CASE STUDY:

Event Management Database Design

EVENTSPHERE

SOLUTIONS

OPTIMIZING EVENT PLANNING AND EXECUTION THROUGH DATA-DRIVEN SOLUTIONS

Presented By
Arpita Deb
September 17, 2024

CONTENTS

ABOUT THE COMPANY

EVENTSPHERE
SOLUTIONS

EventSphere Solutions is a mid-sized corporate Event Management company specializing in organizing internal conferences, product launches, training sessions, and corporate retreats for large enterprises.

The company handles multiple events simultaneously, with each event having its own set of tasks, attendees, and specific requirements. **To streamline operations and ensure all details are properly tracked, EventSphere Solutions needs a centralized database to manage all aspects of their events.**

PROJECT PLAN

EVENTSPHERE SOLUTIONS

1

Business
Objective

Defining the database's purpose, its users and key functions to ensure successful event execution.

2

Schema
Design

Designing the conceptual, logical, and physical database models. Outlining entity relationships, columns, data types, primary and foreign keys to link entities.

3

Database
Creation

Creating the database & tables using SQL Server's Data Definition Language (DDL). Populating them with synthetic yet realistic data.

4

Database
Optimization

Optimizing the tables for faster query & shorter runtime. Automating tasks by adding views, functions and stored procedures.

BUSINESS OBJECTIVE

The primary objective of this event management database is to **provide an efficient, user-friendly system for organizing, tracking, and managing corporate events** for a medium sized company.

DATABASE USE CASES

Centralized
Event Management

Ensuring that all details about corporate events are managed in a single, accessible database, allowing easy retrieval of event information, attendee details, and schedules.

Tracking
Budgets, Attendees & Tasks

Tracking attendee registrations, roles, and participation status to streamline communication and ensure accurate records.
Assigning and tracking the status of tasks to employees of the event organization, ensuring deadlines are met and responsibilities are clearly defined.

Coordinating
With Venues and Partners

Managing venue bookings and ensuring that all necessary equipment and facilities are available and ready for each event. Recording details about partners that provide services for events, such as sponsors, vendors, and marketing partners.

