**ANP-C7971**

**ANP-C7971**

Student\_Id: **AF0401168**

Student\_Name: **K.Dhanalaxmi**

Database Concert Management System

Student\_Id: **AF0401168**

Student\_Name: **K.Dhanalaxmi**

Database Concert Management System

Concert Management System

**1. Introduction**

A **Concert Management System (CMS)** is a software solution designed to streamline and automate various tasks involved in managing concerts. Its primary objectives include efficient concert organization, managing venue details, artist bookings, ticketing, and customer interaction. CMS serves as a centralized platform for event organizers to manage concert information, artist performances, venues, ticket sales, and customer details.

The system allows organizers to plan and schedule concerts, manage artist bookings, sell tickets, and track attendee information. It ensures seamless concert management through a user-friendly platform, providing essential tools to manage the complexities of concert events, ticketing, and artist performances.

1. Entities:

* Concert
* Venue
* Artist
* Organizer
* Attendee
* Ticket

2. Entity Relationships:

1. **Concert** ↔ **Venue**: One-to-One (concert.venue\_id → venue.venue\_id)
2. **Concert** ↔ **Artist**: One-to-Many (artist.concert\_id → concert.concert\_id)
3. **Concert** ↔ **Organizer**: One-to-One (organizer.concert\_id → concert.concert\_id)
4. **Concert** ↔ **Ticket**: One-to-Many (ticket.concert\_id → concert.concert\_id)
5. **Attendee** ↔ **Ticket**: One-to-Many (attendee.ticket\_id → ticket.ticket\_id)
6. **Ticket** ↔ **Payment**: One-to-One (not present, needs payment.ticket\_id → ticket.ticket\_id)
7. **Concert** ↔ **Attendee**: Many-to-Many (Indirect via Ticket)
8. **Artist** ↔ **Booking**: Many-to-One (not present, needs booking.artist\_id and booking.concert\_id)

Attributes:

**1. Concert**

* + concert\_id: INT, Primary Key
  + concert\_name: VARCHAR(20)
  + date: DATE
  + venue\_id: INT, Foreign Key (References venue.venue\_id)

**2. Venue**

* + venue\_id: INT, Primary Key
  + venue\_name: VARCHAR(20)
  + location: VARCHAR(20)
  + capacity: INT
  + ticket: INT (Price of a standard ticket)

**3. Artist**

* + artist\_id: INT, Primary Key
  + concert\_id: INT, Foreign Key (References concert.concert\_id)
  + artist\_name: VARCHAR(20)
  + genre: VARCHAR(20)
  + contact\_info: BIGINT
  + venue: VARCHAR(20) (Specific venue where the artist performs, if applicable)

**4. Organizer**

* + organizers\_id: INT, Primary Key
  + concert\_id: INT, Foreign Key (References concert.concert\_id)
  + organizer\_name: VARCHAR(20)
  + location: VARCHAR(20)
  + contact\_info: BIGINT

**5. Ticket**

* + ticket\_id: INT, Primary Key
  + concert\_id: INT, Foreign Key (References concert.concert\_id)
  + price: INT
  + availability: BOOLEAN (Indicates whether the ticket is available)

**6. Attendee**

* + attendee\_id: INT, Primary Key
  + attendee\_name: VARCHAR(20)
  + attendee\_email: VARCHAR(20)
  + contact\_info: BIGINT
  + ticket\_id: INT, Foreign Key (References ticket.ticket\_id)

Entity Relationship Diagram of Concert Management System

By

concert

Organizer

Takes

Ticket

Provide

Has

Venue

Attendee

Artist

Consist

