**University of British Columbia, Vancouver**

Department of Computer Science

**CPSC 304 Project Cover Page**

Milestone No: \_\_\_2\_\_\_

Date: \_\_\_\_\_\_2023.10.18\_\_\_\_\_\_

Group Number: \_\_\_\_\_132\_\_\_\_\_

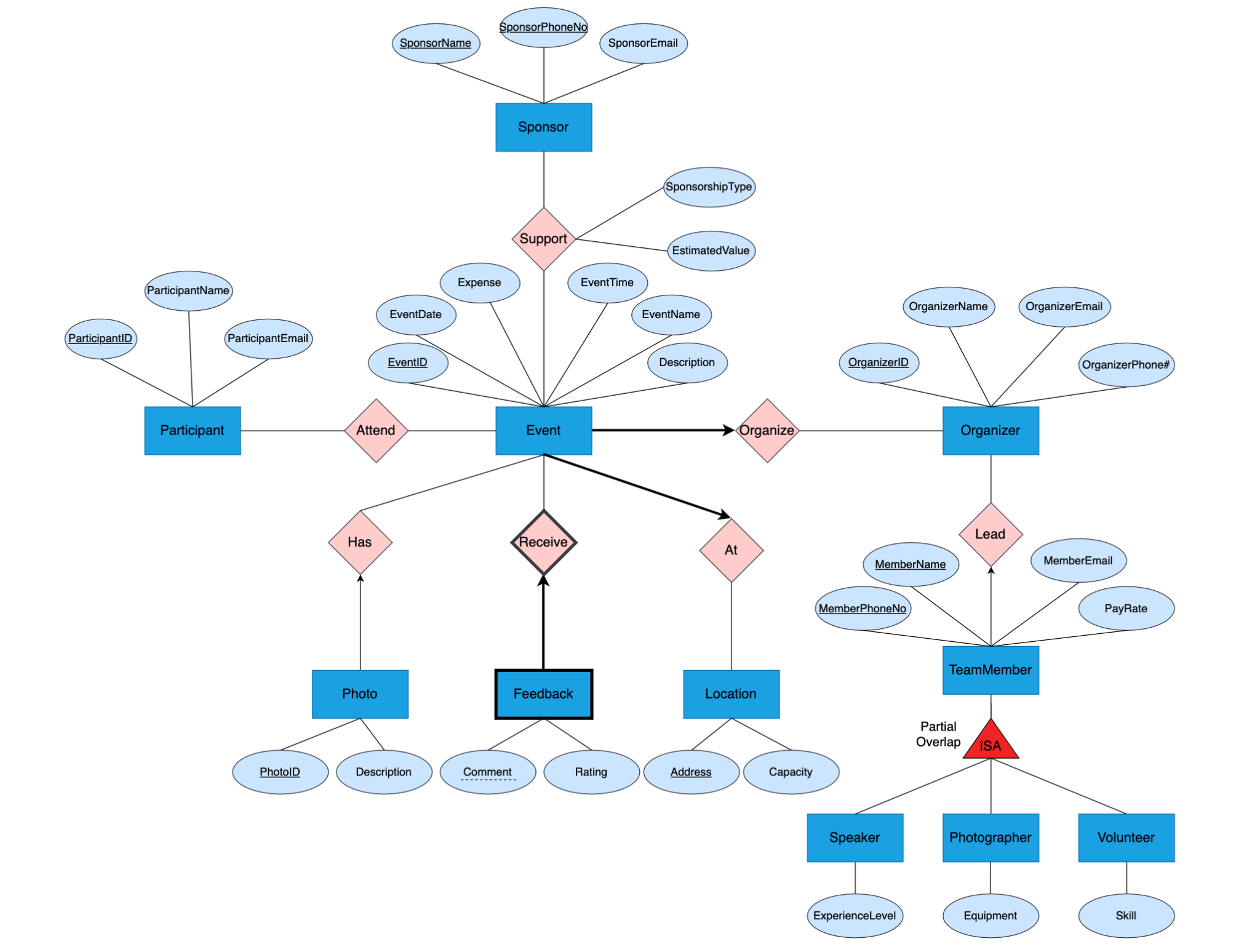
|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Student Number** | **CS Alias (Userid)** | **Preferred E-mail Address** |
| Tianyang Wang | 50797802 | i4d2r | wangti02@student.ubc.ca |
| Howard Sun | 72937774 | y5m8g | sunjiahao1126@gmail.com |
| Jerry Chen | 47063300 | f9c5y | haojunchen2002@gmail.com |

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

1. Project Summary

The UBC Event Collection Application is designed to gather the comprehensive information of an event in the UBC community. It aims to simplify the process of tracking the event-related data. To achieve the goals mentioned above, entities that have crucial information regarding the event are included, thus ensuring that the application effectively collects and stores all the essential data.

