University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: 2

Date: October 13th, 2022

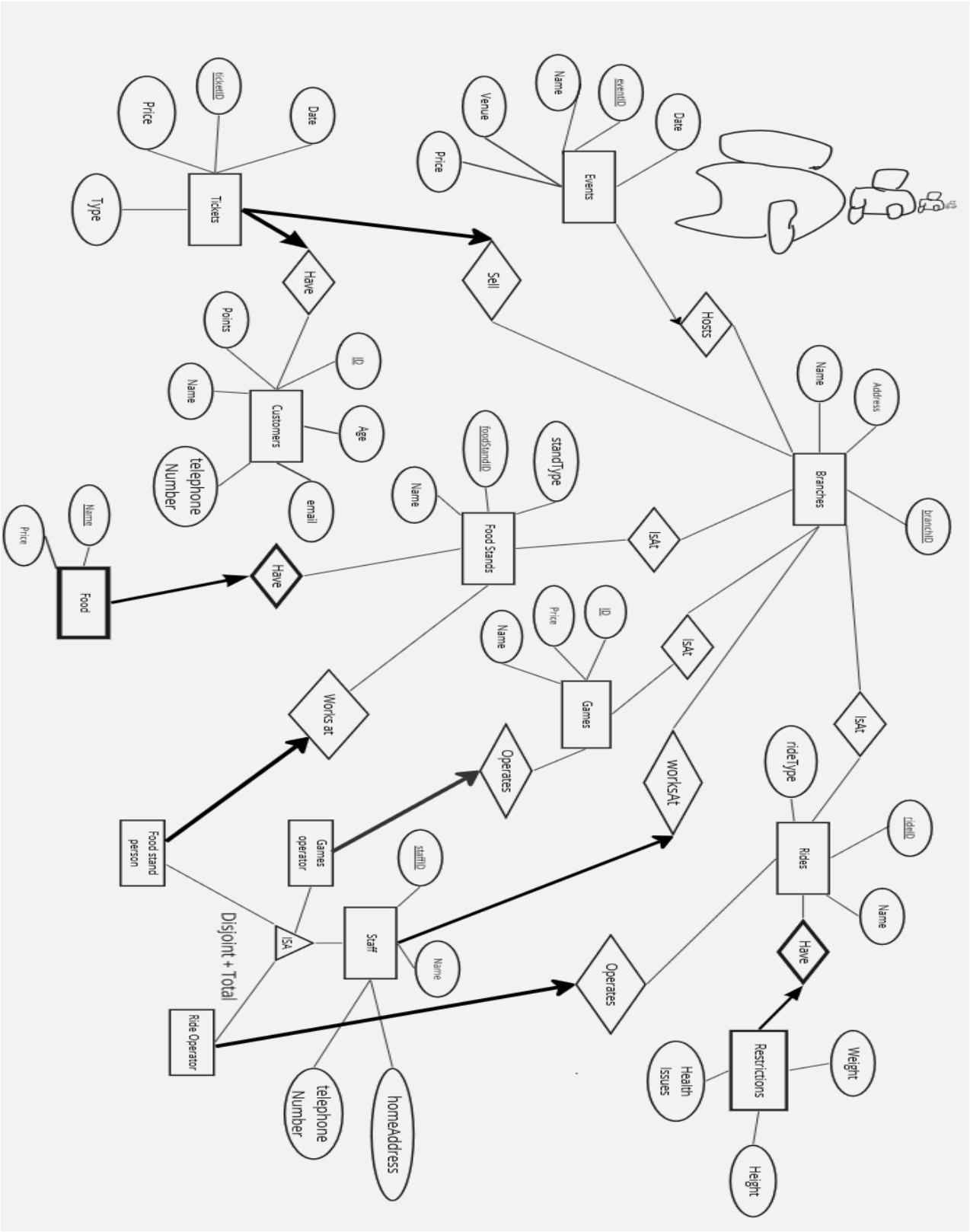
Group Number: 92

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Katrina Huynh	33661737	k2e8u	katrinahunyh2002@hotmail.com
Jaren Agujo	36384543	k6l0t	jaren.agujo@gmail.com
Shubhankar Vakde	48454482	m8c8l	sh2002vk@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

2. ERD Diagram



University of British Columbia, Vancouver
Department of Computer Science

---

3 and 4 Scheme and Functional Dependencies:

