Database Project

Iowa State University's Ticketing System

May 03, 2024

MIS 320 Section B

Giselle Echeverria

Janida Magallanes

Christian Taylor

Matthew Tifft

William Rasmussen

Table Of Contents

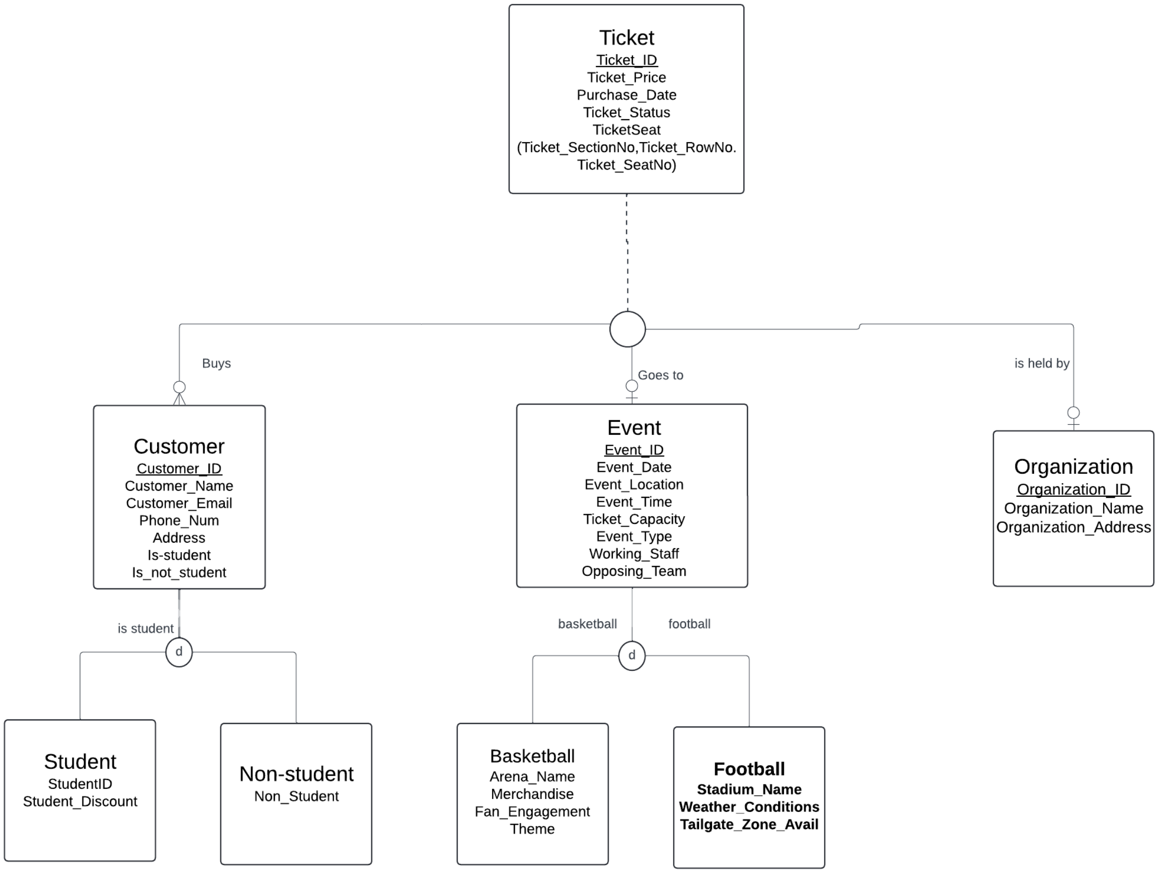
Introduction

We were interested in evaluating Iowa State University's ticketing system, particularly for football and basketball events. With recent changes in ticket scanning procedures, including a transition to mobile and online ticket scanning, we aim to assess the effectiveness of this new system. One prominent issue that has arisen, particularly during the football season, is complaints regarding wait times at entrance points. This problem encompasses various factors, such as the skill level of customers in using their tickets, weather conditions, customer traffic at different entrances, and ticket processing times.

