

Assignment - 01

T. Arunkumar

192372325

CSA0593-[DBMS]

Assignment-01

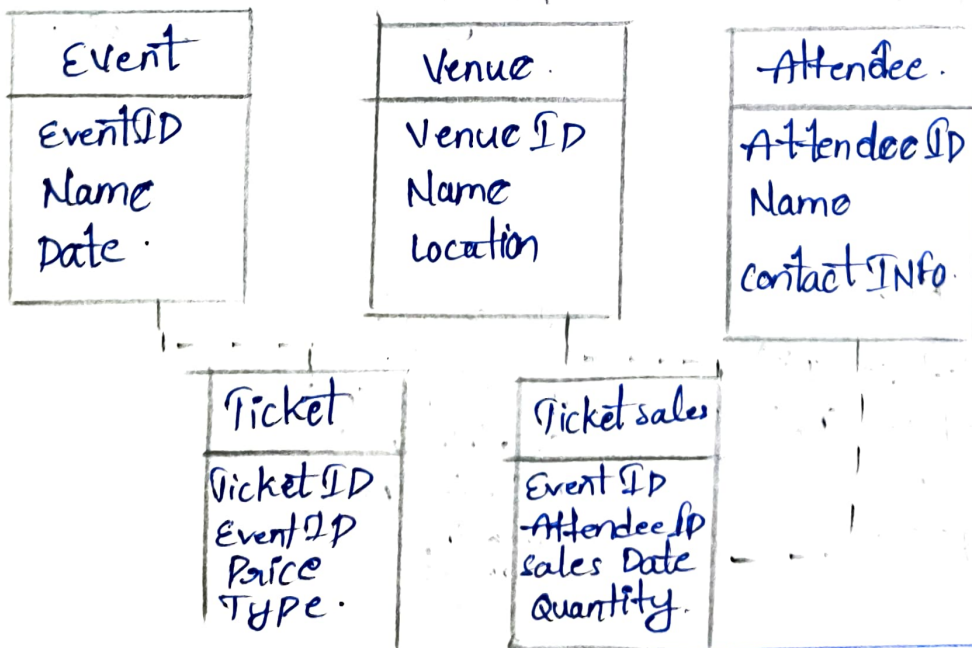
T. Arunkumar
192372325

①

Event management system with Real-Time Analysis
Design a database for managing events, attendee, tickets, and Venue logistics Requirements
create table for events, Venue tickets, with relationship defined between each
write SQL queries to provide real-time analytics on ticket sales attendance and revenue.
Add triggers to automatically update event capacity and notify event organizers when a threshold of tickets is reached.

1. Conceptual ER diagram :-

The conceptual ER diagram provides a high level overview of the main entities and their relationships in the system.

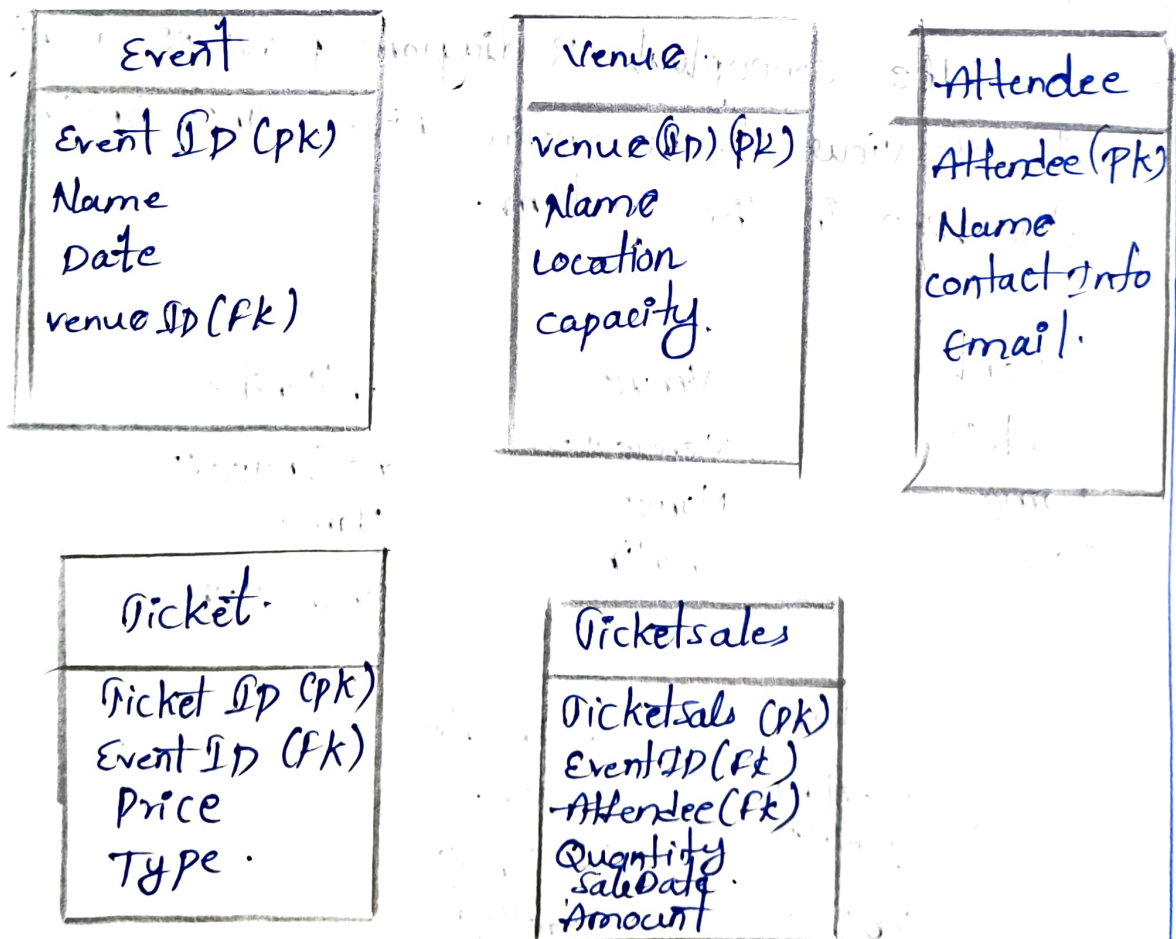


Conceptual Model Explanation :

