The purpose of this project is to develop
and implement an event organizing database system
that can provide user-friendly solution that can facilitate
and improve event organizing. The system will
allow users to create and manage events, book and allocate venues,
register and communicate with attendees, and perform
other related tasks. The system will aim to improve the
efficiency, reduce the costs, and enhance
the quality of event organizing.The project will address the current
problems and limitations of existing event
organizing methods or systems, such as manual processes,
paper-based records, lack of integration
, and so on. These problems can cause errors, delays, waste,
and dissatisfaction among the users and the stakeholders.

Venue

venue_id{PK}
name
capacity
state
city
Street
cost

Customer

customer_id
first_name
last_name
sex
TellNo[1..2]

◀**Supervise**

1..1

◀**Happens**

1..1

0..*

◀**Order**

0..*

Supervisor
0..1

Supervisee
0..*

Staff

staff_id
first_name
last_name
DOB
appointed_date
position
tellNo[1..2]
salary

Organize▶

0..*

0..*

Event

event_id{PK}
event_name
event_date
start_time
end_time
guest_no

◀**Attends**

0..*

0..*

Guest

guest_id{PK}
first_name
last_name

0..*

Is▼

1..1

EvnetType

id{PK}
event_name
cost_per_person

Relations

1)Venue (venue_Id, name, capacity, state , city, street, cost) **-** is created from the entity **Venue.**

2)Event (event_Id, event_name, type, venue_Id, customer_id,event_date, start_time, end_time, guest_no) – is created from the intersection of entities **Event, Customer, venue** and **event_type .**

3)Customer (customer_Id, first_name, last_name, tellNo) – is created from the entity **Customer.**

4)Guest (guest_Id, first_name, last_name) – is created from the entity **Guest.**

5)Staff (staff_id, first_name, last_name, sex, position, DOB, tellNo, appointed_date, salary, supervisor_id) – is created from the recursive relation of the entity **Staff.**

6)EventGuests(evnet_id, guest_id) – is created from the intersection of entities **Event** and **Guests.**

7)EventStaff(evnet_id, staff_id) – is created from the intersection of entities **Event** and **Staff.**

8)EventType(id, name,cost_per_person) – is created from the entitie **EventType.**

To First Normalization Form

Customer (customer_Id, first_name, last_ame)

CustomerTellNo(customer_id, tellNo)

Guest (guest_Id, fName, lName)

Staff (staff_id, first_name, last_name, sex, position, salary, DOB, supervisor_id)

StaffTellNo(staff_id, tellNo)

Other relations are in **first normal form**.

To Second Normalization Form

All relations are in **second normal form.**

To Third Normalization Form

Staff (staff_id, first_name, last_name, sex, position, DOB, supervisor_id)

StaffSalary(position, salary)