DATABASE USERS

Event Managers

Task Managers

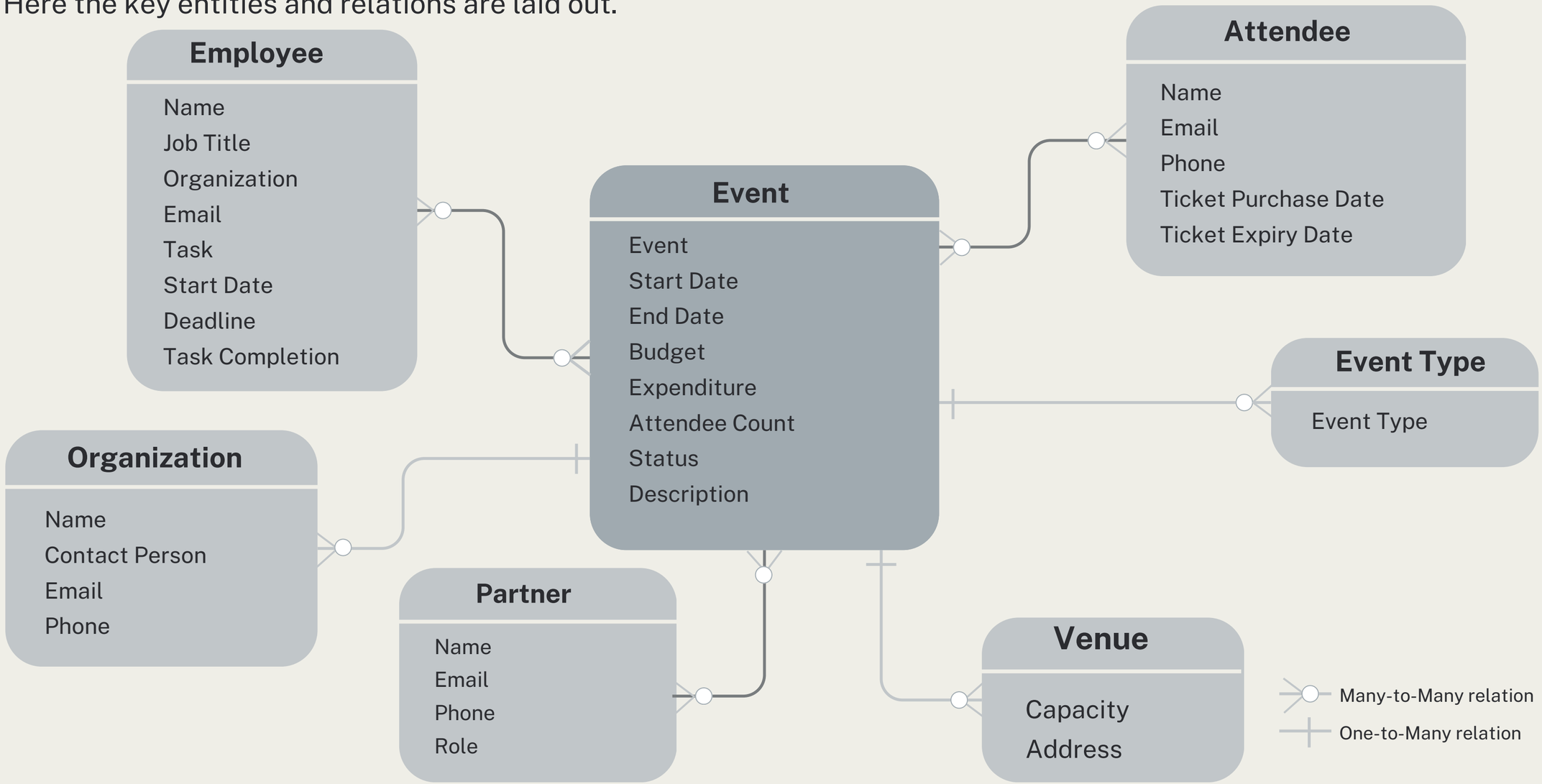
Venue Coordinators

EXECUTIVES

Administrative Staff

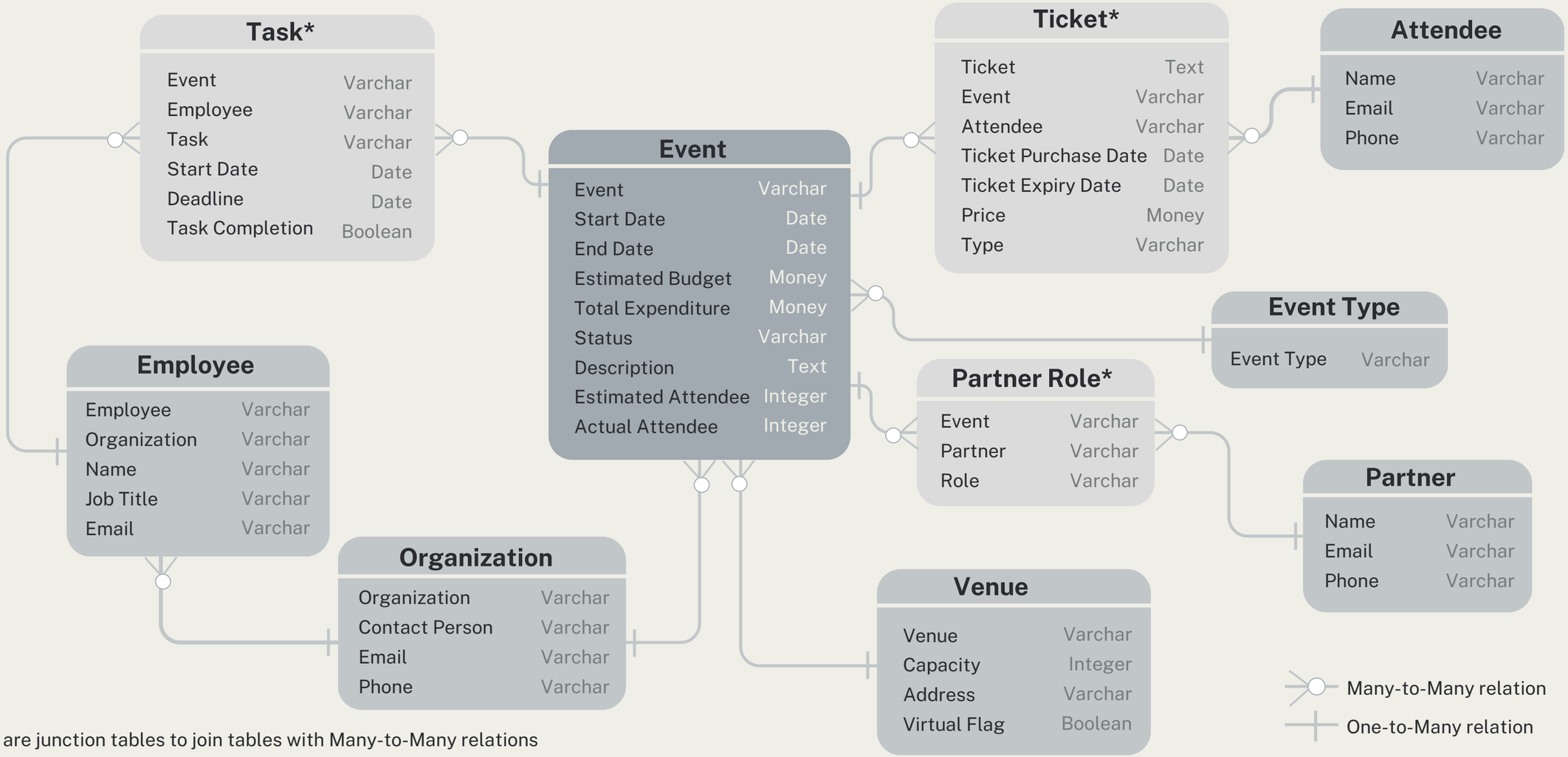
SCHEMA DESIGN: CONCEPTUAL