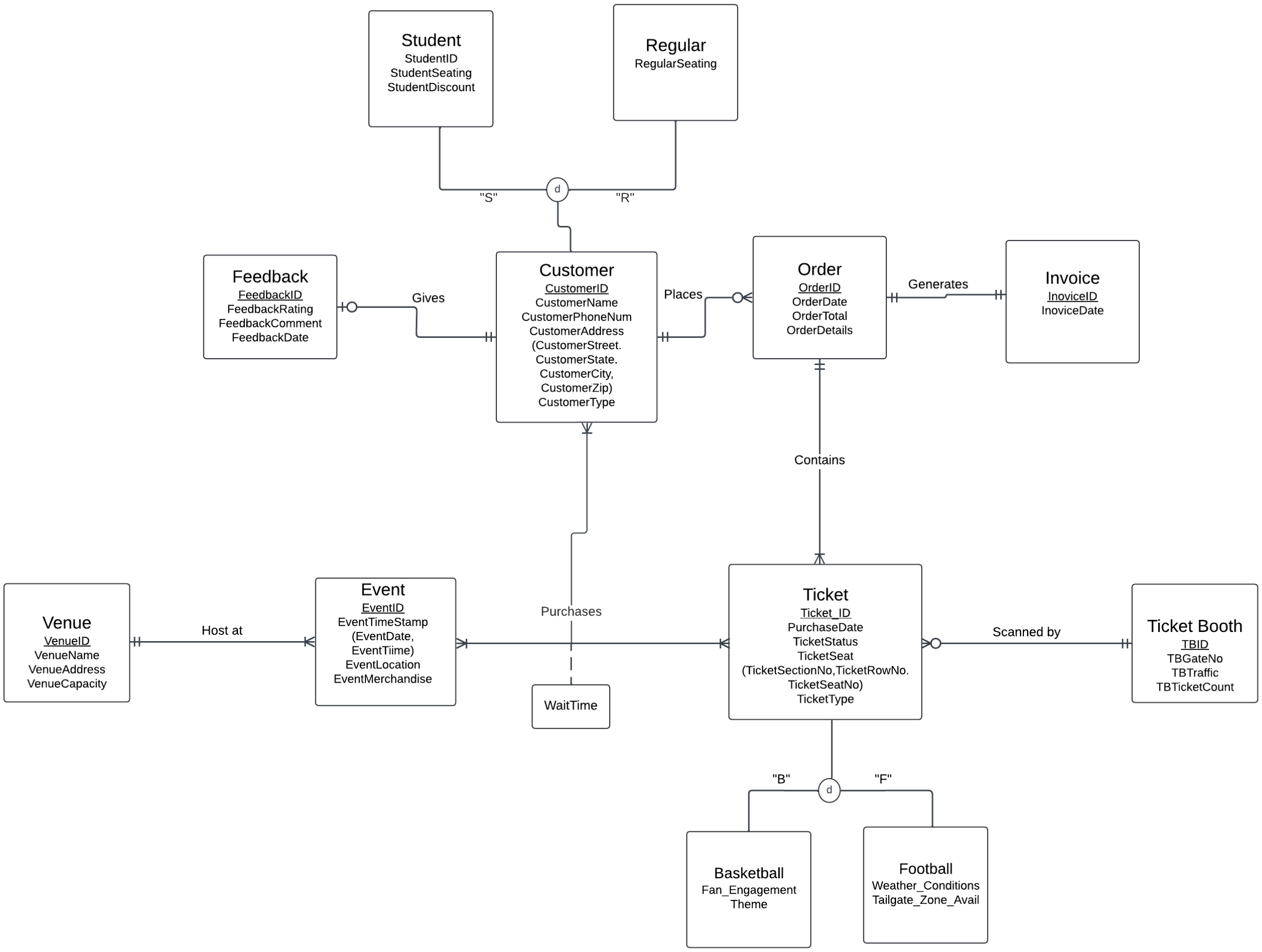
The identified problem of long wait times poses a significant challenge to the efficiency and satisfaction of attendees. It not only leads to frustration among fans but also impacts the overall event experience. However, this challenge also presents an opportunity for improvement. By analyzing the current ticketing system, understanding user experiences, and identifying the root causes of delays, we can implement solutions to streamline the entry process and enhance customer satisfaction.

For our project, our group created a relational database system for the newest Iowa State ticketing system. Our approach to solving the issues with time constraints from scanning tickets we develop an entity relationship diagram that represents the ideal system that ISU should think of using. By showing entities, their attributes, and associated relationships, we determined how to create the database and gather requirements from how the system runs. The database includes tables that store information about customers, feedback, orders, invoices, tickets, ticket booths, venues, events, and times. As a result, the database gives ISU the capability to pinpoint what is causing the time constraints and how to improve the mobile system from customers' feedback. Giving them the opportunity to query and analyze the data and improve the system’s productivity and efficiency.

Conceptual Diagram (original)



