

Database design for music event management:

Entities identified:

Event, Customer, Concert, Band, Members, Album, Agent, Contract, Transaction(Supertype), Tickets(Subtype), Album sale(Subtype), band_receipt(Subtype), Venue

Entity set description:

Event: We assume the role of an agency that organizes shows/events

Event
Event_id (PK) Event_name Start_time End_time

Customer :

Customer
Event_id (FK) Ticket_id(FK) Customer_name Contact_number

Concert - An event can have multiple concerts

Concert
Concert_id (PK) Concert_name Start_time End_time Event_id(FK) Band_id(FK)

Band: Creating a separate entity because each event may have one or more featured bands.

Band
Band_id (PK) Band_name No_of_artists Agent_id(FK)

Members - A band can have one or many performers

Members
Member_id (PK) Member_name genre/instrument Contact number Band_id(FK)

Album : A band may release an album at a show. A band may promote/sell albums at a show.
(you want to track these because you get a cut of the revenues).

Album
Album_id (PK) Album_name release_date Band_id(FK)

Agent : You contract with bands through their agents. You pay the agents based on invoices they send.

Agent
Agent_id (PK) Agent_name Contact number

Contract:

Contract
Contract_id (PK) No of shows Agent_id(FK) Band_id(FK)

Invoice - Connecting invoice to both contract and concert.

Invoice
Invoice_id (PK) Amount to be paid Contract_id(FK) Concert_id(FK)

Creating a transaction supertype which connects to events and customers.

Transaction
Transaction_id(PK) Event_id (FK) Transaction_date

Subtypes of transaction - Album sale and tickets. Album sale connects to entity album

Album_sale
Album_id (FK) no_of_copies_sold Album_unit_price

Tickets : You sell tickets to the Events.

Tickets
Ticket_id (PK) Seat_no Ticket_price

Band receipt: Amount paid to band for each concert

Band receipt
Receipt_id (PK) Invoice_id(FK) Amount paid

Venue:

Venue
Venue_id (PK) Venue_name Address Hourly_rent Event_id(FK)

ER diagram

