University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: 2

Date: October 20, 2023

Group Number: 128

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Jason Zhu	98960727	n6v7c	zhujason4@gmail.com
Michael Cui	16721946	r4a2e	michaelcui11062003@gmail.com
Leo Wang	46986956	k0j9k	Leowang801@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.)

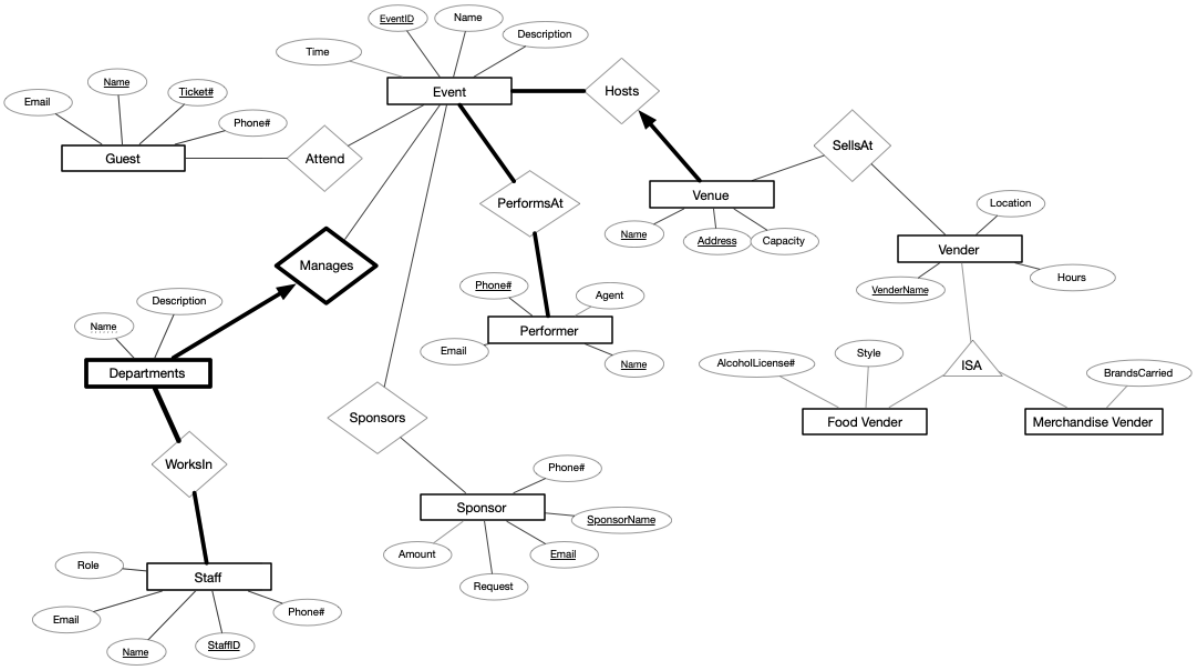
In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

University of British Columbia, Vancouver
Department of Computer Science

2. Project Summary

In our project, we will be creating an event managing platform. It will have ten total entities including Guest, Event, Staff, Department, Sponsor, Performer, Venue, Vendor, Food Vendor, Merchandise Vendor, and the purpose is to manage event planning systems effectively and efficiently.

3. ER Diagram



Vender - Food Vender - Merchandise Vender ISA Constraint:
Partial + Overlap

We incorporated the TA feedback to include constraints for the ISA, because it is needed and we forgot to do this in milestone 1. We also combined name with Phone# for the Performer entity key and also combined SponsorName with Email for the Sponsor key.

4. Schema

Entities

Guest(
Email: VARCHAR,
Name: VARCHAR,
ticket #: INTEGER,
phone #: INTEGER)

University of British Columbia, Vancouver
Department of Computer Science

Event(
Time: DATETIME,
event ID: INTEGER,
Name: VARCHAR,
Description: TEXT)

Staff(
Role: VARCHAR,
Email: VARCHAR,
Name: VARCHAR,
staff ID: INTEGER,
phone #: INTEGER)

Department(
Name: VARCHAR,
Description: TEXT,
event ID: INTEGER)

Sponsor(
sponsor name: VARCHAR,
Amount: DOUBLE,
Request: TEXT,
phone #: INTEGER,
Email: VARCHAR)

Performer(
Agent: VARCHAR,
Name: VARCHAR,
phone #: INTEGER,
Email: VARCHAR)

Venue(
Name: VARCHAR,
Address: VARCHAR,
Capacity: INTEGER,
event ID: INTEGER)