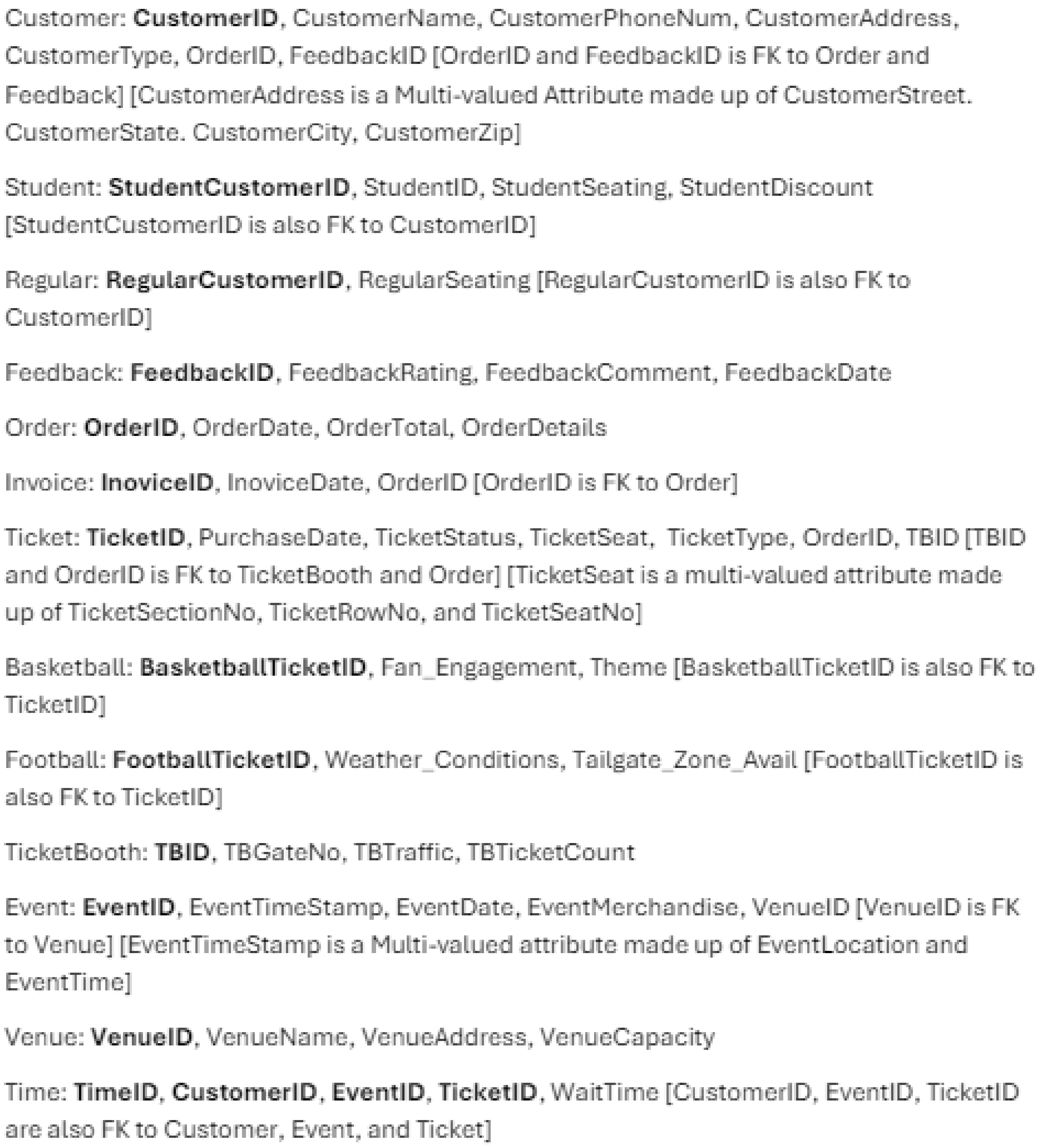
Revised Diagram



Project Assumptions

- An Event may only be hosted by one Venue and must be hosted by a venue.
- A Venue may host multiple events.
- A customer may not place an order but can place multiples.
- An order must generate an invoice and an invoice must be generated by an order.
- A ticket must be scanned at only one ticket booth and each ticket booth can scan zero to many tickets.
- A customer can optionally offer feedback.
- Each feedback entity can only be submitted by one customer.
- Both Customer and Ticket are open to adding more classifications of entity.
- A customer can purchase a ticket to multiple events.
- A ticket to an event may only be purchased by one customer.
- Multiple tickets can be purchased by a customer to an event.
- If a ticket is purchased by a customer, it must be to an event.
- A customer must be involved in a purchased ticket to an event.
- A ticket must be purchased by a customer wanting to go to an event.

Logical Database Design (original)



Customer: **CustomerID**, CustomerName, CustomerPhoneNum, CustomerAddress, CustomerType, OrderID, FeedbackID [OrderID and FeedbackID is FK to Order and Feedback] [CustomerAddress is a Multi-valued Attribute made up of CustomerStreet. CustomerState. CustomerCity, CustomerZip]

Student: **StudentCustomerID**, StudentID, StudentSeating, StudentDiscount [StudentCustomerID is also FK to CustomerID]

Regular: **RegularCustomerID**, RegularSeating [RegularCustomerID is also FK to CustomerID]

Feedback: **FeedbackID**, FeedbackRating, FeedbackComment, FeedbackDate

Order: **OrderID**, OrderDate, OrderTotal, OrderDetails

Invoice: **InoviceID**, InoviceDate, OrderID [OrderID is FK to Order]

Ticket: **TicketID**, PurchaseDate, TicketStatus, TicketSeat, TicketType, OrderID, TBID [TBID and OrderID is FK to TicketBooth and Order] [TicketSeat is a multi-valued attribute made up of TicketSectionNo, TicketRowNo, and TicketSeatNo]

Basketball: **BasketballTicketID**, Fan_Engagement, Theme [BasketballTicketID is also FK to TicketID]

Football: **FootballTicketID**, Weather_Conditions, Tailgate_Zone_Avail [FootballTicketID is also FK to TicketID]