2. ER Diagram

Based on the feedback received for our ER diagram from milestone 1, several important changes have been made to improve the clarity and functionality of our database model:

1. **Modification of Primary Keys:** To streamline our database, we've removed some artificial IDs that were previously designated as primary keys (PK). Instead, we've opted to use existing attributes as primary keys wherever possible. However, some frequently used entities still retain an ID as their primary key for practicality.
2. **Entity Naming Clarification:** We've updated the name of the "Staff" entity to "TeamMember." This adjustment aims to reduce potential confusion, as the system now operates under the premise that every event is organized by a leader, represented as "Organizer" in the diagram. Each organizer has their own team of team members.
3. **Attribute Enhancements:** We've added attributes for sub-entities of “TeamMembers”, providing a more comprehensive representation of the data.
4. **Feedback Relationship:** Previously, feedback was associated with participants. We've altered the relationship by connecting feedback directly to events, which should result in a more efficient structure.
5. **Feedback as a Weak Entity:** In our revised design, we've categorized "Feedback" as a weak entity, while "Event" is no longer classified as a weak entity.
6. **Event and Sponsor Relationship:** We've transformed the relationship between "Event" and "Sponsor" into a many-to-many relationship. This alteration permits events to have multiple sponsors and enables sponsors to support multiple events.
7. **Transitioning from Disjoint to Overlap:** We've shifted our approach from "disjoint" to "overlap" for ISA relationships, allowing team members to take on multiple roles within a team.

2. Relational Schema

**a.**

**Organizer**(OrganizerID: integer , OrganizerName: varchar, OrganizerEmail: varchar, OrganizerPhoneNo: char(10))

**Organized\_Event**(EventID: integer, OrganizerID: integer, EventDate: date, Expense: integer, EventTime: char(5), EventName: varchar, Description: varchar)

*// Eventtime in format “HH:MM”*

**Leaded\_TeamMember**(MemberName: varchar, MemberPhoneNo: char(10), OrganizerID: integer, StaffEmail: varchar, PayRate: integer)

**Speaker**(MemberName: varchar, MemberPhoneNo: char(10), ExpericenceLevel: varchar)

**Photographer**(MemberName: varchar, MemberPhoneNo: char(10), Equipment: varchar)

**Volunteer**(MemberName: varchar, MemberPhoneNo: char(10), Skill: varchar)

**Support**(EventID: integer, SponsorName: varchar, SponsorPhoneNo: char(10), SponsorshipType: varchar, EstimatedValue: integer)

**Sponsor**(SponsorName: varchar, SponsorPhoneNo: char(10), SponsorEmail: varchar)

**Event\_Photo**(EventID: integer, PhotoID: integer, Description: varchar)

**Event\_Feedback**(EventID: integer, Comment: varchar, Rating: integer)

**Attend**(EventID: integer, ParticipantID: integer)

**Participant**(ParticipantID: integer, ParticipantName: varchar, ParticipantEmail: varchar)

**Event\_At**(EventID: integer, Address: varchar, EventDate: date, Expense: integer, EventTime: char(5), EventName: varchar, Description: varchar)

**Location**(Address: varchar, Capacity: integer)

**b.**

Organizer(OrganizerID, OrganizerName, OrganizerEmail, OrganizerPhoneNo)

*primary key*: OrganizerID

*candidate key*: OrganizerID, (OrganizerName, OrganizerPhoneNo)

*unique*: (OrganizerName, OrganizerPhoneNo)

Organized\_Event(EventID, **OrganizerID**, EventDate, Expense, EventTime, EventName, Description)

*primary key*: EventID

*foreign key*: OrganizerID references Organizer

*candidate key*: EventID, (OrganizerID**,** EventDate, EventTime)

*not null*: OrganizerID

*unique*: (OrganizerID**,** EventDate, EventTime)

Leaded\_TeamMember(MemberName, MemberPhoneNo, **OrganizerID**, MemberEmail, PayRate)

*primary key*: (MemberName, MemberPhoneNo)

*foreign key*: OrganizerID references Organizer.

*candidate key*: (MemberName, MemberPhoneNo)

Speaker(**MemberName**, **MemberPhoneNo**, ExpericenceLevel)

*primary key*: (MemberName, MemberPhoneNo)

*foreign key*: (MemberName, MemberPhoneNo) reference Leaded\_TeamMember.

*candidate key*: (MemberName, MemberPhoneNo)

Photographer(**MemberName**, **MemberPhoneNo**, Equipment)

*primary key*: (MemberName, MemberPhoneNo)

*foreign key*: (MemberName, MemberPhoneNo) reference Leaded\_TeamMember.

*candidate key*: (MemberName, MemberPhoneNo)

Volunteer(**MemberName**, **MemberPhoneNo**, Skill)

*primary key*: (MemberName, MemberPhoneNo)

*foreign key*: (MemberName, MemberPhoneNo) reference Leaded\_TeamMember.