Here the key entities and relations are laid out.



SCHEMA DESIGN: LOGICAL

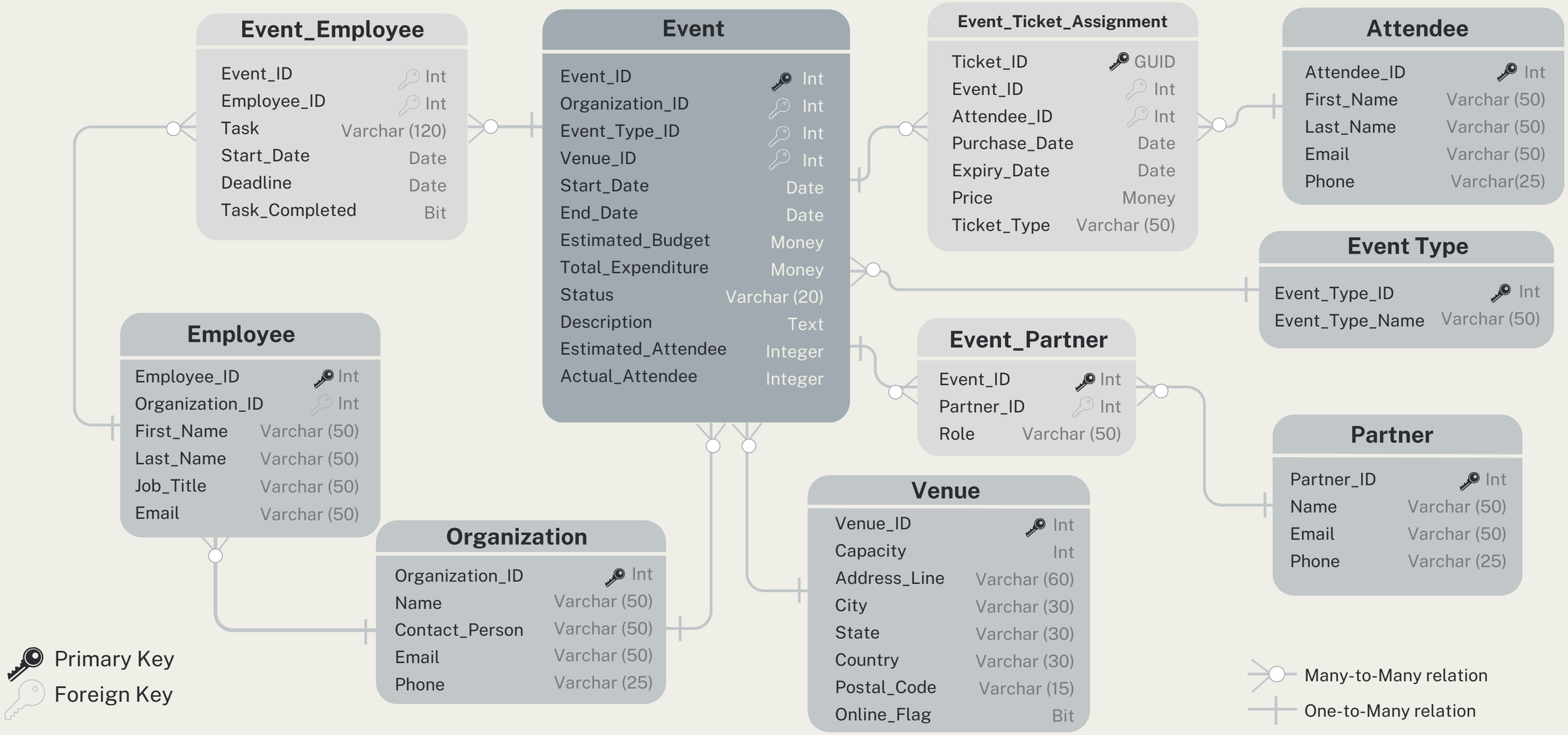
The conceptual database is improved by adding columns and their data types along with junction/linking tables.