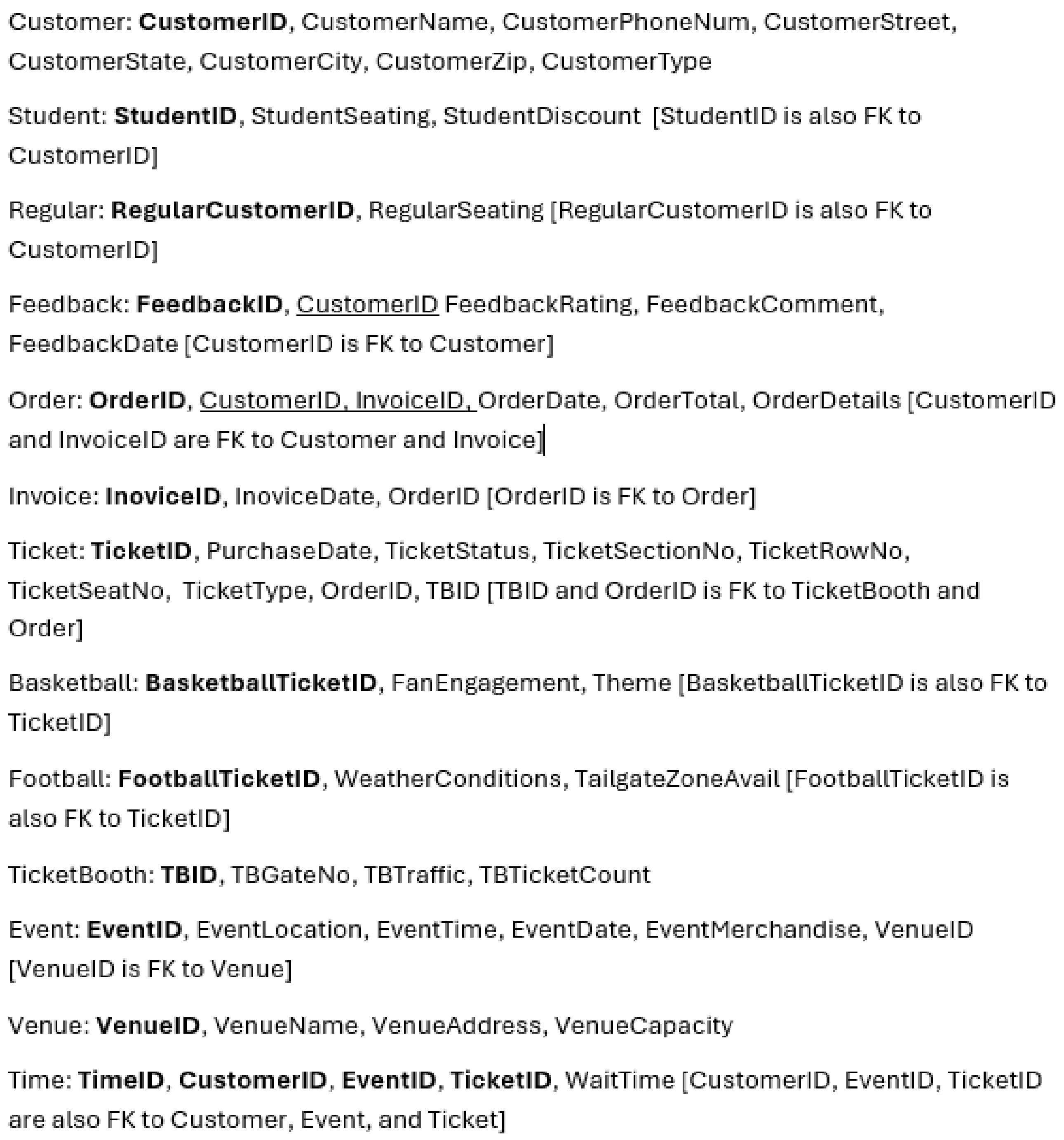
TicketBooth: **TBID**, TBGateNo, TBTraffic, TBTicketCount

Event: **EventID**, EventTimeStamp, EventDate, EventMerchandise, VenueID [VenueID is FK to Venue] [EventTimeStamp is a Multi-valued attribute made up of EventLocation and EventTime]

Venue: **VenueID**, VenueName, VenueAddress, VenueCapacity

Time: **TimeID**, **CustomerID**, **EventID**, **TicketID**, WaitTime [CustomerID, EventID, TicketID are also FK to Customer, Event, and Ticket]

Revised Database Design



Customer: **CustomerID**, CustomerName, CustomerPhoneNum, CustomerStreet, CustomerState, CustomerCity, CustomerZip, CustomerType

Student: **StudentID**, StudentSeating, StudentDiscount [StudentID is also FK to CustomerID]

Regular: **RegularCustomerID**, RegularSeating [RegularCustomerID is also FK to CustomerID]

Feedback: **FeedbackID**, CustomerID FeedbackRating, FeedbackComment, FeedbackDate [CustomerID is FK to Customer]

Order: **OrderID**, CustomerID, InvoiceID, OrderDate, OrderTotal, OrderDetails [CustomerID and InvoiceID are FK to Customer and Invoice]

Invoice: **InoviceID**, InoviceDate, OrderID [OrderID is FK to Order]

Ticket: **TicketID**, PurchaseDate, TicketStatus, TicketSectionNo, TicketRowNo, TicketSeatNo, TicketType, OrderID, TBID [TBID and OrderID is FK to TicketBooth and Order]

Basketball: **BasketballTicketID**, FanEngagement, Theme [BasketballTicketID is also FK to TicketID]

Football: **FootballTicketID**, WeatherConditions, TailgateZoneAvail [FootballTicketID is also FK to TicketID]

TicketBooth: **TBID**, TBGateNo, TBTraffic, TBTicketCount

Event: **EventID**, EventLocation, EventTime, EventDate, EventMerchandise, VenueID [VenueID is FK to Venue]

Venue: **VenueID**, VenueName, VenueAddress, VenueCapacity

Time: **TimeID**, **CustomerID**, **EventID**, **TicketID**, WaitTime [CustomerID, EventID, TicketID are also FK to Customer, Event, and Ticket]