*candidate key*: (MemberName, MemberPhoneNo)

Support(**EventID**, **SponsorName**, **SponsorPhoneNo**, SponsorshipType, EstimatedValue)

*primary key*: (EventID, SponsorName, SponsorPhoneNo)

*foreign key*: EventID references Organized\_Event, (SponsorName, SponsorPhoneNo) references Sponsor

*candidate key*: (EventID, SponsorName, SponsorPhoneNo)

Sponsor(SponsorName, SponsorPhoneNo, SponsorEmail)

*primary key*: (SponsorName, SponsorPhoneNo)

*candidate key*: (SponsorName, SponsorPhoneNo)

Event\_Photo(**EventID**, PhotoID, Description)

*primary key*: PhotoID

*foreign key*: EventID references Organized\_Event

*candidate key*: PhotoID

Event\_Feedback(**EventID**, Comment, Rating)

*primary key*: (EventID, Comment)

*foreign key*: EventID references Organized\_Event

*candidate key*: (EventID, Comment)

*not null*: EventID

Attend(**EventID, ParticipantID**)

*primary key*: (EventID, ParticipantID)

*foreign key*: EventID references Organized\_Event, ParticipantID references Participant

*candidate key*: (EventID, ParticipantID)

Participant(ParticipantID, ParticipantName, ParticipantEmail)

*primary key*: ParticipantID

*candidate key*: (ParticipantName, ParticipantEmail), ParticipantID

*unique*: (ParticipantName, ParticipantEmail)

Event\_At(**EventID**, **Address**, EventDate, Expense, EventTime, EventName, Description)

*primary key*: EventID

*foreign key*: EventID references Organized\_Event, Address references Location

*candidate key*: (Address, EventDate, EventTime), EventID

*not null*: Address

*unique*: (Address, EventDate, EventTime)

Location(Address, Capacity)

*primary key*: Address

*candidate key*: Address

4. Functional Dependencies (FDs)

Organizer(OrganizerID, OrganizerName, OrganizerEmail, OrganizerPhoneNo):

* OrganizerID -> OrganizerName, OrganizerEmail, OrganizerPhoneNo
* OrganizerEmail -> OrganizerName
* OrganizerName, OrganizerPhoneNo -> OrganizerID

*// The work email address should include the name in some form.*

Organized\_Event(EventID, **OrganizerID**, EventDate, Expense, EventTime, EventName, Description)

* EventID -> OrganizerID, EventDate, Expense, EventTime, EventName, Description
* OrganizerID**,** EventDate, EventTime -> EventID, Expense, EventName, Description
* EventName -> Description

Leaded\_TeamMember(MemberName, MemberPhoneNo, **OrganizerID**, MemberEmail, PayRate)

* MemberName, MemberPhoneNo -> OrganizerID, MemberEmail, PayRate
* MemberEmail -> MemberName

Speaker(**MemberName**, **MemberPhoneNo**, ExpericenceLevel)

* MemberName, MemberPhoneNo -> ExpericenceLevel

Photographer(**MemberName**, **MemberPhoneNo**, Equipment)

* MemberName, MemberPhoneNo -> Equipment

Volunteer(**MemberName**, **MemberPhoneNo**, Skill)

* MemberName, MemberPhoneNo -> Skill

Support(**EventID**, **SponsorName**, **SponsorPhoneNo**, SponsorshipType, EstimatedValue)

* EventID, SponsorName, SponsorPhoneNo -> SponsorshipType, EstimatedValue

Sponsor(SponsorName, SponsorPhoneNo, SponsorEmail)

* SponsorName, SponsorPhoneNo -> SponsorEmail

Event\_Photo(**EventID**, PhotoID, Description)

* PhotoID -> EventID, Description

Event\_Feedback(**EventID**, Comment, Rating)

* EventID, Comment -> Rating

Attend(**EventID, ParticipantID**)

Participant(ParticipantID, ParticipantName, ParticipantEmail)

* ParticipantID -> ParticipantName, ParticipantEmail
* ParticipantName, ParticipantEmail -> ParticipantID

Event\_At(EventID, **Address**, EventDate, Expense, EventTime, EventName, Description)

* EventID ->Address, EventDate, Expense, EventTime, EventName, Description
* Address, EventDate, EventTime -> EventID, Expense, EventName, Description
* EventName -> Description

Location(Address, Capacity)

* Address -> Capacity

5. Normalization

Organizer(OrganizerID, OrganizerName, OrganizerEmail, OrganizerPhoneNo)

* OrganizerID -> OrganizerName, OrganizerEmail, OrganizerPhoneNo
* OrganizerEmail -> OrganizerName
* OrganizerName, OrganizerPhoneNo -> OrganizerID

Key for the table:

OrganizerID+ =(OrganizerID, OrganizerName, OrganizerEmail, OrganizerPhoneNo)

OrganizerID ->OrganizerName, OrganizerEmail,OrganizerPhoneNo in 3NF

OrganizerEmail -> OrganizerName not in 3NF

OrganizerName,OrganizerPhoneNo -> OrganizerID in 3NF

Find the minimal cover:

