Project Introduction:

Our project is Event Planner Website, by which people will know about Event planning & Book Consultation for an Event.

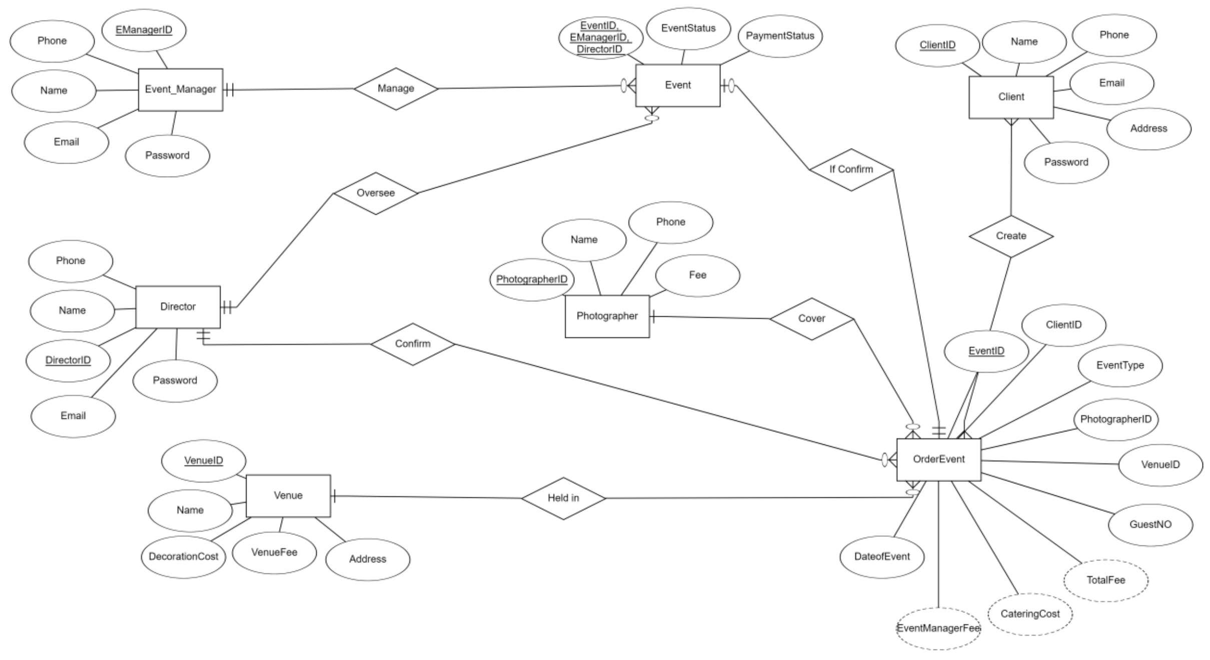
Here we discuss about our project's

- ✓ Entity Relationship Diagram
- ✓ Relational Model
- ✓ SQL Commands
- ✓ Inserting some dummy data to justify the multiplicities

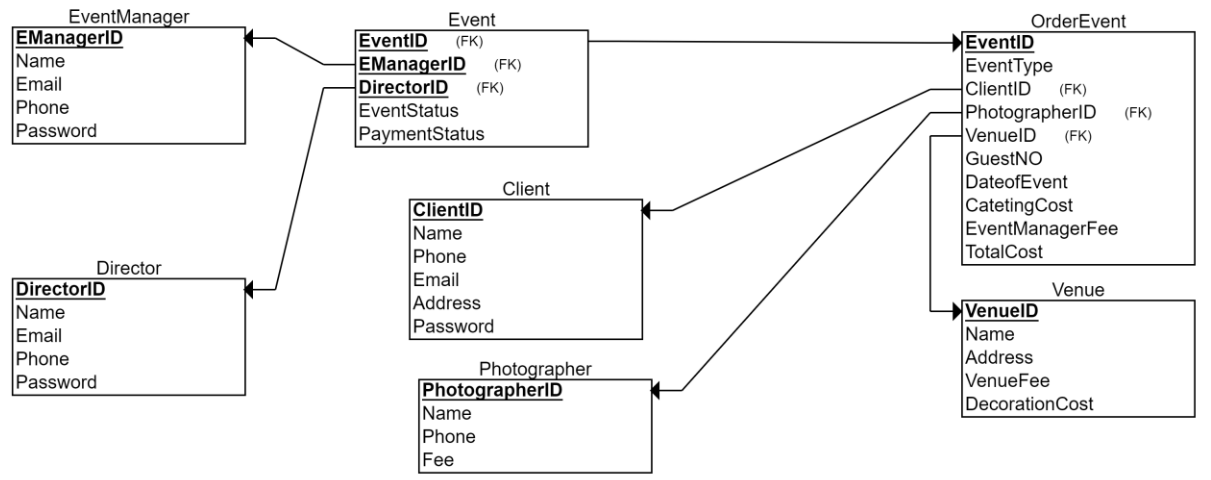
Entities:

- Event_Manager (<u>EManagerID</u>, Name, Email, Phone, Password)
- Director (<u>DirectorID</u>, Name, Email, Phone, Password)
- Client (<u>ClientID</u>, Name, Email, Address, Phone, Password)
- Venue (<u>VenueID</u>, Name, Address, VenueFee, DecorationCost)
- Photographer (<u>PhotographerID</u>, Name, Phone, Fee)
- EventOrder (<u>EventID</u>, ClientID(FK), EventType, VenueID(FK), PhotographerID(FK), GuestNO, DateofEvent, CateringCost, EventManagerFee, TotalCost)
- Event (<u>EventID</u>(FK), <u>EMangerID</u>(FK), <u>DirectorID</u>(FK), EventStatus, PaymentStatus)

Entity Relationship (ER) diagram:



Relational Model

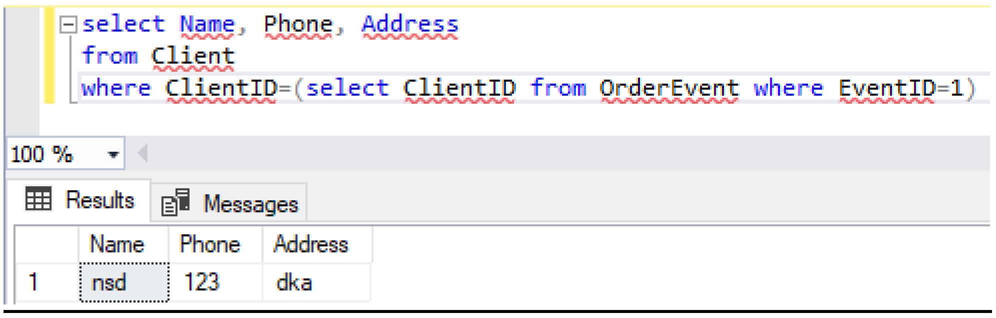


SQL Commands

```
CREATE TABLE EventManager

(

  EManagerID INT NOT NULL,

  Name VARCHAR(100) NOT NULL,

  Email VARCHAR(100) NOT NULL,

  Phone VARCHAR(30) NOT NULL,

  Password VARCHAR(30) NOT NULL,

  PRIMARY KEY (EManagerID)

);


CREATE TABLE Director

(

  DirectorID INT NOT NULL,

  Name VARCHAR(100) NOT NULL,

  Email VARCHAR(100) NOT NULL,

  Phone VARCHAR(30) NOT NULL,

  Password VARCHAR(30) NOT NULL,

  PRIMARY KEY (DirectorID)
```

```
);

CREATE TABLE Venue
(
  VenueID INT NOT NULL,
  Name VARCHAR(100) NOT NULL,
  Address VARCHAR(100) NOT NULL,
  VenueFee INT NOT NULL,
  DecorationCost INT NOT NULL,
  PRIMARY KEY (VenueID)
);

CREATE TABLE Photographer
(
  PhotographerID INT NOT NULL,
  Name VARCHAR(100) NOT NULL,
  Phone VARCHAR(30) NOT NULL,
  Fee INT NOT NULL,
  PRIMARY KEY (PhotographerID)
);

CREATE TABLE Client
(
  ClientID INT NOT NULL,
  Name VARCHAR(100) NOT NULL,
  Phone VARCHAR(30) NOT NULL,
  Email VARCHAR(100) NOT NULL,
  Address INT NOT NULL,
  Password VARCHAR(100) NOT NULL,
  PRIMARY KEY (ClientID)
);

CREATE TABLE OrderEvent
(
  EventID INT NOT NULL,
  EventType VARCHAR(100) NOT NULL,
  GuestNO INT NOT NULL,
  DateofEvent DATE NOT NULL,
  CatetingCost INT NOT NULL,
  EventManagerFee INT NOT NULL,
  TotalCost INT NOT NULL,
  ClientID INT NOT NULL,
  PhotographerID INT NOT NULL,
  VenueID INT NOT NULL,
  PRIMARY KEY (EventID),
  FOREIGN KEY (ClientID) REFERENCES Client(ClientID),
  FOREIGN KEY (PhotographerID) REFERENCES Photographer(PhotographerID),
  FOREIGN KEY (VenueID) REFERENCES Venue(VenueID)
);

CREATE TABLE Event
(
  EventStatus VARCHAR(100) NOT NULL,
  PaymentStatus VARCHAR(100) NOT NULL,
  EManagerID INT NOT NULL,
  DirectorID INT NOT NULL,
  EventID INT NOT NULL,
  PRIMARY KEY (EManagerID, DirectorID, EventID),
  FOREIGN KEY (EManagerID) REFERENCES EventManager(EManagerID),

  FOREIGN KEY (DirectorID) REFERENCES Director(DirectorID),

  FOREIGN KEY (EventID) REFERENCES OrderEvent(EventID)

);
```

Justify the multiplicities



```
select Name, Phone, Address
from Client
where ClientID=(select ClientID from OrderEvent where EventID=1)
```

```
   Name  Phone  Address
1  nsd   123    dka
```

Conclusion:

We have analysed all our functionality of the project related to the data management before designing the entity relationship diagram so that so that proper CRUD and other deriving operations can be done properly without any anomalistic action. For a flawless development of a software a proper ERD designing is mandatory and this will help us proceed to the next step of development.