Table Creation Script

```
create database mis320project;
use mis320project;

CREATE TABLE Customer (
   CustomerID INT NOT NULL,
   CustomerName VARCHAR(40) NOT NULL,
   CustomerPhoneNum VARCHAR(20) NOT NULL,
   CustomerStreet VARCHAR(40) NOT NULL,
   CustomerState CHAR(2) NOT NULL,
   CustomerCity VARCHAR(20) NOT NULL,
   CustomerZIP VARCHAR(10) NOT NULL,
   CustomerType VARCHAR(20) NOT NULL,
   PRIMARY KEY (CustomerID));

CREATE TABLE Student (
   StudentID INT NOT NULL,
   StudentSeating VARCHAR(5) NOT NULL,
   StudentDiscount FLOAT(8,2) NOT NULL,
   PRIMARY KEY (StudentID),
   CustomerID INT REFERENCES Customer);

CREATE TABLE Regular (
   RegularCustomerID INT NOT NULL,
   RegularSeating VARCHAR(5) NOT NULL,
   PRIMARY KEY (RegularCustomerID),
   CustomerID INT REFERENCES Customer);
```

```
CREATE TABLE Feedback (
   FeedbackID INT NOT NULL,
   FeedbackRating INT NOT NULL,
   FeedbackComment VARCHAR(100),
   FeedbackDate DATE,
   PRIMARY KEY (FeedbackID),
   CustomerID INT REFERENCES Customer);

CREATE TABLE `Order` (
   OrderID INT NOT NULL,
   OrderDate DATE,
   OrderTotal FLOAT(10,2) NOT NULL,
   OrderDetails VARCHAR(10000) NOT NULL,
   PRIMARY KEY (OrderID),
   CustomerID INT REFERENCES Customer,
    InvoiceID INT REFERENCES Invoice);

CREATE TABLE Invoice (
   InvoiceID INT NOT NULL,
   InvoiceDate DATE,
   PRIMARY KEY (InvoiceID),
   OrderID INT REFERENCES `Order`);

CREATE TABLE TicketBooth (
   TBID INT NOT NULL,
   TBGateNo INT NOT NULL,
   TBTraffic INT NOT NULL,
   TBTicketCount INT NOT NULL,
   PRIMARY KEY (TBID) );

CREATE TABLE Ticket (
   TicketID INT NOT NULL,
   PurchaseDate DATE,
   TicketStatus VARCHAR(40) NOT NULL,
   TicketSectionNo INT NOT NULL,
   TicketRowNo VARCHAR(10) NOT NULL,
   TicketSeatNo INT NOT NULL,
   TicketType VARCHAR(40) NOT NULL,
   PRIMARY KEY (TicketID),
   OrderID INT REFERENCES `Order` ,
   TBID INT REFERENCES TicketBooth );

CREATE TABLE Basketball (
   BasketballTicketID INT NOT NULL,
   FanEngagement FLOAT(10,2) NOT NULL,
   Theme VARCHAR(40) NOT NULL,
   PRIMARY KEY (BasketballTicketID),
   TicketID INT REFERENCES Ticket);

CREATE TABLE Football (
   FootballTicketID INT NOT NULL,
   WeatherConditions VARCHAR(40) NOT NULL,
   TailgateZoneAvail boolean NOT NULL,
   PRIMARY KEY (FootballTicketID),
   TicketID INT REFERENCES Ticket);

CREATE TABLE Venue (
   VenueID INT NOT NULL,
   VenueName VARCHAR(40) NOT NULL,
   VenueAddress VARCHAR(40) NOT NULL,
   VenueCapacity INT NOT NULL,
   PRIMARY KEY (VenueID));

CREATE TABLE Event (
   EventID INT NOT NULL,
   EventLocation VARCHAR(40) NOT NULL,
   EventTime TIME,
   EventDate DATE,
   EventMerchandise VARCHAR(40) NOT NULL,
   PRIMARY KEY (EventID),
   VenueID INT REFERENCES Venue);

CREATE TABLE `Time` (
   TimeID INT NOT NULL,
   WaitTime TIME NOT NULL,
   PRIMARY KEY (TimeID),
   CustomerID INT REFERENCES Customer,
   EventID INT REFERENCES Event,
   TicketID INT REFERENCES Ticket);
```