Initial tables and FDs:

Staff(staffID: varchar(20), Name: varchar(20), homeAddress: varchar(20), telephoneNumber: int, branchID: varchar(20))
    staffID -> staffName, homeAddress, telephoneNumber, branchID
GamesOperator(**staffID**: varchar(20), **gameStandID**: varchar(20))
FoodStandPerson(**staffID**: varchar(20), **foodStandID**: varchar(20),)
RideOperator(**staffID**: varchar(20), **rideID**:varchar(20))
Tickets(ticketID: varchar(20), Date: varchar(20), **customerID**: varchar(20), **branchID**: varchar(20), price: int, type: varchar(20))
    ticketID -> date, price, type, customerID, branchID
    ticketType -> ticketPrice
Branches(branchID: varchar(20), branchAddress: varchar(20), branchName: varchar(20))
    branchID -> branchAddress, branchName
Events(date: varchar(20), **eventID**: varchar(20), name: varchar(20), venue: varchar(20), price: int, **branchID**: varchar(20))
    eventID -> eventName, eventDate, eventVenue, branchID
    eventName, eventDate -> eventVenue, branchID
GameStand(gameStandID: varchar(20), name: varchar(20), price: int)
    gameID -> gameName, gamePrice
GameStandAt(**gameStandID**: varchar(20), **branchID**: varchar(20))
FoodStands(foodStandID: varchar(20), foodStandName: varchar(20), foodStandType: varchar(20))
    foodStandID -> standType, standName
FoodStandAt(**foodStandID**: varchar(20), **branchID**: varchar(20))
FoodBelongsTo(name: varchar(20), **foodStandID**: varchar(20), price: int)
    name -> price
Rides(**rideID**: varchar(20), name: varchar(20), rideType: varchar(20))
    rideID -> rideName, rideType
RideAt(**rideID**: varchar(20), **branchID**: varchar(20))
Customers(**customerID**: varchar(20), points: int, name: varchar(20), age: int, email: string, telephoneNumber: int)
    customerID -> customerName, points, age, email, customerHomeAddress
    email -> points, name
Restrictions(weight: int, height: int, healthIssues: varchar(20), **rideID:** varchar(20))
    rideID -> weight, height, healthIssues

University of British Columbia, Vancouver
Department of Computer Science

5. Normalization

Customers(**customerID**: string, points: int, name: string, age: int, email: string, telephoneNumber: int)

email -> points, name
customerID -> customerName, points, age, email, customerHomeAddress
BCNF Decomposition

1. find FD that violates BCNF , then create relationships R(A-b) and R(Xb)
   email -> points, name violates,
   R0(email, points, name) R1(customerID, age, telephoneNumber, email)
2. Done, final tables:
   Points(email, points, name) Customer(customerID, age, telephoneNumber, email)

Events(date: date, **eventID**: string, name: string, venue: string, price, **branchID:** string)
eventName, eventDate -> eventVenue, branchID
eventID -> eventName, eventDate, eventVenue

1. find FD that violates BCNF, then create relationships R(A-b) and R(Xb)
   eventName, eventDate -> eventVenue, branchID violates,
   R0(eventVenue, eventDate, eventName, branchID), R1(eventID, eventDate, eventVenue, price)
2. done, final tables:
   EventLocation(eventVenue, eventDate, eventName, **branchID**), Events(eventID, eventDate, eventVenue, price)

Tickets(ticketID: string, Date: string, **customerID**: string, **branchID**: string, price: int, type: string)

ticketID -> date, price, type, customerID, branchID
ticketType -> ticketPrice

1. find FD that violates BCNF, then create relationships R(A-b) and R(Xb)
   ticketType -> ticketPrice violates
   R0(ticketType, ticketPrice), R1(ticketID, date, customerID, branchID, type)
2. done, final tables:
   TicketPrice(ticketType, ticketPrice), Ticket(ticketID, date, **customerID**, **branchID**, type)

FINAL TABLES:

Staff(staffID: varchar(20), Name: varchar(20), homeAddress: varchar(20), telephoneNumber: int, **branchID**: varchar(20))
GamesOperator(**staffID**: varchar(20), **gameStandID**: varchar(20))
FoodStandPerson(**staffID**: varchar(20), **foodStandID**: varchar(20),)
RideOperator(**staffID**: varchar(20), **rideID**: string)
Branches(branchID: varchar(20), branchAddress: varchar(20), branchName: varchar(20))