- ★ Event : Represents each event managed by system.
- ★ venue : Represents the venue where the event place.
- ★ Attendee : Represents individuals attending events.
- ★ Ticket : Represents the type and prices of tickets for event.
- ★ Ticketsale : Represents ticket sales and transactions for each attendee

2. Logical ER diagram :-

The logical ER diagram adds some more detail including primary key, foreign key, and attributes that specify entity relationships.



Logical Model Explanation!

- primary keys (pk): unique Identifiers for each table.
- foreign keys (fk): linking tables eg: Event ID in Ticket referencing Event ID
- Additional Attributes :-

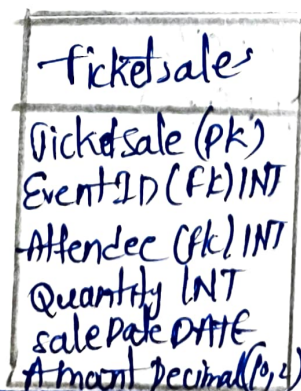
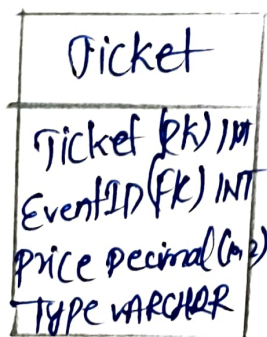
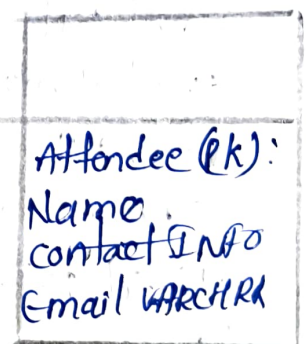
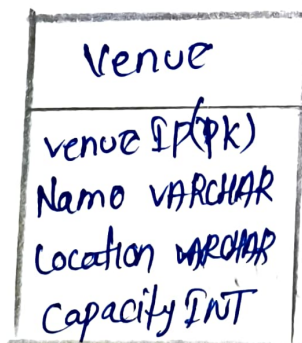
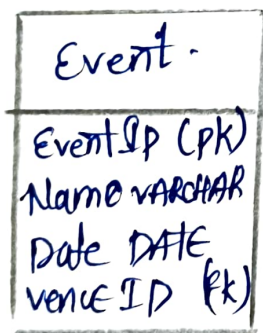
Event includes a venue ID as a foreignkey.

Venue includes capacity attribute for seating limits.

Ticket sales includes Quantity and Amount for each sale.

Physical ER diagram

The physical ER diagram further specifies data types, constraints, and table structures for implementation in a database management system.



Physical Model Explanation

Data Type :

- Integer types (INT) for IDs and Capacities
- VARCHAR for text data like name and types
- Decimal for monetary amounts.

Constraints :

- Each primary key is unique identifying each row
- foreign keys enforce referential integrity.

SQL statements :-

```
CREATE TABLE Events(  
    EventID INT Primary Key,  
    Event Name VARCHAR (255),  
    Event Date DATE,  
    Event Time TIME,  
    venue ID INT,  
    Capacity INT,  
    Ticket price Decimal (10, 2)  
);
```

```
CREATE TABLE venues (  
    venue ID INT Primary Key,  
    venue Name VARCHAR (255)  
    Address VARCHAR (255),  
    City VARCHAR (100),  
    State VARCHAR (100)  
);
```

```
CREATE TABLE Tickets (  
  TicketID INT Primary Key,  
  EventID INT,  
  AttendeeID INT,  
  TicketType VARCHAR(100),  
  TicketStatus VARCHAR(100),  
  PurchaseDate DATE,  
  PurchaseTime TIME,  
  PricePaid Decimal(10,2),  
  Foreign Key (EventID) References  
  Events (EventID),  
  Foreign Key (AttendeeID) References  
  Attendees (AttendeeID)  
);
```

```
CREATE TABLE Attendees (  
  AttendeeID INT Primary Key,  
  FirstName VARCHAR(100),  
  LastName VARCHAR(100),  
  Email VARCHAR(225),  
  Phone VARCHAR(20)  
);
```


CREATE TABLE Waitlist (

Waitlist INT PRIMARY KEY,

EventID INT

AttendeeID INT,

waitlist Date DATE

waitlist Time TIME,

FOREIGN KEY (AttendeeID) REFERENCES

Attendees (AttendeeID)

);

Conclusion.

The proposed Event management system's database effectively manages events, attendees, ticket and venue logistics.

1. Scalable database structure :- The design incorporates separate tables for events, Venues, ticket ensuring scalability and flexibility.
2. Relationships and constraints :- Defined relationships b/w table ensure data consistency, while constraints prevent invalid data entry.

3. Stored procedures :- Implemented procedures streamline ticket sales management including waitlisting for sold out events.
4. Real-time Analytics :- Provided SQL queries offer instant insights into sales, event attendance, and revenue.
5. Automated Triggers :- Implemented triggers update event capacity and notify organizers when ticket thresholds are reached.