1. OrganizerID -> OrganizerName

OrganizerID -> OrganizerEmail

OrganizerID -> OrganizerPhoneNo

OrganizerEmail -> OrganizerName

OrganizerName, OrganizerPhoneNo -> OrganizerID

1. No LHS can be minimized for all FD.
2. Delete the OrganizerID -> OrganizerName as this FD is redundant.

Minimal cover:

OrganizerID -> OrganizerEmail,OrganizerPhoneNo

OrganizerEmail -> OrganizerName

OrganizerName,OrganizerPhoneNo -> OrganizerID

Decompose using lossless join:

Organizer(OrganizerID, **OrganizerEmail**, OrganizerPhoneNo)

Organizer\_EmailandName(OrganizerEmail, OrganizerName)

Organized\_Event(EventID, **OrganizerID**, EventDate, Expense, EventTime, EventName, Description)

* EventID -> OrganizerID, EventDate, Expense, EventTime, EventName, Description
* OrganizerID**,** EventDate, EventTime -> EventID, Expense, EventName, Description
* EventName -> Description

Key for the table:

EventID+ = (EventID, OrganizerID, EventDate, Expense, EventTime, EventName, Description)

(OrganizerID**,** EventDate, EventTime)+ = (EventID, OrganizerID, EventDate, Expense, EventTime, EventName, Description)

EventID -> OrganizerID, EventDate, Expense, EventTime, EventName, Description in 3NF

OrganizerID**,** EventDate, EventTime -> EventID, Expense, EventName, Description in 3NF

EventName -> Description not in 3NF

Find the minimal cover:

1. EventID -> OrganizerID

EventID -> EventDate,

EventID -> Expense

EventID -> EventTime

EventID -> EventName

EventID -> Description

OrganizerID**,** EventDate, EventTime -> EventID

OrganizerID**,** EventDate, EventTime -> Expense

OrganizerID**,** EventDate, EventTime -> EventName

OrganizerID**,** EventDate, EventTime -> Description

EventName -> Description

1. No LHS can be minimized for all FD.
2. Delete OrganizerID**,** EventDate, EventTime -> Description as this is redundant

Minimal cover:

EventID -> OrganizerID, EventDate, Expense, EventTime, EventName, Description

OrganizerID**,** EventDate, EventTime -> EventID, Expense, EventName

EventName -> Description

Decompose using lossless join:

Organized\_Event(EventID, **OrganizerID**, EventDate, Expense, EventTime, **EventName**)

Event\_NameandDescpription(EventName, Description)

Leaded\_TeamMember(MemberName, MemberPhoneNo, **OrganizerID**, MemberEmail, PayRate)

* MemberName, MemberPhoneNo -> OrganizerID, MemberEmail, PayRate
* MemberEmail -> MemberName

Key for this table:

(MemberName, MemberPhoneNo)+ = (MemberName, MemberPhoneNo, OrganizerID, MemberEmail, PayRate)

MemberName, MemberPhoneNo -> OrganizerID, MemberEmail, PayRate in 3NF

MemberEmail -> MemberName in 3NF

All FDs are in 3NF, no need for normalization.

Speaker(**MemberName**, **MemberPhoneNo**, ExpericenceLevel)

* MemberName, MemberPhoneNo -> ExpericenceLevel

Key for the table: (MemberName, MemberPhoneNo)+ = (MemberName, MemberPhoneNo, ExpericenceLevel)

MemberName, MemberPhoneNo -> ExpericenceLevel in 3NF

All FDs are in 3NF, no need for normalization.

Photographer(**MemberName**, **MemberPhoneNo**, Equipment)

* MemberName, MemberPhoneNo -> Equipment

Key for the table: (MemberName, MemberPhoneNo)+ = (MemberName, MemberPhoneNo, Equipment)

MemberName, MemberPhoneNo -> Equipment in 3NF

All FDs are in 3NF, no need for normalization.

Volunteer(**MemberName**, **MemberPhoneNo**, Skill)

* MemberName, MemberPhoneNo -> Skill

Key for the table: (MemberName, MemberPhoneNo)+ = (MemberName, MemberPhoneNo, Skill)

MemberName, MemberPhoneNo -> Skill in 3NF

All FDs are in 3NF, no need for normalization.

Support(**EventID**, **SponsorName**, **SponsorPhoneNo**, SponsorshipType, EstimatedValue)

* EventID, SponsorName, SponsorPhoneNo -> SponsorshipType, EstimatedValue

Key for the table: (EventID, SponsorName, SponsorPhoneNo)+ = (EventID, SponsorName, SponsorPhoneNo, SponsorshipType, EstimatedValue)

EventID, SponsorName, SponsorPhoneNo -> SponsorshipType, EstimatedValue in 3NF

All FDs are in 3NF, no need for normalization.

Sponsor(SponsorName, SponsorPhoneNo, SponsorEmail)

* SponsorName, SponsorPhoneNo -> SponsorEmail

Key for the table: (SponsorName, SponsorPhoneNo)+ = (SponsorName, SponsorPhoneNo, SponsorEmail)

SponsorName, SponsorPhoneNo -> SponsorEmail in 3NF

All FDs are in 3NF, no need for normalization.

