① Event management system with Real-Time Analysis
Design a database for manging 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 trigers to automatically update event capacity and notify event organizers when a threeshold of tickets is reached.

1. Conceptual ER diagram :-

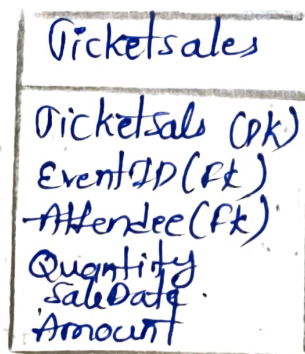The conceptual ER diagram provides a high level overview of the main enities, and their relationships in the system.

Event	Venue	Attendee
EventID	Venue ID	Attendee ID
Name	Name	Name
Date	Location	Contact INfo

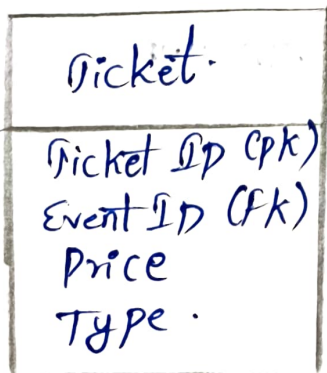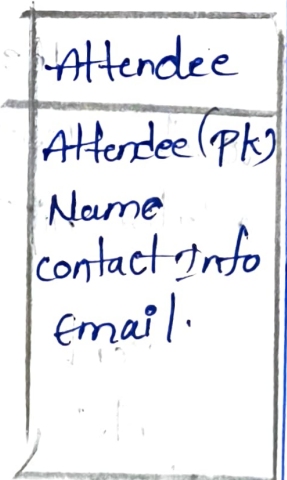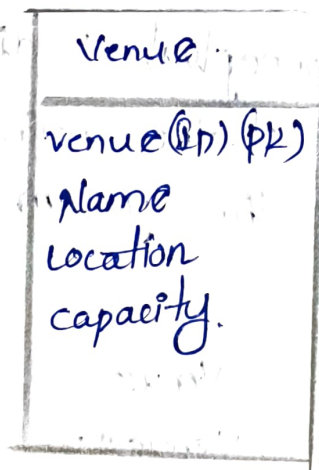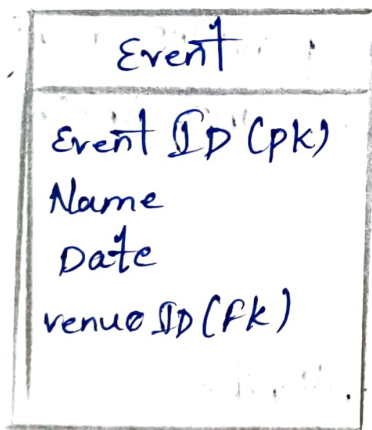| Ticket	Ticket sales
Ticket ID	Event ID
Event ID	Attendee Ip
Price	sales Date
Type	Quantity
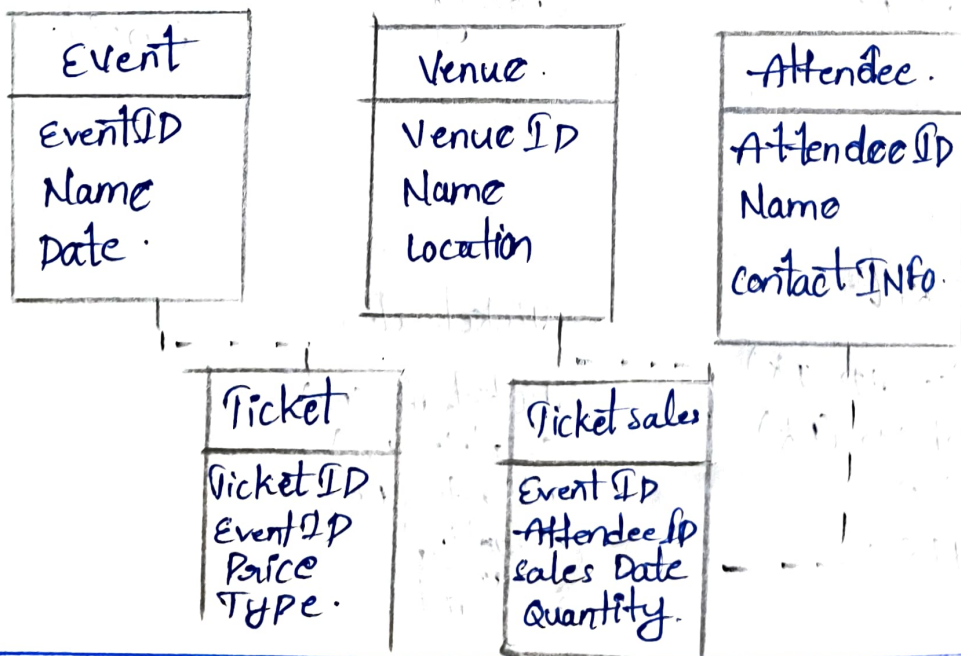
Conceptual Model Explanation :
* Event : Represents each event manged by system.
* venue : Represents the venue where the event place.
* Attendee : Represents indivials attending events.
* Ticket : Represents the type and prices of tickets for event
* Ticketsale : Represents ticket sales and transactions for each attendee

2. Logical ER diagram :-
   The logical ER diagram adds som more detail. including primary key foreign key and attributes that specify entify relationships.

Event
Event ID (pk)
Name
Date
venue ID (FK)

Venue
venue (ID) (PK)
Name
Location
capacity.

Attendee
Attendee (PK)
Name
contact Info
Email.

Ticket.
Ticket Ip (pk)
Event ID (FK)
Price
Type .

Ticketsales
Ticketsals (pk)
EventID (Ft)
Attendee (FK)
Quantity
SaleDate.
Amount

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.

Event.
EventID (pk)
Name VARCHAR
Date DATE
venue ID (fk)

Venue
venue ID (pk)
Name VARCHAR
Location VARCHAR
Capacity INT

Attendee (pk):
Name
contact INFO
Email VARCHAR

Ticket
Ticket (pk) INT
EventID (fk) INT
Price Decimal (m,p)
Type VARCHAR

Ticketsale
Ticketsale (pk)
EventID (fk) INT
Attendee (fk) INT
Quantity INT
Sale Date DATE
Amount Decimal (10,4)

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 :-

```sql
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)
);
```

```sql
CREATE TABLE Tickets (
    Ticket ID INT Primary key,
    Event ID INT,
    Attendee ID INT,
    Ticket type VARCHAR (100),
    Ticket status VARCHAR (100),
    Purchase Date DATE,
    purchase Time TIME,
    price paid Decimal (10,2),
    foreign key (Event ID) References
Events (Event ID),
    FOREIGN KEY (Attendee ID), REFERENCES
Attendees (Attendee ID)
);
CREATE TABLE Attendees (
    AttendeeID INT Primary key,
    first Name VARCHAR (100),
    Last Name VARCHAR (100),
    Email VARCHAR (225)
    phone VARCHAR (20)
);
```

```sql
CREATE TABLE Waitlist (
Waitlist INT PRIMARY KEY,
EventID INT,
AttendeeID INT,
waitlist Date DATE,
waitlist Time TIME,
FOREIGN KEY (AttendeeID) REFERENCES
Attendees (AttendeeID)
);
```