* These are junction tables to join tables with Many-to-Many relations

SCHEMA DESIGN: PHYSICAL

The physical database is implemented in SQL Server. Here the Primary and Foreign Keys are defined.



SQL IMPLEMENTATION

Using **Data Definition Language (DDL)**, the EventSphere Database and its leaf tables are created. Here a few code snippets are shown.

```sql
-- Creating the database

CREATE DATABASE [EventSphere Database];
```

```sql
-- Table 2: Venue

CREATE TABLE Venue
(
  venue_ID INT PRIMARY KEY IDENTITY(300,1) NOT NULL,
  capacity INT NOT NULL,
  address_line VARCHAR(60) NULL,
  city VARCHAR(30) NULL,
  state VARCHAR(30) NULL,
  postal_code VARCHAR(15) NULL,
  country VARCHAR(30) NULL
);

ALTER TABLE Venue
ADD OnlineFlag BIT;
```

```sql
-- Table 1:Event

CREATE TABLE Event
(
  event_ID INT PRIMARY KEY IDENTITY(1,1),
  event_Type_ID INT FOREIGN KEY REFERENCES Event_Type(event_Type_ID) NOT NULL,
  organization_ID INT FOREIGN KEY REFERENCES Organization(organization_ID) NOT NULL,
  venue_ID INT FOREIGN KEY REFERENCES Venue(venue_ID) NOT NULL,
  start_date DATE NOT NULL,
  end_date DATE NULL,
  budget_estimated MONEY NULL,
  total_expenditure MONEY NULL,
  description TEXT NULL,
  status VARCHAR(20) NULL,
  estimated_attendance INT NULL,
  actual_attendance INT NULL
);
```

```sql
-- Table 11: Event_Ticket_Assignment junction table

CREATE TABLE Event_Ticket_Assignment
(
    ticket_id UNIQUEIDENTIFIER DEFAULT NEWID() PRIMARY KEY, -- GUID as the primary key
    attendee_id INT FOREIGN KEY REFERENCES Attendee(attendee_id) ,
    event_id INT FOREIGN KEY REFERENCES Event(event_id),
    purchase_date DATETIME NULL,
    expiry_date DATETIME NULL,
    Price MONEY NULL,
    ticket_type VARCHAR(50) NULL
);
```

SQL IMPLEMENTATION

Using **Data Manipulation Language (DML)**, the EventSphere Database and its leaf tables are populated with synthetic data. Here a few code snippets are shown.

```sql
-- 1. Event_Type

INSERT INTO
    [dbo].[Event_Type]([event_type_name])
VALUES
    ('Virtual'),
    ('In-Person'),
    ('Hybrid'),
    ('Conference'),
    ('Workshop'),
    ('Webinar'),
    ('Seminar'),
    ('Trade Show'),
    ('Networking Event'),
    ('Product Launch')
```

```sql
-- 4.Attendee

INSERT INTO [EventSphere Database].dbo.Attendee
(   first_name
    ,last_name
    ,email
    ,phone
)
SELECT  FirstName AS first_name,
        LastName AS last_name,
        EmailAddress AS email,
        PhoneNumber AS phone
FROM [AdventureWorks2022].[Person].[Person] P
JOIN [AdventureWorks2022].[Person].[EmailAddress] E ON P.[BusinessEntityID] = E.[BusinessEntityID]
JOIN [AdventureWorks2022].[Person].[PersonPhone] PH ON P.BusinessEntityID = PH.BusinessEntityID
;
```

```sql
-- 3. Partner

insert into Partner (name, email, phone) values ('Brita MacCurley', 'bmaccurley0@wordpress.com', null);
insert into Partner (name, email, phone) values ('Barron Bathurst', 'bbathurst1@over-blog.com', '+86 747 713 7923');
insert into Partner (name, email, phone) values ('Taffy Crozier', 'tcrozier2@blogspot.com', '+972 387 774 9112');
insert into Partner (name, email, phone) values ('Beatrice Blaxall', 'bblaxall3@163.com', '+86 562 734 1936');
insert into Partner (name, email, phone) values ('Fairleigh Hryniewicki', 'fhryniewicki4@upenn.edu', '+66 628 967 8265');
insert into Partner (name, email, phone) values ('Gillan Hadden', 'ghadden5@cloudflare.com', '+86 843 285 9904');
```