GameStand(gameStandID: varchar(20), name: varchar(20), price: int)
GameStandAt(**gameStandID**: varchar(20), **branchID**: varchar(20))
FoodStands(foodStandID: varchar(20), foodStandName: varchar(20), foodStandType: varchar(20))
FoodStandAt(**foodStandID**: varchar(20), **branchID**: string)
FoodBelongsTo(name: varchar(20), **foodStandID**: varchar(20), price: int)
Rides(rideID: varchar(20), name: varchar(20), rideType: varchar(20))
RideAt(**rideID**: varchar(20), **branchID**: varchar(20))
Restrictions(weight: int, height: int, healthIssues: varchar(20), **rideID:** varchar(20))

EventLocation(eventVenue: string, eventDate: string, eventName: string, **branchID**: string)
Events(eventID: string, eventDate: string, eventVenue: string, price: int)
Points(email: varchar(20), points: int, name: varchar(20))
Customer(customerID: varchar(20), age: int, **email:** varchar(20)**,** telephoneNumber: int)
TicketPrice(ticketType: varchar(20), ticketPrice: int)
Tickets(ticketID: varchar(20), date: varchar(20), **customerID:** varchar(20), **branchID:** varchar(20), **type:** varchar(20))

6. The SQL DDL statements required to create all the tables from item #5.

```
CREATE TABLE Branches(
        branchID varchar(20) PRIMARY KEY,
        branchAddress varchar(20) UNIQUE,
        branchName varchar(20)
);

CREATE TABLE Staff(
        staffID varchar(20) PRIMARY KEY,
        name varchar(20),
        homeAddress varchar(20),
        telephoneNumber INT UNIQUE,
        branchID varchar(20),
        FOREIGN KEY(branchID) REFERENCES Branches(branchID)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);

CREATE TABLE GameStand(
```

```
        gameStandID varchar(20) PRIMARY KEY,
        name varchar(20),
        price INT
);

CREATE TABLE GamesOperator(
        gameStandID varchar(20),
        staffID varchar(20),
        PRIMARY KEY(gameStandID, staffID),
        FOREIGN KEY(gameStandID) REFERENCES GameStand(gameStandID)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
        FOREIGN KEY(staffID) REFERENCES Staff(staffID)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);

CREATE TABLE GameStandAt(
        gameStandID varchar(20),
        branchID varchar(20),
        PRIMARY KEY(gameStandID, branchID),
        FOREIGN KEY(gameStandID) REFERENCES GameStand(gameStandID)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
        FOREIGN KEY(branchID) REFERENCES Branches(branchID)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);

CREATE TABLE FoodStands(
        foodStandID varchar(20) PRIMARY KEY,
        foodStandName varchar(20),
        foodStandType varchar(20)
);

CREATE TABLE FoodStandPerson(
        foodStandID varchar(20),
        staffID varchar(20),
        PRIMARY KEY(foodStandID, staffID),
        FOREIGN KEY(foodStandID) REFERENCES FoodStands(foodStandID)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
        FOREIGN KEY(staffID) REFERENCES Staff(staffID)
```

```
        ON DELETE CASCADE
        ON UPDATE CASCADE
);

CREATE TABLE FoodStandAt(
        foodStandID varchar(20),
        branchID varchar(20),
        PRIMARY KEY(foodStandID, branchID),
        FOREIGN KEY(foodStandID) REFERENCES FoodStands(foodStandID)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
        FOREIGN KEY(branchID) REFERENCES Branches(branchID)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);

CREATE TABLE FoodBelongsTo(
        name varchar(20),
        foodStandID varchar(20),
        price varchar(20),
        PRIMARY KEY(name, foodStandID),
        FOREIGN KEY(foodStandID) REFERENCES FoodStands(foodStandID)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);

CREATE TABLE Rides(
        rideID varchar(20),
        name varchar(20),
        rideType varchar(20),
        PRIMARY KEY(rideID)
);

CREATE TABLE RideOperator(
        rideID varchar(20),
        staffID varchar(20),
        PRIMARY KEY(rideID, staffID),
        FOREIGN KEY(rideID) REFERENCES Rides(rideID)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
        FOREIGN KEY(staffID) REFERENCES Staff(staffID)
        ON DELETE CASCADE
        ON UPDATE CASCADE
```

```
);


CREATE TABLE RideAt(
        rideID varchar(20),
        branchID varchar(20),
        PRIMARY KEY(rideID, branchID),
        FOREIGN KEY(rideID) REFERENCES Rides(rideID)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
        FOREIGN KEY(branchID) REFERENCES Branches(branchID)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);


CREATE TABLE Restrictions(
        weight INT,
        height INT,
        healthIssues varchar(20),
        rideID varchar(20),
        PRIMARY KEY(rideID),
        FOREIGN KEY(rideID) REFERENCES Rides(rideID)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);

CREATE TABLE EventLocation(
        eventVenue varchar(20),
        eventDate varchar(20),
        eventName varchar(20),
        branchID varchar(20),
        PRIMARY KEY(eventDate, eventName),
        FOREIGN KEY(branchID) REFERENCES Branches(branchID)
)

CREATE TABLE Events(
        eventID varchar(20) PRIMARY KEY,
        eventDate varchar(20),
        eventVenue varchar(20),
        price INT
)
```

```
CREATE TABLE POINTS(
        email varchar(20) PRIMARY KEY,
        points INT,
        name varchar(20)
)

CREATE TABLE Customer(
        customerID varchar(20) PRIMARY KEY,
        age INT,
        email varchar(20),
        telephoneNumber INT,
        FOREIGN KEY(email) REFERENCES Points(email)
        ON DELETE CASCADE
        ON UPDATE CASCADE
)

CREATE TABLE TicketPrice(
        ticketType varchar(20) PRIMARY KEY,
        ticketPrice INT
)

CREATE TABLE Tickets(
        ticketID varchar(20) PRIMARY KEY,
        date varchar(20),
        customerID varchar(20),
        branchID varchar(20),
        ticketType varchar(20),
        FOREIGN KEY(customerID) REFERENCES Customer(customerID)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
        FOREIGN KEY(branchID) REFERENCES Branches(branchID)
        ON DELETE SET NULL
        ON UPDATE CASCADE,
        FOREIGN KEY(ticketType) REFERENCES TicketPrice(ticketType)
        ON DELETE SET NULL
        ON UPDATE CASCADE
)
```