Other relations are in **third normal form.**

Staff (staff_id, first_name, last_name, sex, position_id, DOB, supervisor_id) **Primary Key** staff_id **Foreign Key s**upervisor_id **references** Staff(staff_id) **Foreign Key** position_id **references** StaffSalary(id)	**Customer** (customer_Id, first_name, last_name) **Primary Key** customer_Id
Venue (venue_Id, name, capacity, state, city, street) **Primary Key** venue_Id	**Guest** (guest_Id, first_name, last_name) **Primary Key** guest_Id
Event (event_Id, event_name, type_id, venue_Id, customer_id, event_date, start_time, guest_no) **Primary Key** event_Id **Foreign Key** venue_id **references** venue(venue_id) **Foreign Key** customer_id **references** customer(customer_id) **Foreign Key** type_id **references EventType**(id)	**EventType** (id, name, cost_per_person) **Primary Key** id
EventGuests(evnet_id, guest_id) **Primary Key** event_id, guest_id **Foreign Key** event_id **references** Event(event_id) **Foreign Key** guest_id **references** Guest(guest_id)	**EventStaff**(evnet_id, staff_id) **Primary Key** event_id, staff_id **Foreign Key** event_id **references** Event(event_id) **Foreign Key** staff_id **references** Staff(staff_id)
StaffSalary(id,position, salary) **Primary Key** id	

```
CREATE TABLE staffPosition (
    id INT AUTO_INCREMENT PRIMARY KEY,
    position VARCHAR(100) not NULL,
    salary DECIMAL(10, 2) not null
);

CREATE TABLE Staff (
    id INT AUTO_INCREMENT PRIMARY KEY,
    first_name VARCHAR(100) not null,
    last_name VARCHAR(100) not null,
    sex CHAR(1) not null,
    position_id INT not null,
    DOB DATE not null,
    supervisor_id INT not null,
    password VARCHAR(100) not null,
    tellNo1 VARCHAR(20) not null,
    tellNo2 VARCHAR(20) not null,
    event_work int DEFAULT(0),
    apointed_date date not NULL,
    FOREIGN KEY (position_id) REFERENCES staffPosition(id),
    FOREIGN KEY (supervisor_id) REFERENCES Staff(id)
);

CREATE TABLE Customer (
    id INT AUTO_INCREMENT PRIMARY KEY,
    first_name VARCHAR(100) not null,
    last_name VARCHAR(100) not null,
    tellNo1 VARCHAR(20) not null,
    tellNo2 VARCHAR(20),
    password VARCHAR(100) not null,
    sex VARCHAR(1) not null
);
```

```
CREATE TABLE Venue (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100) not null,
    capacity INT not null,
    state VARCHAR(100) not Null,
    city VARCHAR(100) not null,
    price DECIMAL(10, 2) not null,
    street VARCHAR(100)
);

CREATE TABLE Guest (
    id INT AUTO_INCREMENT PRIMARY KEY,
    first_name VARCHAR(100) not null,
    last_name VARCHAR(100) not null
);

CREATE TABLE EventType (
    id INT AUTO_INCREMENT PRIMARY KEY,
    type_name VARCHAR(100) not null,
    cost_per_person DECIMAL(10, 2) not null
);

CREATE TABLE Event (
    id INT AUTO_INCREMENT PRIMARY KEY,
    event_name VARCHAR(100),
    type_id INT not null,
    venue_id INT not null,
    event_date DATE not null,
    start_time TIME not null,
    end_time TIME not null,
    customer_id INT not null,
    guest int not null,
    event_cost decimal(10,2) not null,
    FOREIGN KEY (venue_id) REFERENCES Venue(id),
    FOREIGN KEY (type_id) REFERENCES EventType(id),
    FOREIGN KEY (customer_id) REFERENCES Customer(id)
);

CREATE TABLE EventGuests (
```

```
CREATE TABLE EventGuests (
    event_id INT,
    guest_id INT,
    PRIMARY KEY (event_id, guest_id),
    FOREIGN KEY (event_id) REFERENCES Event(id),
    FOREIGN KEY (guest_id) REFERENCES Guest(id)
);

CREATE TABLE EventStaff (
    event_id INT,
    staff_id INT,
    PRIMARY KEY (event_id, staff_id),
    FOREIGN KEY (event_id) REFERENCES Event(id),
    FOREIGN KEY (staff_id) REFERENCES Staff(id)
);

INSERT into staffposition(id,position, salary)
VALUES (1,'Wedding Planner', 50000),
(2,'Birthday Planner', 48000),
(3,'Graduation Planner', 47000),
(4,'Event Manager', 55000),
(5,'Human Resource', 60000),
(6,'Supervisor', 65000),
(7,'Waiter', 30000),
(8,'Security', 40000),
(9,'Valet', 35000);

INSERT into eventtype(id, type_name, cost_per_person)
VALUES(1, "WEDDING", 500),
      (2, "BIRTHDAY", 400),
      (3, "GRADUATION", 350);
```