

EVENT MANAGEMENT SYSTEM

Team members:

Mazen Ahmed Mahmoud | Sec: 11

Mazen Emad | Sec: 11

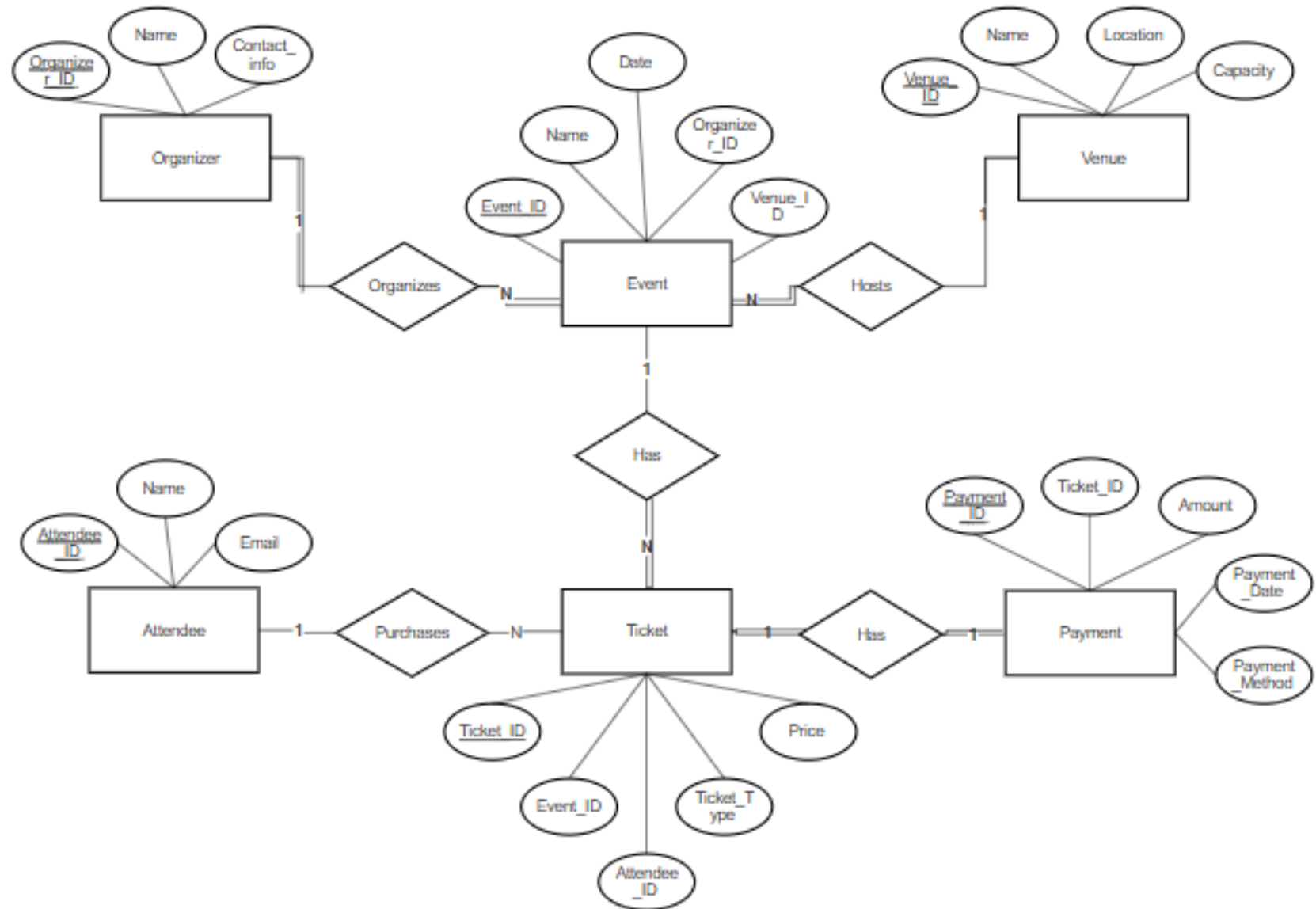