University of British Columbia, Vancouver
Department of Computer Science

7. INSERT statements to populate each table with at least 5 tuples.

```
INSERT
INTO Tickets(ticketID, Date, customerID, branchID, type)
VALUES ('T00000001', 'October 15th 2022', 'C00000001', 'B00000001', 'General')
INSERT
INTO Tickets(ticketID, Date, customerID, branchID, price)
VALUES ('T00000002', 'October 16th 2022', 'C00000002', 'B00000002', 'General')
INSERT
INTO Tickets(ticketID, Date, customerID, branchID, price)
VALUES ('T00000003', 'October 18th 2022', 'C00000003', 'B00000003', 'Senior')
INSERT
INTO Tickets(ticketID, Date, customerID, branchID, price)
VALUES ('T00000004', 'October 18th 2022', 'C00000004', 'B00000002', 'Child')
INSERT
INTO Tickets(ticketID, Date, customerID, branchID, price)
VALUES ('T00000005', 'October 18th 2022', 'C00000005', 'B00000003', 'Child')

INSERT
INTO Branches(branchID, branchAddress, branchName)
VALUES ('B00000001', '1234 Unicorn Road', 'PNE #2')
INSERT
INTO Branches(branchID, branchAddress, branchName)
VALUES ('B00000002', '1234 Unicorn Street', 'PNE #3')
INSERT
INTO Branches(branchID, branchAddress, branchName)
VALUES ('B00000003', '1234 Unicorn Drive', 'PNE #4')
INSERT
INTO Branches(branchID, branchAddress, branchName)
VALUES ('B00000004', '1234 Unicorn Crescent', 'PNE #5')
INSERT
INTO Branches(branchID, branchAddress, branchName)
VALUES ('B00000005', '1234 Unicorn Lane', 'PNE #6')

INSERT
INTO Customers(customerID, age, email, telephoneNumber)
VALUES ('C00000001', 20, 'shubhGreaterThankatrina@ethereum.org', 7781231234)
INSERT
INTO Customers(customerID, age, email, telephoneNumber)
VALUES ('C00000002', 20, 'boogers@gmail.com', 6044206969)
INSERT
INTO Customers(customerID, points, name, age, email, telephoneNumber)
```