SQL IMPLEMENTATION

EVENTSPHERE SOLUTIONS

Number of records in each tables:

955 rows
Event

10 rows
Event Type

19,972 rows
Attendee

647 rows
Employee

37,107 rows
Event Employee

458 rows
Venue

183 rows
Partner

154 rows
Organization

1,315 rows
Event Partner

396,140 rows
Event Ticket Assignment

SQL IMPLEMENTATION

Using **Data Query Language (DQL)**, performed a few queries on the database table.

```sql
/*
1. What are the upcoming events scheduled for the next month?

Target User: Event Managers, Executives
Objective: Provide a list of events scheduled within a certain date
range to help in tracking and preparing for future events.

*/

SELECT [event_ID],
       [start_date],
       [status]
FROM
    [dbo].[Event]
WHERE
    YEAR([start_date]) = 2024 -- Current year = 2024
    AND MONTH([start_date]) = 10;-- Next Month = October
```

event_ID	start_date	status
2002	2024-10-28	Scheduled
2003	2024-10-24	Scheduled
2006	2024-10-12	NULL
2011	2024-10-26	NULL
2013	2024-10-21	NULL
2019	2024-10-16	NULL
2021	2024-10-02	NULL
2022	2024-10-22	Scheduled
2023	2024-10-24	NULL
2028	2024-10-12	NULL

```sql
/*
11. How many events have been canceled, postponed, or rescheduled in the last year?

Target User: Executives, Event Managers

Objective: Track event cancellations or postponements to assess operational disruptions or changes
in planning.

*/

SELECT Status = [status],
    Event_Count = COUNT(*)
FROM
    [dbo].[Event]
WHERE
    YEAR([start_date]) = 2023
GROUP BY
    [status];
```

	Status	Event_Count
1	Cancelled	14
2	Completed	48
3	Rescheduled	19

```sql
/*
7. What is the overall revenue generated from ticket sales for an event?

Target User: Event Managers, Executives
Objective: Summarize ticket sales revenue to evaluate financial performance for each event.

*/

SELECT [Event_ID],
    Revenue_from_Tickets = SUM([Price])
FROM
    [dbo].[EventDetails]
GROUP BY
    [Event_ID]
ORDER BY
    2 DESC;
```

Event_ID	Revenue_from_Tickets
1394	60600.21
1424	54762.14
2023	53889.38
1164	53852.68
1939	52201.71
1526	51804.92
1624	51630.32
1677	50949.76
1159	50913.69

OPTIMIZATION

EVENTSPHERE SOLUTIONS

Indexes

Added Clustered & Non-clustered indexes in various columns that are most likely to be queried against or joined with other tables, for example:
EventEmployeeID_idx
EventPartnerID_idx
AttendeeName_idx

Views

Created 4 Views in the database system that facilitates easy and quick analysis without repeatedly joining multiple tables. Examples:
dbo.EventDetails
dbo.EmployeePerEvent
dbo.VenueCapacityLimits

Functions & Stored Procedures

Created 3 Stored Procedures and 2 functions for performing advanced analytics and automating tasks. Examples:
dbo.PartnersReport
dbo.TopPerformingTickets
ufn_EventsByEventType(EventType)

CHALLENGES AND SOLUTIONS

EVENTSPHERE SOLUTIONS

Challenge #1: **Due to synthetic nature of the data, inconsistencies may arise during deeper analysis.**

Solution: Refine the data generation process to improve accuracy or apply data cleaning techniques before analysis.

Challenge #2: **Tables like Attendee, Event, or Ticket may not be properly scaled.**

Solution: Adjust the data distribution to better reflect real-world scenarios, ensuring proportional relationships across tables.

Challenge #3: **The database is not fully normalized to keep it simple for analysis.**

Solution: Gradually normalize specific parts of the database where redundancy significantly impacts performance, while maintaining a balance for simplicity.