Mohaned Magdy | Sec: 15

Marko Emil | Sec: 10

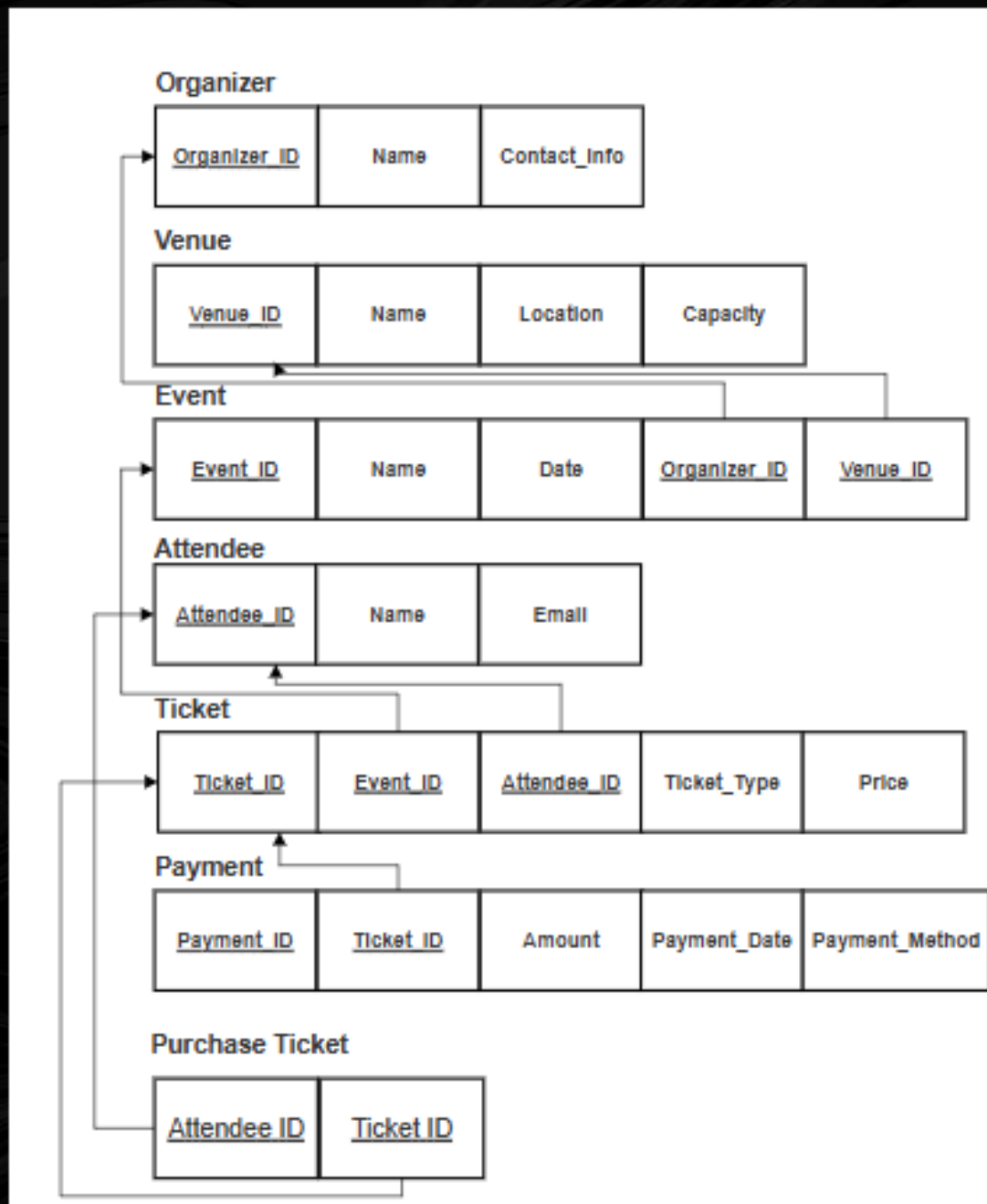
Fareed Ali | Sec: 10

ENG: Asmaa Hussein