VALUES (‘C00000003’, 65, ‘katisbest@hotmail.com’, 7781231235)
INSERT
INTO Customers(customerID, age, email, telephoneNumber)
VALUES (‘C00000004’, 10, ‘georgie2012@icloud.com’, 7781231236)
INTO Customers(customerID, age, email, telephoneNumber)
VALUES (‘C00000005’, 11, ‘cassidy11@icloud.com’, 7781231237)

INSERT
INTO Points(email, points, name)
VALUES(‘shubhGreaterThankatrina@ethereum.org’, 50, ‘Shubh’)
INSERT
INTO Points(email, points, name)
VALUES(‘boogers@gmail.com’, 69, ‘Jaren’)
INSERT
INTO Points(email, points, name)
VALUES(‘katisbest@hotmail.com’, 50, ‘Katrina’)
INSERT
INTO Points(email, points, name)
VALUES(‘georgie2012@icloud.com’, 25, ‘Georgie’)
INSERT
INTO Points(email, points, name)
VALUES(‘cassidy11@icloud.com’, 40, ‘Cassidy’)

INSERT
INTO FoodStands(foodStandID, foodStandName, foodStandType)
VALUES(‘F00000001’, ‘Tim Jortons’, ‘Fast Food’)
INSERT
INTO FoodStands(foodStandID, foodStandName, foodStandType)
VALUES(‘F00000002’, ‘Dunkin Cronuts’, ‘Fast Food’)
INSERT
INTO FoodStands(foodStandID, foodStandName, foodStandType)
VALUES(‘F00000003’, ‘Lemonade R Us’, ‘Drinks’)
INSERT
INTO FoodStands(foodStandID, foodStandName, foodStandType)
VALUES(‘F00000004’, ‘Beaver Nails’, ‘Dessert’)
INSERT
INTO FoodStands(foodStandID, foodStandName, foodStandType)
VALUES(‘F00000005’, ‘Triple Dohs’, ‘Fast Food’)

INSERT
INTO GameStand(gameStandID, name, price)
VALUES (‘G00000001’, ‘Bowling’, 5)
INSERT

INTO GameStand(gameStandID, name, price)
VALUES (‘G00000002’, ‘Shooting Range’, 5)
INSERT
INTO GameStand(gameStandID, name, price)
VALUES (‘G00000003’, ‘Rigged Ladder Game’, 10)
INSERT
INTO GameStand(gameStandID, name, price)
VALUES (‘G00000004’, ‘Baseball’, 5)
INSERT
INTO GameStand(gameStandID, name, price)
VALUES (‘G00000005’, ‘Dead Hang’, 10)

// We only have 3 because those are the only ticket types we will have
INSERT
INTO TicketPrice(ticketType, ticketPrice)
VALUES(‘General’, 35)
INSERT
INTO TicketPrice(ticketType, ticketPrice)
VALUES(‘Senior’, 10)
INSERT
INTO TicketPrice(ticketType, ticketPrice)
VALUES(‘Child’, 25)