Event\_Photo(**EventID**, PhotoID, Description)

* PhotoID -> EventID, Description

Key for the table: PhotoID+ = (PhotoID, EventID, Description)

PhotoID -> EventID, Description in 3NF

All FDs are in 3NF, no need for normalization.

Event\_Feedback(**EventID**, Comment, Rating)

* EventID, Comment -> Rating

Key for the table: (EventID, Comment)+ = (EventID, Comment Rating)

EventID, Comment -> Rating in 3NF

All FDs are in 3NF, no need for normalization.

Attend(**EventID, ParticipantID**)

No FDs, so the table is already in BCNF.

Participant(ParticipantID, ParticipantName, ParticipantEmail)

* ParticipantID -> ParticipantName, ParticipantEmail
* ParticipantName, ParticipantEmail -> ParticipantID

Key for this table:

ParticipantID+  = (ParticipantID, ParticipantName, ParticipantEmail)

(ParticipantName, ParticipantEmail)+  = (ParticipantID, ParticipantName, ParticipantEmail)

ParticipantID -> ParticipantName, ParticipantEmail in 3NF

ParticipantName, ParticipantEmail -> ParticipantID in 3NF

All FDs are in 3NF, no need for normalization.

Event\_At(EventID, **Address**, EventDate, Expense, EventTime, EventName, Description)

* EventID ->Address, EventDate, Expense, EventTime, EventName, Description
* Address, EventDate, EventTime -> EventID, Expense, EventName, Description
* EventName -> Description

Key for this table:

EventID+ = (EventID, Address, EventDate, Expense, EventTime, EventName, Description)

(Address, EventDate, EventTime)+ = (EventID, Address, EventDate, Expense, EventTime, EventName, Description)

EventID -> Address, EventDate, Expense, EventTime, EventName, Description in 3NF

Address, EventDate, EventTime -> EventID, Expense, EventName, Description in 3NF

EventName -> Description not in 3NF

Find the minimal cover:

EventID -> Address

EventID -> EventDate,

EventID -> Expense

EventID -> EventTime

EventID -> EventName

EventID -> Description

Address**,** EventDate, EventTime -> EventID

Address**,** EventDate, EventTime -> Expense

Address**,** EventDate, EventTime -> EventName

Address**,** EventDate, EventTime -> Description

EventName -> Description

1. No LHS can be minimized for all FD.
2. Delete Address**,** EventDate, EventTime -> Description as this is redundant

Minimal cover:

EventID -> Address, EventDate, Expense, EventTime, EventName, Description

Address**,** EventDate, EventTime -> EventID, Expense, EventName

EventName -> Description

Decompose using lossless join:

Event\_At(EventID, **Address**, EventDate, Expense, EventTime, **EventName**)

Event\_NameandDescpription(EventName, Description)

Location(Address, Capacity)

* Address -> Capacity

Relation table with only 2 attributes is already in BCNF and 3NF

All tables after normalization:

Organizer(OrganizerID: integer, OrganizerName:varchar, **OrganizerEmail**: varchar, OrganizerPhoneNo: char(10))

*primary key*: OrganizerID

*candidate key*: OrganizerID, (OrganizerName, OrganizerPhoneNo)

*foreign key*: OrganizerEmail references OrganizerEmailandName

*not null*: OrganizerEmail

*unique*: (OrganizerName, OrganizerPhoneNo)

Organizer\_EmailandName(OrganizerEmail: varchar, OrganizerName: varchar)

*primary key*: OrganizerEmail

*candidate key*: OrganizerEmail

Organized\_Event(EventID: integer, **OrganizerID**: integer, EventDate: date, Expense: integer, EventTime: char(5), **EventName**: varchar)

*primary key*: EventID

*foreign key*: OrganizerID references Organizer, EventName references Event\_NameandDescpription

*candidate key*: EventID, (OrganizerID**,** EventDate, EventTime)

*not null*: OrganizerID, EventName

*unique*: (OrganizerID**,** EventDate, EventTime)

Event\_NameandDescription(EventName: varchar, Description: varchar)

*primary key*: EventName

*candidate key*: EventName

Leaded\_TeamMember(MemberName: char(10), MemberPhoneNo: char(10), **OrganizerID**: integer, StaffEmail: varchar, PayRate: integer)

*primary key*: (MemberName, MemberPhoneNo)

*foreign key*: OrganizerID references Organizer.

*candidate key*: (MemberName, MemberPhoneNo)

Speaker(**MemberName**: varchar, **MemberPhoneNo**: char(10), ExpericenceLevel: varchar)

*primary key*: (MemberName, MemberPhoneNo)

*foreign key*: (MemberName, MemberPhoneNo) reference Leaded\_TeamMember.