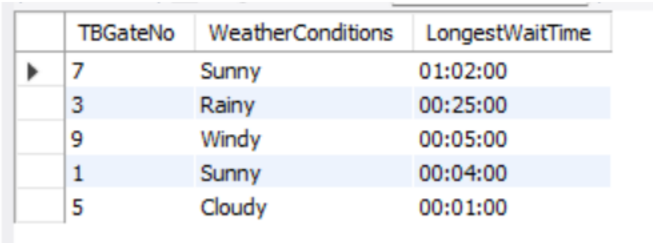
Table Insert Script

```
INSERT INTO Customer VALUES (1000, 'John Doe', '123-456-7890', '123 Main St', 'CA', 'Los Angeles', '90001', 'Regular');

INSERT INTO Customer VALUES (1001, 'Jane Smith', '987-654-3210', '456 Elm St', 'IA', 'Marshalltown', '10001', 'Student');

INSERT INTO Customer VALUES (1002, 'Michael Johnson', '555-123-4567', '789 Oak St', 'TX', 'Houston', '77001', 'Regular');

INSERT INTO Customer VALUES (1003, 'Emily Brown', '111-222-3333', '321 Maple St', 'FL', 'Miami', '33101', 'Regular');

INSERT INTO Customer VALUES (1004, 'David Lee', '999-888-7777', '555 Pine St', 'WA', 'Seattle', '98101', 'Student');

INSERT INTO Customer VALUES (1005, 'Sarah Taylor', '444-555-6666', '777 Cedar St', 'IL', 'Chicago', '60601', 'Regular');

INSERT INTO Customer VALUES (1006, 'Chris Evans', '666-777-8888', '888 Birch St', 'IA', 'Ames', '94101', 'Student');

INSERT INTO Customer VALUES (1007, 'Jessica Martinez', '222-333-4444', '999 Walnut St', 'IA', 'Ames', '75201', 'Regular');

INSERT INTO Customer VALUES (1008, 'Andrew Wilson', '777-888-9999', '111 Ash St', 'IA', 'Ames', '14201', 'Student');

INSERT INTO Customer VALUES (1009, 'Olivia Johnson', '888-999-0000', '222 Spruce St', 'IA', 'Des Moines', '50301', 'Regular');


INSERT INTO Student VALUES (57382, 'A1', 0.10, 1008);

INSERT INTO Student VALUES (94105, 'A2', 0.15, 1006);

 INSERT INTO Student VALUES (20876, 'A3', 0.20, 1004);

INSERT INTO Student VALUES (68934, 'A4', 0.25, 1001);
INSERT INTO Regular VALUES (1, 'A101', 1009);
INSERT INTO Regular VALUES (2, 'B202', 1007);
INSERT INTO Regular VALUES (3, 'C303', 1005);
INSERT INTO Regular VALUES (4, 'D404', 1003);
INSERT INTO Regular VALUES (5, 'E505', 1002);
INSERT INTO Regular VALUES (6, 'F606', 1000);

INSERT INTO Feedback VALUES (2000, 4, 'Great service!', '2024-04-01', 1003);
INSERT INTO Feedback VALUES (2001, 3, 'Could be better, the wait to get in was long.', '2024-04-02', 1009);
INSERT INTO Feedback VALUES (2002, 5, 'Excellent experience overall.', '2024-04-03', 1000);
INSERT INTO Feedback VALUES (2003, 2, 'Disappointed with the food quality.', '2024-04-04', 1004);
INSERT INTO Feedback VALUES (2004, 4, 'Friendly staff.', '2024-04-05', 1005);
INSERT INTO Feedback VALUES (2005, 5, 'Best place in town!', '2024-04-06', 1006);
INSERT INTO Feedback VALUES (2006, 3, 'Average service.', '2024-04-07', 1001);
INSERT INTO Feedback VALUES (2007, 4, 'Would recommend to friends.', '2024-04-08', 1002);
INSERT INTO Feedback VALUES (2008, 2, 'Long waiting time.', '2024-04-09', 1007);
INSERT INTO Feedback VALUES (2009, 5, 'Absolutely amazing!', '2024-04-10', 1008);

INSERT INTO `Order` VALUES (3000, '2024-04-01', 100.00, '2 Adult Tickets for Football Match - Section 102, Row D, Seats 5 and 6', 1000, 4000);
INSERT INTO `Order` VALUES (3001, '2024-04-02', 150.50, '3 Student Tickets for Basketball Game - Upper Deck, Row 15, Seats 20, 21, and 22', 1001, 4001);
INSERT INTO `Order` VALUES (3002, '2024-04-03', 200.25, 'VIP Box Tickets for Football Match - Box A, Seats 1 to 10', 1002, 4002);

INSERT INTO `Order` VALUES (3003, '2024-04-04', 120.75, 'Family Pass for Basketball Game - Section 205, Row F, Seats 1 to 4', 1003, 4003);

INSERT INTO `Order` VALUES (3004, '2024-04-05', 180.00, 'Corporate Suite Tickets for Football Match - Suite C, Seats 1 to 20', 1004, 4004);

INSERT INTO `Order` VALUES (3005, '2024-04-06', 90.80, '4 General Admission Tickets for Basketball Game - Upper Deck', 1005, 4005);

INSERT INTO `Order` VALUES (3006, '2024-04-07', 220.00, 'Season Pass for Football Matches - Access to all home games', 1006, 4006);

INSERT INTO `Order` VALUES (3007, '2024-04-08', 130.20, 'VIP Experience for Basketball Game - Courtside Seats, Dinner with Players', 1007, 4007);

INSERT INTO `Order` VALUES (3008, '2024-04-09', 170.40, 'Single Ticket for Football Match - Section 105, Row G, Seat 15', 1008, 4008);

INSERT INTO `Order` VALUES (3009, '2024-04-10', 163.75, 'Single Ticket for Football Match - Section 103, Row C, Seat 10', 1009, 4009);


INSERT INTO Invoice VALUES (4000, '2024-04-01', 3000);

INSERT INTO Invoice VALUES (4001, '2024-04-02', 3001);

INSERT INTO Invoice VALUES (4002, '2024-04-03', 3002);

INSERT INTO Invoice VALUES (4003, '2024-04-04', 3003);

INSERT INTO Invoice VALUES (4004, '2024-04-05', 3004);

INSERT INTO Invoice VALUES (4005, '2024-04-06', 3005);

INSERT INTO Invoice VALUES (4006, '2024-04-07', 3006);

INSERT INTO Invoice VALUES (4007, '2024-04-08', 3007);