Vendor(
vendor name: VARCHAR,
Location: VARCHAR,
Hours: INTEGER)

Food Vendor(
vendor name: VARCHAR,
alcohol license number: INTEGER,
Style: VARCHAR)

Merchandise Vendor(
vendor name: VARCHAR,
brands carried: TEXT)

Relationships

Attend(
guest name: VARCHAR,
guest ticket #: INTEGER,
event ID: INTEGER)

Sponsors(**event ID**: INTEGER,
sponsor name: VARCHAR,
sponsor email: VARCHAR)

Performs At(
event ID: INTEGER,
performer name: VARCHAR,
performer phone #: INTEGER)

Sells At(
venue name: VARCHAR,
venue address: VARCHAR,
vender name: VARCHAR)

WorksIn(
staff name: VARCHAR**,**
staff ID: INTEGER,
department name: VARCHAR)

5. Functional Dependencies

Guest

- Name, ticket # → email, phone #
- Email → name, ticket #

Staff

- Name, ID → phone #, email, role

- Email → name, role, ID

Sponsor
- Name → email, amount, request, phone #
- Email → name, amount, request

Performer
- Name → email, agent, phone #
- Email → name, agent

Venue
- Name, address → capacity, event ID

Event
- ID → time, name, description

Vendor
- Name → location, hours

Food Vendor
- Name → alcohol license, style

Merchandise Vendor
- Name → brands carried

Department
- Name → description, event ID

6. Normalization

Guest(email, name, ticket #, phone #)
Functional Dependencies: Email → name, ticket #
Relation1, (Email, name, ticket #), Relation2, (Email, phone #)

Staff(Role, email, name, staff ID, phone #)
Functional Dependencies: Email → name, role, ID
Relation3, (Email, name, role, ID), Relation4, (Email, phone #)

Sponsor(sponsor name, amount, request, phone #, email)
Functional Dependencies: Email → name, amount, request
Relation5, (Email, name, amount, request), Relation6, (Email, phone #)

Performer(agent, name, phone #, email)
Functional Dependencies: Email → name, agent
Relation7, (Email, name, agent), Relation8,(Email, phone #)

7. SQL DDL Statements

```
CREATE TABLE Guest
        (name VARCHAR,
        email VARCHAR,
        ticket_number INTEGER,
        phone_number INTEGER,
        PRIMARY KEY (name, ticket_number))

CREATE TABLE Staff
        (staff_id INTEGER,
        name VARCHAR,
        email VARCHAR,
        phone_number INTEGER,
        role VARCHAR,
        PRIMARY KEY (staff_id, name))

CREATE TABLE Sponsor
        (sponsor_name VARCHAR,
        amount INTEGER,
        request VARCHAR,
        phone_number INTEGER,
        email VARCHAR,
        PRIMARY KEY (sponsor_name, email))

CREATE TABLE Performer
        (name VARCHAR PRIMARY KEY,
        agent VARCHAR,
        phone_number INTEGER,
        email VARCHAR)

CREATE TABLE Event(
        time: DATETIME,
        event_id: INTEGER PRIMARY KEY,
        name: VARCHAR,
        description: VARCHAR)
```

```
CREATE TABLE Department(
        name VARCHAR,
        description VARCHAR,
        event_id INTEGER,
        PRIMARY KEY (name, event_id),
        FOREIGN KEY (event_id) REFERENCES Event)

CREATE TABLE Venue(
        name: VARCHAR,
        address: VARCHAR,
        capacity: INTEGER,
        event_id INTEGER,
        PRIMARY KEY (name, address, event_id),
        FOREIGN KEY (event_id) REFERENCES Event))

CREATE TABLE Vendor(
        name: VARCHAR,
        location: VARCHAR,
        hours: INTEGER,
        PRIMARY KEY (name))

Food Vendor(
        name: VARCHAR,
        alcohol_number: INTEGER,
        style: VARCHAR,
        PRIMARY KEY (name),
        FOREIGN KEY (name) REFERENCES Vendor)

Merchandise Vendor(
        name: VARCHAR,
        brands_carried: VARCHAR
        PRIMARY KEY (name),
        FOREIGN KEY (name) REFERENCES Vendor)
```

8. INSERT statements

```
INSERT INTO Guest (name, email, ticket_number, phone_number)
VALUES
('Leo Wang', 'leo.wang@gmail.com', 1001, 123456789),
('Michael Cui', 'michael.cui@gmail.com', 1002, 123456789),
('Kaiser Ninomiya', 'kaiser.ninomiya@gmail.com', 1003, 123456789),
('Jason Zhu', 'jason.zhu@gmail.com', 1004, 123456789),
```

