EventSphere Event Management Database Normalization Steps

Entities in the Database:

1. **Event:** `Event_ID` (PK), `Event_Type_ID` (FK), `Organization_ID`(FK), `Venue_ID` (FK), `Estimated_Budget`, `Total_Expenditure`, `Description`, `Start_Date`, `End_Date`, `Status`, `Estimated_Attendance`, `Actual_Attendance`
2. **Venue:** `Venue_ID` (PK), `Capacity`, `Address_Line`, `City`, `State`, `Postal_Code`, `Country, Online_Flag`
3. **Attendee:** `Attendee_ID` (PK), `First_Name`, `Last_Name`, `Email`, `Phone`
4. **Employee:** `Employee_ID` (PK), `Organization_ID` (FK), `First_Name`, `Last_Name`, `Job_Title`, `Email`
5. **Organization:** `Organization_ID` (PK), `Name`, `Contact_Person`, `Phone`, `Email`
6. **Partner:** `Partner_ID` (PK), `Name`, `Email`, `Phone`
7. **Event_Type:** `Event_Type_ID` (PK), `Event_Type_Name`

Relations Between Entities:

- **Event** with **Event_Type**: Many-to-One (M:1)
- **Event** with **Venue**: Many-to-One (M:1)
- **Event** with **Organization**: Many-to-One (M:1)
- **Employee** with **Organization**: Many-to-One (M:1)
- **Event** with **Partner**: Many-to-Many (M:M)
- **Event** with **Employee**: Many-to-Many (M:M)
- **Ticket** with **Attendee**: Many-to-Many (M:M))

Additional Junction Tables for Many-to-Many Relationships:

1. **Event_Partner:** `Event_ID` (FK), `Partner_ID` (FK), `Role`
2. **Event_Ticket_Assignment:** `Ticket_ID` (PK), `Attendee_ID` (FK), Event_ID (FK), `Purchase_date`, `Expiry_date, Price, Ticket_Type`

3. **Event_Employee:** `Event_ID` (FK), `Employee_ID` (FK), `Task`, `Start_date`, `Deadline`, `Task_completed`

Step 1: First Normal Form (1NF)

- **Objective:** Ensure that each field contains only atomic values (no sets or lists) and that each record is uniquely identifiable.
- **Implementation:**
  - Every table contains rows where each attribute has a single value.
  - Each table has a primary key (`PK`) that uniquely identifies each record:
    - **Event:** `Event_ID`
    - **Venue:** `Venue_ID`
    - **Attendee:** `Attendee_ID`
    - **Employee:** `Employee_ID`
    - **Organization:** `Organization_ID`
    - **Partner:** `Partner_ID`
    - **Event_Type:** `Event_Type_ID`
    - **Event_Partner:** Composite key (`Event_ID`, `Partner_ID`)
    - **Event_Ticket_Assignment:** `Ticket_ID` (Even though `Attendee_ID` and `Event_ID` can uniquely identify a record, the `Ticket_ID` is used as a Primary Key in this junction table for easy joining and calculations. The foreign keys are kept as well for analysis purposes.)
    - **Event_Employee:** Composite key (`Event_ID`, `Employee_ID`)

Step 2: Second Normal Form (2NF)

- **Objective:** Ensure that all non-key attributes are fully dependent on the primary key. This step is especially important for tables with composite primary keys.
- **Implementation:**
  - For each table, every non-key attribute is fully functionally dependent on the primary key.

    For example:

    - In the **Event** table, all fields like `Estimated_Budget`, `Description`, `Status`, etc., are fully dependent on `Event_ID`.
    - In the **Ticket** table, `Price` and `Ticket_Type` are fully dependent on `Ticket_ID`.