INSERT
INTO FoodBelongsTo(name, foodStandID, price)
VALUES(‘Double Double’, ‘F00000001’, 2)
INSERT
INTO FoodBelongsTo(name, foodStandID, price)
VALUES(‘Donut’, ‘F00000002’, 2)
INSERT
INTO FoodBelongsTo(name, foodStandID, price)
VALUES(‘Lemonade’, ‘F00000003’, 5)
INSERT
INTO FoodBelongsTo(name, foodStandID, price)
VALUES(‘Nutella’, ‘F00000004’, 9)
INSERT
INTO FoodBelongsTo(name, foodStandID, price)
VALUES(‘Milkshake’, ‘F00000005’, 7)

```
INSERT
INTO Rides(rideID, name, rideType)
VALUES('R00000001', 'Elevator', 'Spooky')
INSERT
INTO Rides(rideID, name, rideType)
VALUES('R00000002', 'Atmosphere', 'Spooky')
INSERT
INTO Rides(rideID, name, rideType)
VALUES('R00000003', 'Easy Train', 'Kids')
INSERT
INTO Rides(rideID, name, rideType)
VALUES('R00000004', 'Butterfly', 'Kids')
INSERT
INTO Rides(rideID, name, rideType)
VALUES('R00000005', 'Screwdriver', 'Spooky')

INSERT
INTO Restrictions(weight, height, healthIssues, rideID)
VALUES(200, 200, 'pregnant', 'R00000001')
INSERT
INTO Restrictions(weight, height, healthIssues, rideID)
VALUES(200, 200, 'pregnant', 'R00000002')
INSERT
INTO Restrictions(weight, height, healthIssues, rideID)
VALUES(30, 70, 'none', 'R00000003')
INSERT
INTO Restrictions(weight, height, healthIssues, rideID)
VALUES(30, 70, 'none', 'R00000004')
INSERT
INTO Restrictions(weight, height, healthIssues, rideID)
VALUES(200, 200, 'pregnant', 'R00000005')

INSERT
INTO Staff(staffID, name, homeAddress, telephoneNumber, branchID)
VALUES ('S00000001',  'Joe', '123 Dogwood Drive', 7782221234, 'B00000001')
INSERT
INTO Staff(staffID, name, homeAddress, telephoneNumber, branchID)
VALUES ('S00000002',  'Jess', '123 Dogwood Street', 7783331234, 'B00000002')
INSERT
INTO Staff(staffID, name, homeAddress, telephoneNumber, branchID)
VALUES ('S00000003',  'Darcy', '123 Dogwood Crescent', 7784441234, 'B00000003')
INSERT
```

```
INTO Staff(staffID, name, homeAddress, telephoneNumber, branchID)
VALUES ('S00000004',  'Sabrina', '123 Dogwood Way', 7785551234, 'B00000004')
INSERT
INTO Staff(staffID, name, homeAddress, telephoneNumber, branchID)
VALUES ('S00000005',  'Jack', '123 Dogwood Lane', 7786661234, 'B00000005')

INSERT
INTO Staff(staffID, name, homeAddress, telephoneNumber, branchID)
VALUES ('S00000006',  'Yeji', '124 Dogwood Drive', 7783333333, 'B00000001')
INSERT
INTO Staff(staffID, name, homeAddress, telephoneNumber, branchID)
VALUES ('S00000007',  'Ryujin', '123 Dogwood Street', 7784444444, 'B00000002')
INSERT
INTO Staff(staffID, name, homeAddress, telephoneNumber, branchID)
VALUES ('S00000008',  'Lia', '124 Dogwood Crescent', 7785555555, 'B00000003')
INSERT
INTO Staff(staffID, name, homeAddress, telephoneNumber, branchID)
VALUES ('S00000009',  'Chaeryeong', '124 Dogwood Way', 7786666666, 'B00000004')
INSERT
INTO Staff(staffID, name, homeAddress, telephoneNumber, branchID)
VALUES ('S00000010',  'Yuna', '69 Dogwood Lane', 7787777777, 'B00000005')

INSERT
INTO Staff(staffID, name, homeAddress, telephoneNumber, branchID)
VALUES ('S00000011',  'John', '333 Dogwood Drive', 7788888888, 'B00000001')
INSERT
INTO Staff(staffID, name, homeAddress, telephoneNumber, branchID)
VALUES ('S00000012',  'Blyke', '333 Dogwood Street', 7789999999, 'B00000002')
INSERT
INTO Staff(staffID, name, homeAddress, telephoneNumber, branchID)
VALUES ('S00000013',  'Sera', '333 Dogwood Crescent', 7781234567, 'B00000003')
INSERT
INTO Staff(staffID, name, homeAddress, telephoneNumber, branchID)
VALUES ('S00000014',  'Arlo', '333 Dogwood Way', 7781111111, 'B00000004')
INSERT
INTO Staff(staffID, name, homeAddress, telephoneNumber, branchID)
VALUES ('S00000015',  'Isen', '333 Dogwood Lane', 7782222222, 'B00000005')

INSERT
INTO GamesOperator(staffID, gameStandID)
VALUES('B00000001', 'G00000001')
INSERT
INTO GamesOperator(staffID, gameStandID)
```

VALUES('B00000002', 'G00000002')
INSERT
INTO GamesOperator(staffID, gameStandID)
VALUES('B00000003', 'G00000003')
INSERT
INTO GamesOperater(staffID, gameStandID)
VALUES('B00000004', 'G00000004')
INSERT
INTO GamesOperator(staffID, gameStandID)
VALUES('B00000005', 'G00000005')