*candidate key*: (MemberName, MemberPhoneNo)

Photographer(**MemberName**: varchar, **MemberPhoneNo**: char(10), Equipment: varchar)

*primary key*: (MemberName, MemberPhoneNo)

*foreign key*: (MemberName, MemberPhoneNo) reference Leaded\_TeamMember.

*candidate key*: (MemberName, MemberPhoneNo)

Volunteer(**MemberName**: varchar, **MemberPhoneNo**: char(10), Skill: varchar)

*primary key*: (MemberName, MemberPhoneNo)

*foreign key*: (MemberName, MemberPhoneNo) reference Leaded\_TeamMember.

*candidate key*: (MemberName, MemberPhoneNo)

Support(**EventID**: integer, **SponsorName**: varchar, **SponsorPhoneNo**: char(10), SponsorshipType: varchar, EstimatedValue: integer)

*primary key*: (EventID, SponsorName, SponsorPhoneNo)

*foreign key*: EventID references Organized\_Event, (SponsorName, SponsorPhoneNo) references Sponsor

*candidate key*: (EventID, SponsorName, SponsorPhoneNo)

Sponsor(SponsorName: varchar, SponsorPhoneNo: char(10), SponsorEmail: varchar)

*primary key*: (SponsorName, SponsorPhoneNo)

*candidate key*: (SponsorName, SponsorPhoneNo)

Event\_Photo(**EventID**: integer, PhotoID: integer, Description: varchar)

*primary key*: PhotoID

*foreign key*: EventID references Organized\_Event

*candidate key*: PhotoID

Event\_Feedback(**EventID**: integer, Comment: varchar, Rating: integer)

*primary key*: (EventID, Comment)

*foreign key*: EventID references Organized\_Event

*candidate key*: (EventID, Comment)

*not null*: EventID

Attend(**EventID**: integer**, ParticipantID**: integer)

*primary key*: (EventID, ParticipantID)

*foreign key*: EventID references Organized\_Event, ParticipantID references Participant

*candidate key*: (EventID, ParticipantID)

Participant(ParticipantID: integer, ParticipantName: varchar, ParticipantEmail: varchar)

*primary key*: ParticipantID

*candidate key*: (ParticipantName, ParticipantEmail), ParticipantID

Event\_At(**EventID**: integer, **Address**: varchar, EventDate: date, Expense: integer, EventTime: char(5), **EventName**: varchar)

*primary key*: EventID

*foreign key*: EventID references Organized\_Event, Address references Location, EventName references Event\_NameandDescpription

*candidate key*: (Address, EventDate, EventTime), EventID

*not null*: Address, EventName

*unique*: (Address, EventDate, EventTime)

Location(Address: varchar, Capacity: integer)

*primary key*: Address

*candidate key*: Address

6. SQL DDL Statements

CREATE TABLE Organizer\_EmailandName (

OrganizerEmail VARCHAR(255) PRIMARY KEY,

OrganizerName VARCHAR(255)

);

CREATE TABLE Event\_NameandDescription (

EventName VARCHAR(255) PRIMARY KEY,

Description VARCHAR(255)

);

CREATE TABLE Organizer (

OrganizerID INTEGER PRIMARY KEY,

OrganizerName VARCHAR(255),

OrganizerEmail VARCHAR(255) NOT NULL,

OrganizerPhoneNo CHAR(10),

UNIQUE (OrganizerName, OrganizerPhoneNo),

FOREIGN KEY (OrganizerEmail) REFERENCES Organizer\_EmailandName(OrganizerEmail)

);

CREATE TABLE Organized\_Event (

EventID INTEGER PRIMARY KEY,

OrganizerID INTEGER NOT NULL,

EventDate DATE,

Expense INTEGER,

EventTime CHAR(5),

EventName VARCHAR(255) NOT NULL,

FOREIGN KEY (OrganizerID) REFERENCES Organizer(OrganizerID),

FOREIGN KEY (EventName) REFERENCES Event\_NameandDescription(EventName)

);

CREATE TABLE Leaded\_TeamMember (

MemberName VARCHAR,

MemberPhoneNo CHAR(10),

OrganizerID INTEGER,

StaffEmail VARCHAR(255),

PayRate INTEGER,

PRIMARY KEY (MemberName, MemberPhoneNo),

FOREIGN KEY (OrganizerID) REFERENCES Organizer(OrganizerID)

);

CREATE TABLE Speaker (

MemberName VARCHAR(255),

MemberPhoneNo CHAR(10),

ExperienceLevel VARCHAR(255),

PRIMARY KEY (MemberName, MemberPhoneNo),

FOREIGN KEY (MemberName, MemberPhoneNo) REFERENCES Leaded\_TeamMember(MemberName, MemberPhoneNo)

);

CREATE TABLE Photographer (

MemberName VARCHAR(255),

MemberPhoneNo CHAR(10),

Equipment VARCHAR(255),

PRIMARY KEY (MemberName, MemberPhoneNo),

FOREIGN KEY (MemberName, MemberPhoneNo) REFERENCES Leaded\_TeamMember(MemberName, MemberPhoneNo)