INSERT INTO Invoice VALUES (4008, '2024-04-09', 3008);

INSERT INTO Invoice VALUES (4009, '2024-04-10', NULL);
INSERT INTO Ticket VALUES (5000, '2024-04-01', 'Valid', 102, 'D', 5, 'Football', 3000, 100);
INSERT INTO Ticket VALUES (5001, '2024-04-02', 'Valid', 201, 'F', 20, 'Basketball', 3001, 101);
INSERT INTO Ticket VALUES (5002, '2024-04-03', 'Valid', 105, 'A', 1, 'Football', 3002, 102);
INSERT INTO Ticket VALUES (5003, '2024-04-04', 'Valid', 205, 'F', 1, 'Basketball', 3003, 103);
INSERT INTO Ticket VALUES (5004, '2024-04-05', 'Valid', 303, 'C', 1, 'Football', 3004, 104);
INSERT INTO Ticket VALUES (5005, '2024-04-06', 'Valid', 401, 'A', 1, 'Basketball', 3005, 105);
INSERT INTO Ticket VALUES (5006, '2024-04-07', 'Valid', 101, 'A', 1, 'Football', 3006, 106);
INSERT INTO Ticket VALUES (5007, '2024-04-08', 'Valid', 104, 'B', 2, 'Basketball', 3007, 107);
INSERT INTO Ticket VALUES (5008, '2024-04-09', 'Valid', 102, 'G', 15, 'Football', 3008, 108);
INSERT INTO Ticket VALUES (5009, '2024-04-10', 'Valid', 103, 'C', 10, 'Basketball', 3009, 109);

INSERT INTO Basketball VALUES (600, 85.50, 'Classic', 5001);
INSERT INTO Basketball VALUES (601, 92.75, 'Retro Night', 5003);
INSERT INTO Basketball VALUES (602, 79.25, 'Superhero Night', 5005);
INSERT INTO Basketball VALUES (603, 85.00, 'Fan Appreciation', 5007);

INSERT INTO Football VALUES (700, 'Sunny', 1, 5000);
INSERT INTO Football VALUES (701, 'Rainy', 0, 5002);
INSERT INTO Football VALUES (702, 'Cloudy', 1, 5004);
INSERT INTO Football VALUES (703, 'Sunny', 1, 5006);
INSERT INTO Football VALUES (704, 'Windy', 0, 5008);

INSERT INTO TicketBooth VALUES (100, 1,6 , 3000);
INSERT INTO TicketBooth VALUES (101, 2, 4, 3500);
INSERT INTO TicketBooth VALUES (102, 3, 6, 1000);
INSERT INTO TicketBooth VALUES (103, 4, 5, 2000);
INSERT INTO TicketBooth VALUES (104, 5, 5, 3000);
INSERT INTO TicketBooth VALUES (105, 6, 4, 3500);
INSERT INTO TicketBooth VALUES (106, 7, 3, 4000);
INSERT INTO TicketBooth VALUES (107, 8, 6, 500);
INSERT INTO TicketBooth VALUES (108, 9, 5, 1000);
INSERT INTO TicketBooth VALUES (109, 10, 6, 550);

INSERT INTO Venue VALUES (300, 'Stadium A', '123 Main Street', 20000);
INSERT INTO Venue VALUES (301, 'Arena B', '456 Elm Street', 15000);
INSERT INTO Venue VALUES (302, 'Field C', '789 Oak Street', 30000);
INSERT INTO Venue VALUES (303, 'Park D', '101 Pine Street', 25000);
INSERT INTO Venue VALUES (304, 'Center E', '222 Maple Street', 18000);
INSERT INTO Venue VALUES (305, 'Gymnasium F', '333 Cedar Street', 22000);
INSERT INTO Venue VALUES (306, 'Auditorium G', '444 Walnut Street', 28000);
INSERT INTO Venue VALUES (307, 'Coliseum H', '555 Birch Street', 26000);
INSERT INTO Venue VALUES (308, 'Pavilion I', '666 Pine Street', 19000);
INSERT INTO Venue VALUES (309, 'Hall J', '777 Oak Street', 21000);

INSERT INTO Event VALUES (8000, 'Stadium A', '19:00:00', '2024-05-01', 'Team Jerseys', 300);
INSERT INTO Event VALUES (8001, 'Arena B', '18:30:00', '2024-05-05', 'Caps and Hats', 301);
INSERT INTO Event VALUES (8002, 'Field C', '20:00:00', '2024-05-10', 'Scarves and Flags', 302);
INSERT INTO Event VALUES (8003, 'Park D', '17:00:00', '2024-05-15', 'Keychains and Magnets', 303);
INSERT INTO Event VALUES (8004, 'Center E', '16:30:00', '2024-05-20', 'T-shirts and Hoodies', 304);
INSERT INTO Event VALUES (8005, 'Gymnasium F', '19:30:00', '2024-05-25', 'Water Bottles and Towels', 305);

INSERT INTO Event VALUES (8006, 'Auditorium G', '18:00:00', '2024-05-30', 'Posters and Pennants', 306);

INSERT INTO Event VALUES (8007, 'Coliseum H', '20:30:00', '2024-06-01', 'Autographed Memorabilia', 307);

INSERT INTO Event VALUES (8008, 'Pavilion I', '17:30:00', '2024-06-05', 'Foam Fingers and Pom Poms', 308);

INSERT INTO Event VALUES (8009, 'Hall J', '18:45:00', '2024-06-10', 'Sunglasses and Visors', 309);


INSERT INTO `Time` VALUES (200, '00:04:00', 1000, 8000, 5000);

INSERT INTO `Time` VALUES (201, '00:01:00', 1001, 8001, 5001);

INSERT INTO `Time` VALUES (202, '00:25:00', 1002, 8002, 5002);

INSERT INTO `Time` VALUES (203, '00:04:00', 1003, 8003, 5003);

INSERT INTO `Time` VALUES (204, '00:01:00', 1004, 8004, 5004);

INSERT INTO `Time` VALUES (205, '00:05:00', 1005, 8005, 5005);

INSERT INTO `Time` VALUES (206, '01:02:00', 1006, 8006, 5006);

INSERT INTO `Time` VALUES (207, '01:15:00', 1007, 8007, 5007);

INSERT INTO `Time` VALUES (208, '00:05:00', 1008, 8008, 5008);

INSERT INTO `Time` VALUES (209, '00:11:00', 1009, 8009, 5009);
```