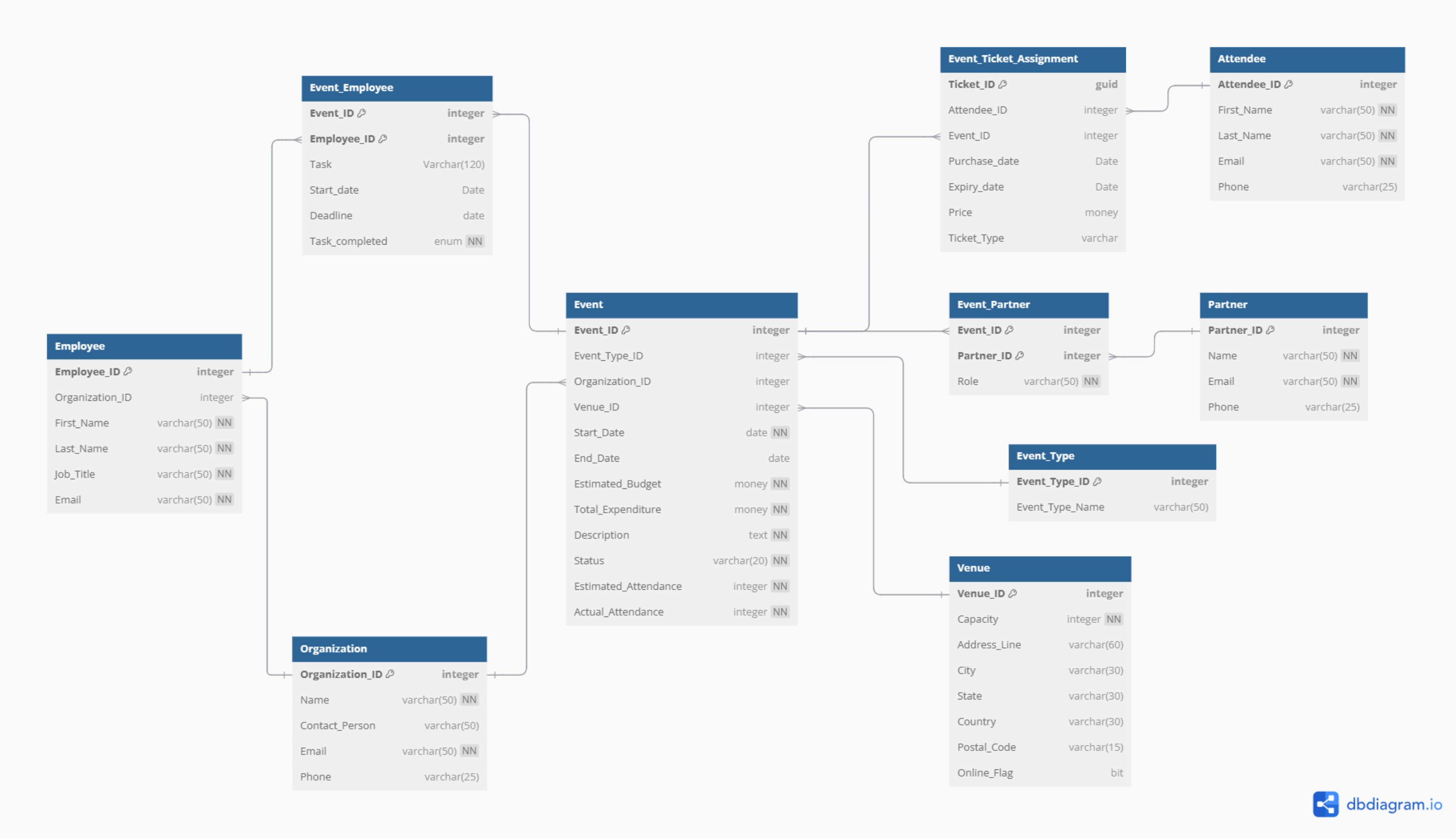
    - For composite key tables (like `Event_Partner`, `Event_Employee`), the non-key attributes (like `Role`, `Purchase_date`, etc.) are dependent on the full composite key.