('Stephen Qiao', 'stephen.qiao@gmail.com', 1005, 123456789);

INSERT INTO Staff (staff_id, name, email, phone_number, role)
VALUES
(1, 'Leo Wang', 'leo.wang@gmail.com', 123456789, 'Manager'),
(2, 'Michael Cui', 'michael.cui@gmail.com', 123456789, 'Supervisor'),
(3, 'Kaiser Ninomiya', 'kaiser.ninomiya@gmail.com', 123456789, 'Technician'),
(4, 'Jason Zhu', 'jason.zhu@gmail.com', 123456789, 'Coordinator'),
(5, 'Stephen Qiao', 'stephen.qiao@gmail.com', 123456789, 'Support Staff');

INSERT INTO Sponsor (sponsor_name, amount, request, phone_number, email)
VALUES
('Facebook', 10000, 'Display advertisement at Main Stage', 123456789, ‘facebook@gmail.com'),
('Microsoft', 5000, 'Advertisement', 123456789, 'microsoft@gmail.com'),
('Nike', 7000, 'Pop-up shop', 123456789, 'nike@gmail.com'),
('Apple', 12000, 'Video sponsor', 123456789, 'apple@gmail.com'),
('Alibaba', 8000, 'Advertisement', 123456789, 'alibaba@gmail.com');

INSERT INTO Performer (name, agent, phone_number, email)
VALUES
('NBA Youngboy', 'Rich Paul', 123456789, 'youngboyneverbrokeagain@gmail.com'),
('Drake', 'Chubbs', 123456789, 'aubreygraham@gmail.com'),
('Travis Scott', 'Kylie Jenner', 123456789, 'tscott@gmail.com'),
('Kanye West', 'Pete Davidson', 123456789, 'ye@gmail.com'),
('Taylor Swift', 'Katy Perry', 123456789, 'taylorswift@gmail.com');

INSERT INTO Event (time, event_id, name, description)
VALUES
('2023-11-01 08:00:00', 1, 'NBA Youngboy Concert', 'Concert for rapper NBA Youngboy '),
('2023-11-02 18:00:00', 2, 'NBA game', 'Golden State Warriors vs. Toronto Raptors'),
('2023-11-03 20:00:00', 3, 'NHL game', 'Toronto Maple Leafs vs. Calgary Flames'),
('2023-11-04 14:00:00', 4, 'Drake Concert', 'live performance by the 6ix god'),
('2023-11-05 12:00:00', 5, 'WNBA game', 'New York Liberty vs. Las Vegas Aces');

INSERT INTO Department (name, description, event_id)
VALUES
('Tech', 'Controls event technology', 1),
('Food', 'Manages food sellers at event', 2),
('Lights', 'Controls Lighting at event', 3),
('Audio', 'Controls event audio', 4),
(‘Bookings’, 'Manages bookings', 5);

```
INSERT INTO Venue (name, address, capacity, event_id)
VALUES
('Apple', '123 Granville Street', 500, 1),
('McDonald’s', '456 Burrard Avenue', 800, 2),
('Nike', '789 University Blvd', 600, 3),
('Auto Group', '101 Student Union Blvd', 1000, 4),
('Merch Store', '202 Lougheed Circle', 1500, 5);

INSERT INTO Vendor (name, location, hours)
VALUES
('VenderA', 'North Side', 8),
('VenderB', 'East Side', 6),
('VenderC', 'West Side', 7),
('VenderD', 'South Side', 5),
('VenderE', 'Central Area', 10);
('VenderF', 'Central Area', 10);
('VenderG', 'Central Area', 10);
('VenderH', 'Central Area', 10);
('VenderI', 'Central Area', 10);
('VenderJ', 'Central Area', 10);

INSERT INTO "Food Vendor" (name, alcohol_license_number, style)
VALUES
('VenderA', 1001, 'Italian'),
('VenderB', 1002, 'Chinese'),
('VenderC', 1003, 'Mexican'),
('VenderD', 1004, 'Indian'),
('VenderE', 1005, 'French');

INSERT INTO "Merchandise Vendor" (name, brands_carried)
VALUES
('VenderF', 'Nike'),
('VenderG', 'Lululemon'),
('VenderH', 'Arcteryx'),
('VenderI', 'Adidas'),
('VenderJ', 'Prada, Gucci');
```