CONCLUSION

EVENTSPHERE
SOLUTIONS

- In this case study, I **designed and implemented** a database for event management, focusing on key objectives like managing events, attendees, tasks, and reports.
- The process involved **four phases**: database schema design, SQL Server implementation, data population with mock data, and database optimization using indexes, views, and stored procedures.
- The final database is **efficient and realistic**, with all resources, including **SQL scripts** and **documentation**, available on GitHub.

DATABASE FAQ

EVENTSPHERE
SOLUTIONS

Q1. What is Database Normalization?

A: It’s the process of organizing data to reduce data redundancy and improve efficiency.

Q2. What are Primary & Foreign keys?

A: A primary key uniquely identifies each record in a table, while a foreign key links records between tables.

Q3. What is Data Definition Language (DDL)?

A: DDL defines the structure of a database, such as creating, altering, or dropping tables.

Q4. When should we use Views in a database?

A: Use views to simplify complex queries or limit data access for specific users.

Q5. Why use Junction tables?

A: Junction tables manage many-to-many relationships between two tables by linking their primary keys.

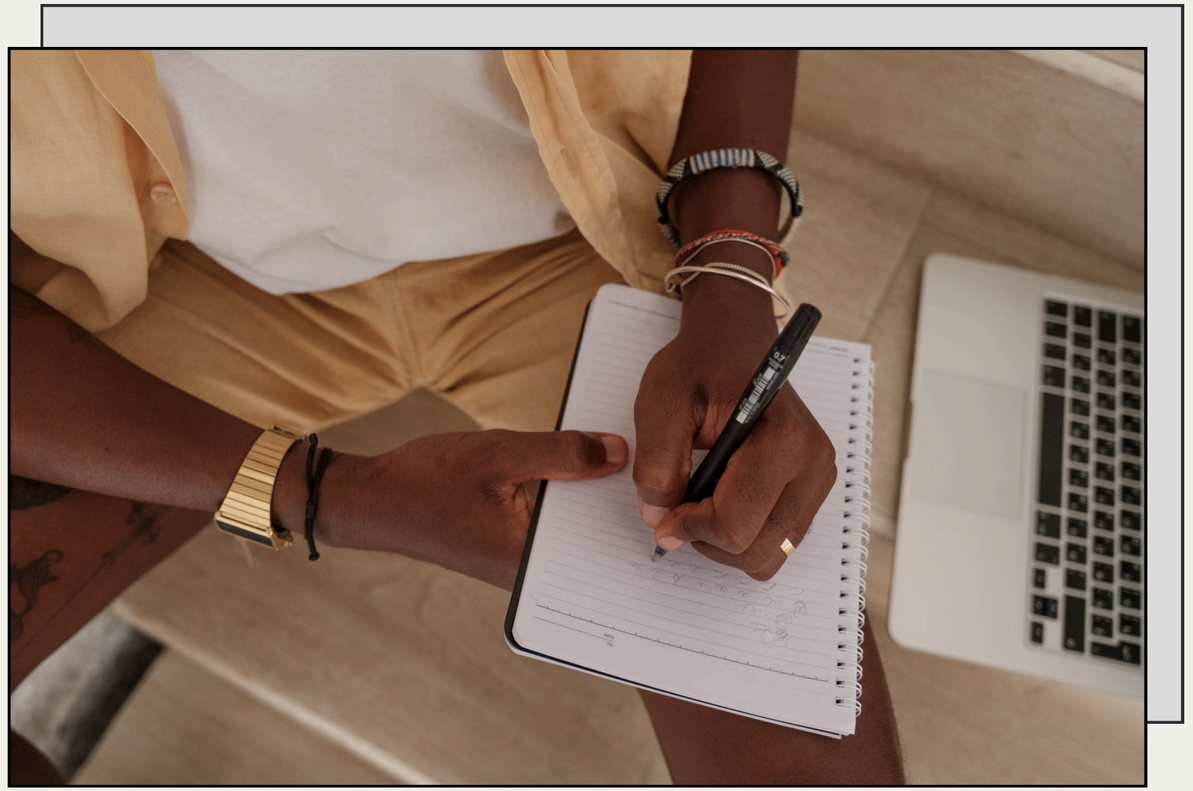
Q6. What is the role of Stored Procedures?

A: Stored procedures are pre-written SQL codes that can be executed to perform tasks, improving performance and consistency.

RESOURCES

EVENTSPHERE
SOLUTIONS

The Project documentation and SQL codes are available on GitHub.



EVENTSPHERE
SOLUTIONS

Thank you!

FOR YOUR TIME

Created By
Arpita Deb
September 17, 2024

LinkedIn
GitHub
Medium