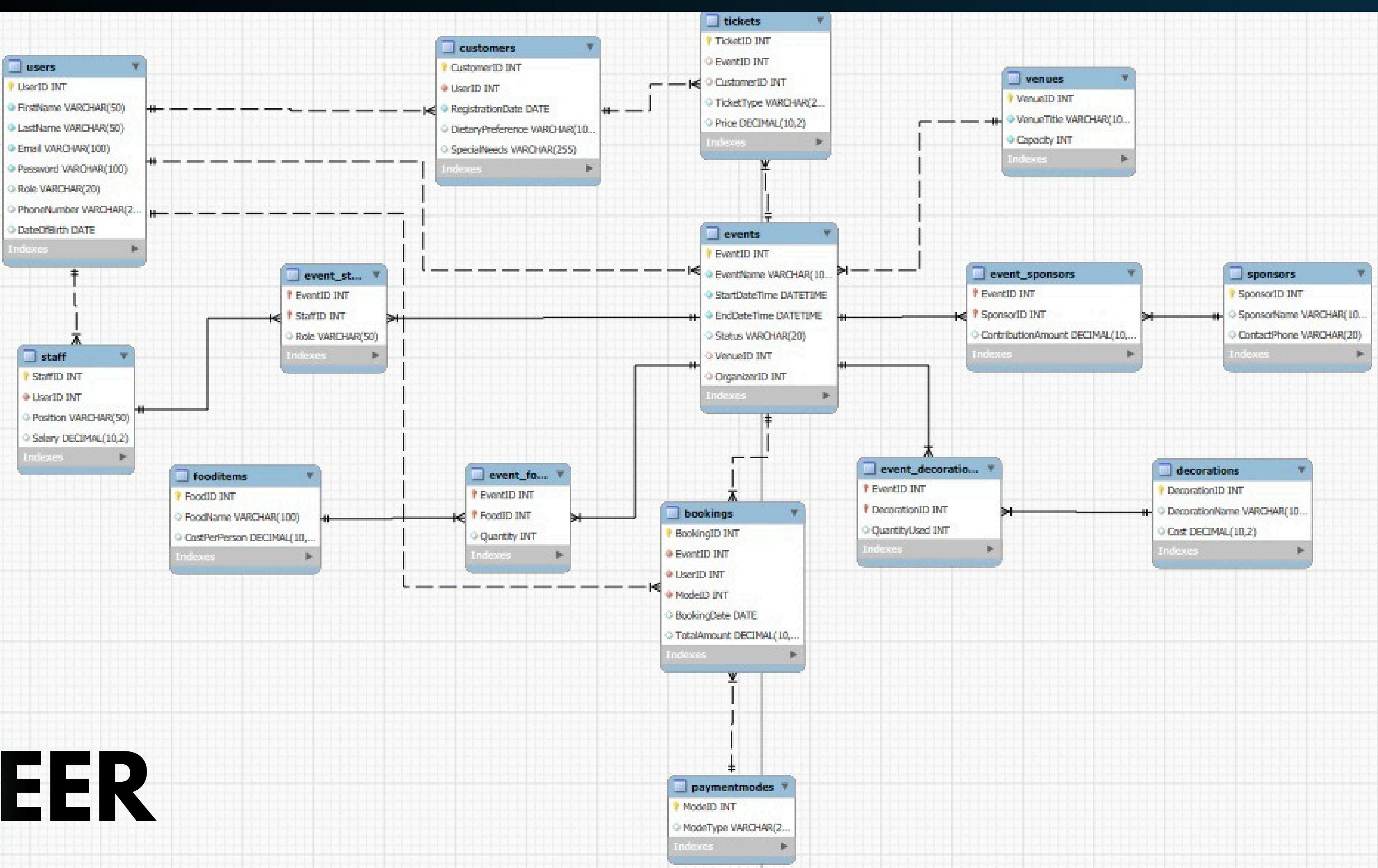
DFD - level 1



ERD



EER



SQL operations used in EMS

DDL

- CREATE DATABASE
- USE DATABASE
- CREATE TABLE
- ALTER TABLE
- DROP TABLE

DML

- INSERT
- UPDATE
- DELETE
- SELECT

AGGREGATE FUNCTIONS

- COUNT()
- SUM()
- MAX()
- MIN()

Normalization Technique

Event Management System is normalized up to Third Normal Form (3NF)

WHY 3NF?

- Third Normal Form (3NF) is used because the system contains many interconnected entities (users, events, venues, bookings, payments, etc.).
- Each entity has its own separate table, avoiding data duplication.
- All non-key attributes depend only on the primary key.
- There are no partial or transitive dependencies.
- This structure improves data consistency and accuracy.
- It also makes the database easier to maintain and update.

Normalization Tables

Normalized Table	Created From	Purpose
AddressDetails	Venues	Stores address information separately
VenueManagers	Venues	Separates venue manager details
Departments	Staff	Eliminates repeated department names
FoodCategories	FoodItems	Normalizes food type and cuisine
DecorationTypes	Decorations	Separates decoration categories
SponsorIndustries	Sponsors	Removes repeated industry names

NESTED QUERIES

```
UPDATE STAFF  
SET DEPARTMENTID = (  
    SELECT DEPARTMENTID  
    FROM DEPARTMENTS  
    WHERE STAFF.DEPARTMENT = DEPARTMENTS.DEPARTMENTNAME  
);
```

```
UPDATE FOODITEMS  
SET CATEGORYID = (  
    SELECT CATEGORYID  
    FROM FOODCATEGORIES  
    WHERE FOODITEMS.TYPE = FOODCATEGORIES.TYPE  
        AND FOODITEMS.CUISINETYPE = FOODCATEGORIES.CUISINETYPE  
);
```

JOINS

- SIMPLE JOIN

```
SELECT E.EVENTID,E.EVENTNAME,V.VENUETITLE, E.ENDDATETIME  
FROM EVENTS E  
JOIN VENUES V ON E.VENUEID = V.venueID;
```

- LEFT JOIN

```
SELECT USERS.FIRSTNAME,USERS.LASTNAME,BOOKINGS.BOOKINGID  
FROM USERS  
LEFT JOIN BOOKINGS  
ON USERS.USERID = BOOKINGS.USERID;
```

LIMITATIONS

- The system does not include a graphical user interface (GUI); all operations are performed using SQL queries.
- A live dashboard could not be fully implemented because Tableau requires a paid version for real-time database connections.
- The database focuses only on backend data design and does not handle frontend user interaction.
- User authentication and security features (password encryption, login sessions) are not implemented

Thank You