INSERT
INTO FoodStandPerson(staffID, foodStandID)
VALUES('S00000006', 'F00000001')
INSERT
INTO FoodStandPerson(staffID, foodStandID)
VALUES('S00000007', 'F00000002')
INSERT
INTO FoodStandPerson(staffID, foodStandID)
VALUES('S00000008', 'F00000003')
INSERT
INTO FoodStandPerson(staffID, foodStandID)
VALUES('S00000009', 'F00000004')
INSERT
INTO FoodStandPerson(staffID, foodStandID)
VALUES('S00000010', 'F00000005')

RideOperator(**staffID**: string, **rideID**: string)
INSERT
INTO RideOperator(staffID, rideID)
VALUES('S00000011', 'R00000001')
INSERT
INTO RideOperator(staffID, rideID)
VALUES('S00000012', 'R00000002')
INSERT
INTO RideOperator(staffID, rideID)
VALUES('S00000013', 'R00000003')
INSERT
INTO RideOperator(staffID, rideID)
VALUES('S00000014', 'R00000004')
INSERT
INTO RideOperator(staffID, rideID)
VALUES('S00000015', 'R00000005')

```
INSERT
INTO GameStandAt(gameStandID, branchID)
VALUES('G00000001', 'B00000001')
INSERT
INTO GameStandAt(gameStandID, branchID)
VALUES('G00000002', 'B00000002')
INSERT
INTO GameStandAt(gameStandID, branchID)
VALUES('G00000003', 'B00000003')
INSERT
INTO GameStandAt(gameStandID, branchID)
VALUES('G00000004', 'B00000004')
INSERT
INTO GameStandAt(gameStandID, branchID)
VALUES('G00000005', 'B00000005')

INSERT
INTO RideAt(rideID, branchID)
VALUES('R00000001', 'B00000001')
INSERT
INTO RideAt(rideID, branchID)
VALUES('R00000002', 'B00000002')
INSERT
INTO RideAt(rideID, branchID)
VALUES('R00000003', 'B00000003')
INSERT
INTO RideAt(rideID, branchID)
VALUES('R00000004', 'B00000004')
INSERT
INTO RideAt(rideID, branchID)
VALUES('R00000005', 'B00000005')

INSERT
INTO FoodStandAt(foodStandID, branchID)
VALUES('F00000001', 'B00000001')
INSERT
INTO FoodStandAt(foodStandID, branchID)
VALUES('F00000002', 'B00000002')
INSERT
INTO FoodStandAt(foodStandID, branchID)
VALUES('F00000003', 'B00000003')
INSERT
```

INTO FoodStandAt(foodStandID, branchID)
VALUES('F00000004', 'B00000004')
INSERT
INTO FoodStandAt(foodStandID, branchID)
VALUES('F00000005', 'B00000005')

INSERT
INTO EventLocation(eventVenue, eventDate, eventName, branchID)
VALUES('PNE #2', 'January 5th 2023', 'Rock n Roll', 'B00000001')
INSERT
INTO EventLocation(eventVenue, eventDate, eventName, branchID)
VALUES('PNE #3', 'March 5th 2024', 'Circus', 'B00000002')
INSERT
INTO EventLocation(eventVenue, eventDate, eventName, branchID)
VALUES('PNE #4', 'December 10th 2022', 'Beer Garden', 'B00000003')
INSERT
INTO EventLocation(eventVenue, eventDate, eventName, branchID)
VALUES('PNE #5', 'January 22nd 2023', 'Beer Garden', 'B00000004')
INSERT
INTO EventLocation(eventVenue, eventDate, eventName, branchID)
VALUES('PNE #6', 'May 18th 2023', 'Rock n Roll', 'B00000005')

INSERT
INTO Events(eventID, eventDate, eventVenue, price)
VALUES('E00000001', 'January 5th 2023', 'PNE #2', 40)
INSERT
INTO Events(eventID, eventDate, eventVenue, price)
VALUES('E00000002', 'March 5th 2024', 'PNE #3', 20)
INSERT
INTO Events(eventID, eventDate, eventVenue, price)
VALUES('E00000003', 'December 10th 2022', 'PNE #4', 10)
INSERT
INTO Events(eventID, eventDate, eventVenue, price)
VALUES('E00000004', 'January 22nd 2023', 'PNE #5', 10)
INSERT
INTO Events(eventID, eventDate, eventVenue, price)
VALUES('E00000005', 'May 18th 2023', 'PNE #6', 40)