);

CREATE TABLE Volunteer (

MemberName VARCHAR(255),

MemberPhoneNo CHAR(10),

Skill VARCHAR(255),

PRIMARY KEY (MemberName, MemberPhoneNo),

FOREIGN KEY (MemberName, MemberPhoneNo) REFERENCES Leaded\_TeamMember(MemberName, MemberPhoneNo)

);

CREATE TABLE Sponsor (

SponsorName VARCHAR(255),

SponsorPhoneNo CHAR(10),

SponsorEmail VARCHAR(255),

PRIMARY KEY (SponsorName, SponsorPhoneNo)

);

CREATE TABLE Support (

EventID INTEGER,

SponsorName VARCHAR(255),

SponsorPhoneNo CHAR(10),

SponsorshipType VARCHAR(255),

EstimatedValue INTEGER,

PRIMARY KEY (EventID, SponsorName, SponsorPhoneNo),

FOREIGN KEY (EventID) REFERENCES Organized\_Event(EventID),

FOREIGN KEY (SponsorName, SponsorPhoneNo) REFERENCES Sponsor(SponsorName, SponsorPhoneNo)

);

CREATE TABLE Event\_Photo (

PhotoID INTEGER PRIMARY KEY,

EventID INTEGER,

Description VARCHAR(255),

FOREIGN KEY (EventID) REFERENCES Organized\_Event(EventID)

);

CREATE TABLE Event\_Feedback (

EventID INTEGER,

Comment VARCHAR(255),

Rating INTEGER,

PRIMARY KEY (EventID, Comment),

FOREIGN KEY (EventID) REFERENCES Organized\_Event(EventID)

);

CREATE TABLE Participant (

ParticipantID INTEGER PRIMARY KEY,

ParticipantName VARCHAR(255),

ParticipantEmail VARCHAR(255),

UNIQUE (ParticipantName, ParticipantEmail)

);

CREATE TABLE Attend (

EventID INTEGER,

ParticipantID INTEGER,

PRIMARY KEY (EventID, ParticipantID),

FOREIGN KEY (EventID) REFERENCES Organized\_Event(EventID),

FOREIGN KEY (ParticipantID) REFERENCES Participant(ParticipantID)

);

CREATE TABLE Location (

Address VARCHAR(255) PRIMARY KEY,

Capacity INTEGER

);

CREATE TABLE Event\_At (

EventID INTEGER,

Address VARCHAR(255),

EventDate DATE,

Expense INTEGER,

EventTime CHAR(5),

EventName VARCHAR(255) NOT NULL,

PRIMARY KEY (EventID),

FOREIGN KEY (EventID) REFERENCES Organized\_Event(EventID),

FOREIGN KEY (Address) REFERENCES Location(Address)

);

8. Tuples INSERT statements

INSERT INTO Event\_NameandDescription (EventName, Description)

VALUES

('Tech Expo 2023', 'A cutting-edge technology exposition with industry experts.'),

('Business Mixer', 'A networking event for professionals and entrepreneurs.'),

('Community Charity Gala', 'A grand gala to support local community initiatives.'),

('Product Showcase', 'An exclusive product showcase event for innovation enthusiasts.'),

('International Food Festival', 'A diverse food festival celebrating global cuisines.');

INSERT INTO Organizer\_EmailandName (OrganizerEmail, OrganizerName)

VALUES

('jayson@email.com', 'Jayson Ward'),

('fernanda@email.com', 'Fernanda Hickman'),

('jerry@email.com', 'Jerry Ling'),

('eva@email.com', 'Eva Hendrix'),

('branden@email.com', 'Branden Malone');

INSERT INTO Organizer (OrganizerID, OrganizerEmail, OrganizerPhoneNo)

VALUES

(1, 'jayson@email.com', '1940384920'),

(2, 'fernanda@email.com', '4935948294'),

(3, 'jerry@email.com', '2948203874'),

(4, 'eva@email.com', '8483920584'),

(5, 'branden@email.com', '4893059203');

INSERT INTO Organized\_Event (EventID, OrganizerID, EventDate, Expense, EventTime, EventName)

VALUES

(1, 1, '2023-10-25', 5000, '10:00', 'Tech Expo 2023'),

(2, 2, '2023-11-15', 7500, '14:30', 'Business Mixer'),

(3, 3, '2023-12-02', 3000, '09:00', 'Community Charity Gala'),

(4, 4, '2023-12-10', 6000, '15:00', 'Product Showcase'),

(5, 5, '2023-12-20', 4500, '11:30', 'International Food Festival');

INSERT INTO Leaded\_TeamMember (MemberName, MemberPhoneNo, OrganizerID, StaffEmail, PayRate)