Step 3: Third Normal Form (3NF)

- **Objective:** Ensure that all attributes are dependent only on the primary key and not on other non-key attributes (no transitive dependencies).
- **Implementation:**
    - **Event Table:**
        - All fields such as `Event_Type_ID`, `Estimated_Budget`, `Start_Date`, etc., are directly related to `Event_ID` with no transitive dependencies. Thus, the **Event** table is in 3NF.
    - **Venue Table:**
        - Attributes like `Capacity`, `Address_Line`, `City`, etc., are directly dependent on `Venue_ID`. Therefore, the **Venue** table is in 3NF.
    - **Attendee Table:**
        - All fields like `First_Name`, `Email`, etc., are dependent only on `Attendee_ID`, ensuring the **Attendee** table is in 3NF.
    - **Event_Ticket_Assignment Table:**
        - Attributes such as `Price`, `Event_ID`, and `Ticket_Type` are all dependent on `Ticket_ID`, so the **Event_Ticket_Assignment** table is in 3NF. (In principle, we can separate `Ticket_Type` into a different table which will uniquely identify each `Ticket_Type` and can be used as a foreign key in this table. However, for reporting purposes and to keep the database design simple, I've kept the `Ticket_Type` in the **Ticket** table.)
    - **Employee Table:**
        - Fields like `First_Name`, `Job_Title`, etc., are all directly dependent on `Employee_ID`, ensuring the **Employee** table is in 3NF.
    - **Organization Table:**
        - Fields like `Name`, `Contact_Person`, and `Email` are all directly dependent on `Organization_ID`. The **Organization** table is in 3NF.
    - **Partner Table:**
        - Fields like `Name`, `Email`, and `Phone` are all directly dependent on `Partner_ID`, ensuring the **Partner** table is in 3NF.

- **Event_Type Table:**
  - The field `Event_Type_Name` is directly dependent on `Event_Type_ID`, so the **Event_Type** table is in 3NF.
- **Event_Partner and Event_Employee Tables:**
  - These junction tables have composite keys with no non-key attributes (or attributes that are fully dependent on the composite keys), so they are already in 3NF.

By following these steps, the database schema is normalized up to the Third Normal Form (3NF), ensuring that data redundancy is minimized and data integrity is maintained across all tables.



Event_Employee		
Event_ID	**integer**	
Employee_ID	**integer**	
Task	Varchar(120)	
Start_date	Date	
Deadline	date	
Task_completed	enum	NN

Employee		
Employee_ID	**integer**	
Organization_ID	integer	
First_Name	varchar(50)	NN
Last_Name	varchar(50)	NN
Job_Title	varchar(50)	NN
Email	varchar(50)	NN

Organization		
Organization_ID	**integer**	
Name	varchar(50)	NN
Contact_Person	varchar(50)	
Email	varchar(50)	NN
Phone	varchar(25)	

Event		
Event_ID	**integer**	
Event_Type_ID	integer	
Organization_ID	integer	
Venue_ID	integer	
Start_Date	date	NN
End_Date	date	
Estimated_Budget	money	NN
Total_Expenditure	money	NN
Description	text	NN
Status	varchar(20)	NN
Estimated_Attendance	integer	NN
Actual_Attendance	integer	NN

Event_Ticket_Assignment		
Ticket_ID	**guid**	
Attendee_ID	integer	
Event_ID	integer	
Purchase_date	Date	
Expiry_date	Date	
Price	money	
Ticket_Type	varchar	

Attendee		
Attendee_ID	**integer**	
First_Name	varchar(50)	NN
Last_Name	varchar(50)	NN
Email	varchar(50)	NN
Phone	varchar(25)	

Event_Partner		
Event_ID	**integer**	
Partner_ID	**integer**	
Role	varchar(50)	NN

Partner		
Partner_ID	**integer**	
Name	varchar(50)	NN
Email	varchar(50)	NN
Phone	varchar(25)	

Event_Type		
Event_Type_ID	**integer**	
Event_Type_Name	varchar(50)	

Venue		
Venue_ID	**integer**	
Capacity	integer	NN
Address_Line	varchar(60)	
City	varchar(30)	
State	varchar(30)	
Country	varchar(30)	
Postal_Code	varchar(15)	
Online_Flag	bit	

Final Database Design