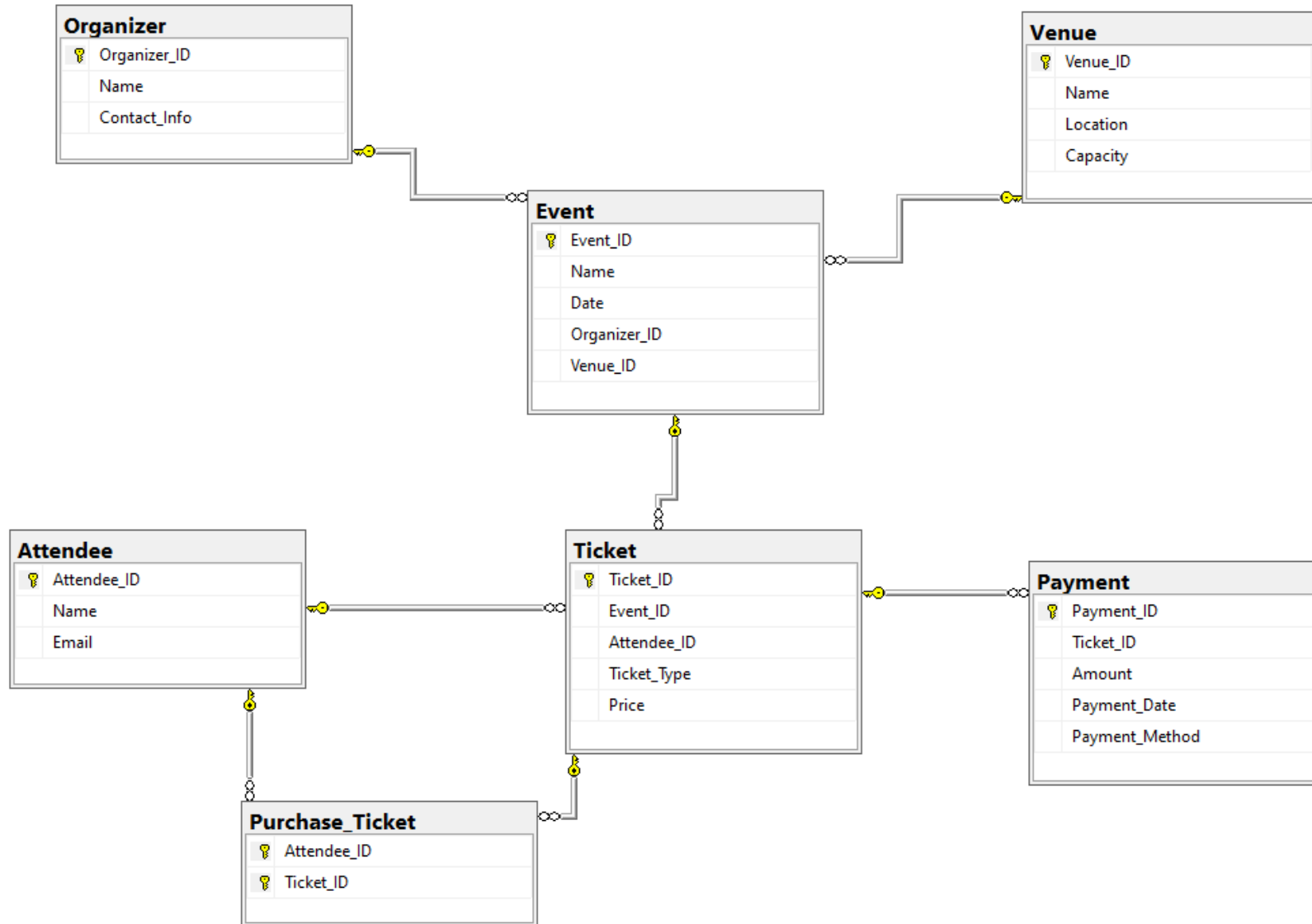
The ER conceptual schema diagram:



The Relational Schema for the case study:



Database Diagram



SQL SERVER CODE

```
= CREATE DATABASE EventManagementSystem;

USE EventManagementSystem;

= CREATE TABLE Organizer (
    Organizer_ID INT IDENTITY(1,1),
    Name NVARCHAR(255) NOT NULL,
    Contact_Info NVARCHAR(255) UNIQUE NOT NULL,
    CONSTRAINT pk_Organizer PRIMARY KEY (Organizer_ID),
    CONSTRAINT uq_Contact_Info UNIQUE (Contact_Info)
);

= CREATE TABLE Venue (
    Venue_ID INT IDENTITY(1,1),
    Name NVARCHAR(255) NOT NULL,
    Location NVARCHAR(255) NOT NULL,
    Capacity INT NOT NULL CHECK (Capacity > 0),
    CONSTRAINT pk_Venue PRIMARY KEY (Venue_ID),
    CONSTRAINT uq_Venue_Name_Location UNIQUE (Name, Location)
);
```


SQL SERVER CODE

```
= CREATE TABLE Event (  
    Event_ID INT IDENTITY(1,1),  
    Name NVARCHAR(255) NOT NULL,  
    Date DATE NOT NULL CHECK (Date >= GETDATE()),  
    Organizer_ID INT NOT NULL,  
    Venue_ID INT NOT NULL,  
    CONSTRAINT pk_Event PRIMARY KEY (Event_ID),  
    CONSTRAINT fk_Event_Organizer FOREIGN KEY (Organizer_ID) REFERENCES Organizer(Organizer_ID),  
    CONSTRAINT fk_Event_Venue FOREIGN KEY (Venue_ID) REFERENCES Venue(Venue_ID),  
    CONSTRAINT uq_Event_Name_Date_Venue UNIQUE (Name, Date, Venue_ID)  
);  
  
= CREATE TABLE Attendee (  
    Attendee_ID INT IDENTITY(1,1),  
    Name NVARCHAR(255) NOT NULL,  
    Email NVARCHAR(255) UNIQUE NOT NULL,  
    CONSTRAINT pk_Attendee PRIMARY KEY (Attendee_ID),  
    CONSTRAINT uq_Attendee_Email UNIQUE (Email)  
);
```

SQL SERVER CODE

```
CREATE TABLE Ticket (  
    Ticket_ID INT IDENTITY(1,1),  
    Event_ID INT NOT NULL,  
    Attendee_ID INT NOT NULL,  
    Ticket_Type NVARCHAR(50) NOT NULL CHECK (Ticket_Type IN ('VIP', 'Standard', 'Economy')),  
    Price DECIMAL(10, 2) NOT NULL CHECK (Price >= 0),  
    CONSTRAINT pk_Ticket PRIMARY KEY (Ticket_ID),  
    CONSTRAINT fk_Ticket_Event FOREIGN KEY (Event_ID) REFERENCES Event(Event_ID),  
    CONSTRAINT fk_Ticket_Attendee FOREIGN KEY (Attendee_ID) REFERENCES Attendee(Attendee_ID),  
    CONSTRAINT uq_Ticket_Event_Attendee UNIQUE (Event_ID, Attendee_ID)  
);  
  
CREATE TABLE Payment (  
    Payment_ID INT IDENTITY(1,1),  
    Ticket_ID INT NOT NULL,  
    Amount DECIMAL(10, 2) NOT NULL CHECK (Amount >= 0),  
    Payment_Date DATE DEFAULT GETDATE() NOT NULL,  
    Payment_Method NVARCHAR(50) NOT NULL CHECK (Payment_Method IN ('Credit Card', 'Cash', 'Bank Transfer')),  
    CONSTRAINT pk_Payment PRIMARY KEY (Payment_ID),  
    CONSTRAINT fk_Payment_Ticket FOREIGN KEY (Ticket_ID) REFERENCES Ticket(Ticket_ID)  
);  
  
CREATE TABLE Purchase_Ticket (  
    Attendee_ID INT NOT NULL,  
    Ticket_ID INT NOT NULL,  
    CONSTRAINT pk_Purchase_Ticket PRIMARY KEY (Attendee_ID, Ticket_ID),  
    CONSTRAINT fk_Purchase_Ticket_Attendee FOREIGN KEY (Attendee_ID) REFERENCES Attendee(Attendee_ID),  
    CONSTRAINT fk_Purchase_Ticket_Ticket FOREIGN KEY (Ticket_ID) REFERENCES Ticket(Ticket_ID)  
);
```



THANK YOU