VALUES

('Johnathon Bauer', '3049543029', 1, 'bauer@email.com', 20),

('Beckhan Collins', '2039504394', 2, 'collins@email.com', 18),

('Britney Vasquez', '4873948203', 3, 'vasquez@email.com', 22),

('Allison Orozco', '3590495834', 4, 'orozco@email.com', 19),

('Eden Edwards', '2039305894', 5, 'edwards@email.com', 21)

INSERT INTO Speaker (MemberName, MemberPhoneNo, ExpericenceLevel)

VALUES

(‘Sabrina Rosales', '3828372394', 'Expert'),

(‘Aryana Kirk', '2325837485', 'Intermediate'),

(‘Tristian Fischer', '5845867589', 'Advanced'),

(‘Karia Marsh', '2324385938', 'Expert'),

(‘Mckenzie Walker', '1383948239', 'Intermediate');

INSERT INTO Photographer (MemberName, MemberPhoneNo, Equipment)

VALUES

(‘Brent Schaefer', '2938293849', 'Professional Camera'),

(‘Donald Esparza', '1224038293', 'Digital SLR Camera'),

(‘Coby Pope', '3340594029', 'Point-and-Shoot Camera'),

(‘Reginald Juarez', '4739485798', 'Mirrorless Camera'),

(‘Zackary Leblanc', '2302839589', 'Drone Camera');

INSERT INTO Volunteer (MemberName, MemberPhoneNo, Skill)

VALUES

(‘Brayan Blackwell', '1292849384', 'Event Planning'),

(‘Enrique Estes', '3938203898', 'Catering'),

(‘Molly Brady', '2045849384', 'Decoration'),

(‘Serenity Owens', '5840398909', 'Registration'),

(‘Averie Herring', '7890798768', 'Security');

INSERT INTO Support (EventID, SponsorName, SponsorPhoneNo, SponsorshipType, EstimatedValue)

VALUES

(1, ‘Damarion Kelly', '7398747293', 'Gold', 10000),

(2, ‘Cayden Yates', '1750493059', 'Silver', 7500),

(3, ‘Kassandra Ross', '5948493789', 'Bronze', 5000),

(4, ‘Leandro Hill', '1743985748', 'Platinum', 15000),

(5, ‘Alexa Hutchinson', '2730950943', 'Gold', 11000);

INSERT INTO Sponsor (SponsorName, SponsorPhoneNo, SponsorEmail)

VALUES

('John Smith', '4839283049', 'john.sponsor@email.com'),

('Alice Johnson', '2839584938', 'alice.sponsor@email.com'),

('Robert Davis', '3849283049', 'robert.sponsor@email.com'),

('Emily Wilson', '3940293059', 'emily.sponsor@email.com'),

('Daniel Miller', '8098495748', ‘[daniel.sponsor@email.com](mailto:daniel.sponsor@email.com)');

INSERT INTO Event\_Photo (EventID, PhotoID, Description)

VALUES

(1, 1, 'Capturing the Excitement at Tech Expo 2023'),

(2, 2, 'Networking Moments at the Business Mixer'),

(3, 3, 'Elegant Decorations at the Community Charity Gala'),

(4, 4, 'Unveiling New Products at the Product Showcase'),

(5, 5, 'Exploring Global Flavors at the International Food Festival');

INSERT INTO Event\_Feedback (EventID, Comment, Rating)

VALUES

(1, 'Great event, learned a lot!', 5),

(2, 'Enjoyed the networking opportunities.', 4),

(3, 'Wonderful gala, fantastic cause.', 5),

(4, 'Impressive product showcase!', 4),

(5, 'Delicious food from around the world.', 5);

INSERT INTO Attend (EventID, ParticipantID)

VALUES

(1, 1),

(1, 2),

(2, 3),

(3, 4),

(4, 5);

INSERT INTO Participant (ParticipantID, ParticipantName, ParticipantEmail)

VALUES

(1, 'Thomas Porter', 'thomas@email.com'),

(2, 'Cheyanne Conner, 'cheyanne@email.com'),

(3, 'Izayah Gallagher','izayah@email.com'),

(4, 'Eliza Kaufman', 'eliza@email.com'),

(5, 'Barrett Mcfarland', 'barrett@email.com');

INSERT INTO Event\_At (EventID, Address, EventDate, Expense, EventTime, EventName)

VALUES

(1, '123 Tech Expo Street', '2023-10-25', 5000, '10:00', 'Tech Expo 2023'),

(2, '456 Business Mixer Avenue', '2023-11-15', 7500, '14:30', 'Business Mixer'),

(3, '789 Charity Gala Lane', '2023-12-02', 3000, '09:00', 'Community Charity Gala'),

(4, '101 Product Showcase Road', '2023-12-10', 6000, '15:00', 'Product Showcase'),

(5, '555 Food Festival Drive', '2023-12-20', 4500, '11:30', 'International Food Festival');

INSERT INTO Location (Address, Capacity)

VALUES

('123 Tech Expo Street', 1000),

('456 Business Mixer Avenue', 500),

('789 Charity Gala Lane', 300),

('101 Product Showcase Road', 800),

('555 Food Festival Drive', 1200);