Queries

Query 1: Which entrance gates have the longest wait times during football events, and how does weather affect these wait times?

```
SELECT TB.TBGateNo, F.WeatherConditions, MAX(T.WaitTime) AS LongestWaitTime

FROM TicketBooth TB

JOIN Ticket TC ON TB.TBID = TC.TBID

JOIN Football F ON F.ticketID = TC.ticketID

JOIN Time T ON TC.ticketid = T.ticketed

GROUP BY TB.TBGateNo, F.WeatherConditions

ORDER BY LongestWaitTime DESC;
```



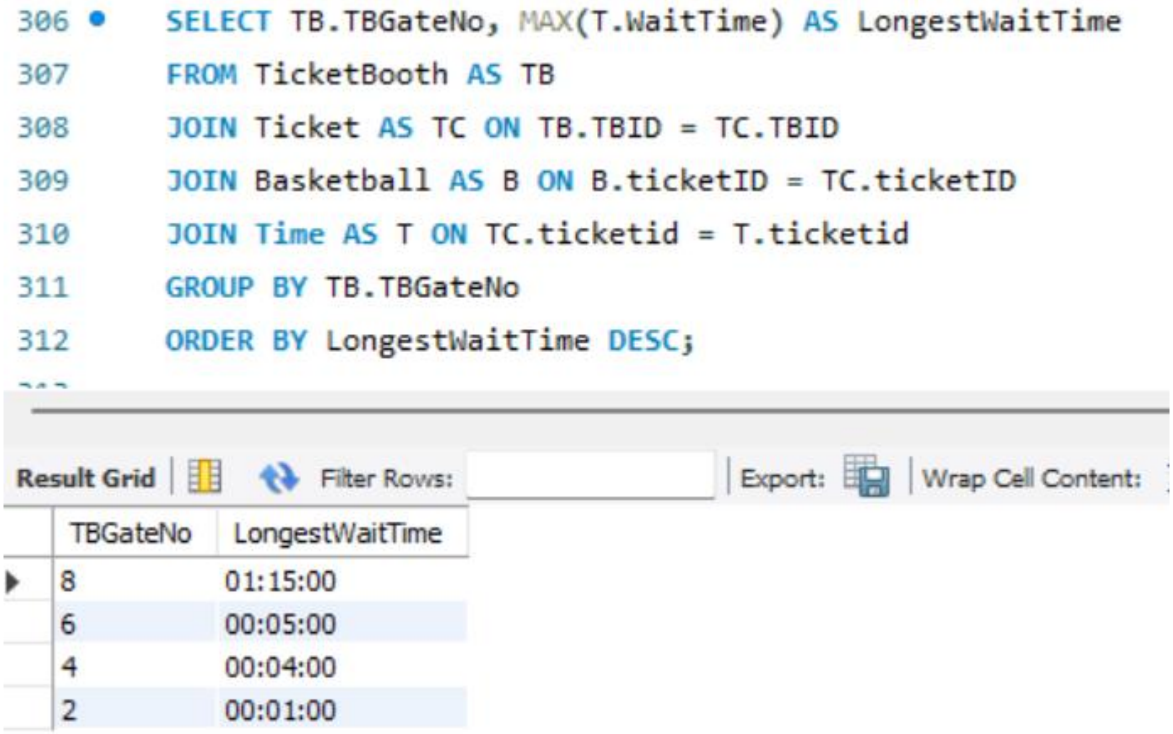
TBGateNo	WeatherConditions	LongestWaitTime
7	Sunny	01:02:00
3	Rainy	00:25:00
9	Windy	00:05:00
1	Sunny	00:04:00
5	Cloudy	00:01:00

Purpose: This query can help analyze which entrance gates are the busiest during football games and how the weather affects how long people must wait. This allows ISU to allocate resources effectively. For instance, if more staff are needed on particular gate to decrease the waiting time and if the weather played a factor in it.

Query 2:

```
SELECT TB.TBGateNo, MAX(TWaitTime) AS LongestWaitTime

From TicketBooth AS TB

JOIN Ticket AS TC ON TB.TBID = TC.TBID

JOIN Basketball AS B ON B.ticketID = TC.ticketID

JOIN Time AS T ON TC.ticketid = T.ticketid

GROUP BY TB.TBGateNo
```

ORDER BY LongestWaitTime DESC;



```
SELECT TB.TBGateNo, MAX(T.WaitTime) AS LongestWaitTime
FROM TicketBooth AS TB
JOIN Ticket AS TC ON TB.TBID = TC.TBID
JOIN Basketball AS B ON B.ticketID = TC.ticketID
JOIN Time AS T ON TC.ticketid = T.ticketid
GROUP BY TB.TBGateNo
ORDER BY LongestWaitTime DESC;
```

TBGateNo	LongestWaitTime
8	01:15:00
6	00:05:00
4	00:04:00
2	00:01:00

Purpose: This query helps analyze which entrance gates are the busiest during basketball games. It will show where logjams occur and help identify ways to let foot traffic flow freely.

Query 3:

SELECT t.TBGateNo, t.TBTraffic, t,TBTicketCount, ` Time` .WaitTime

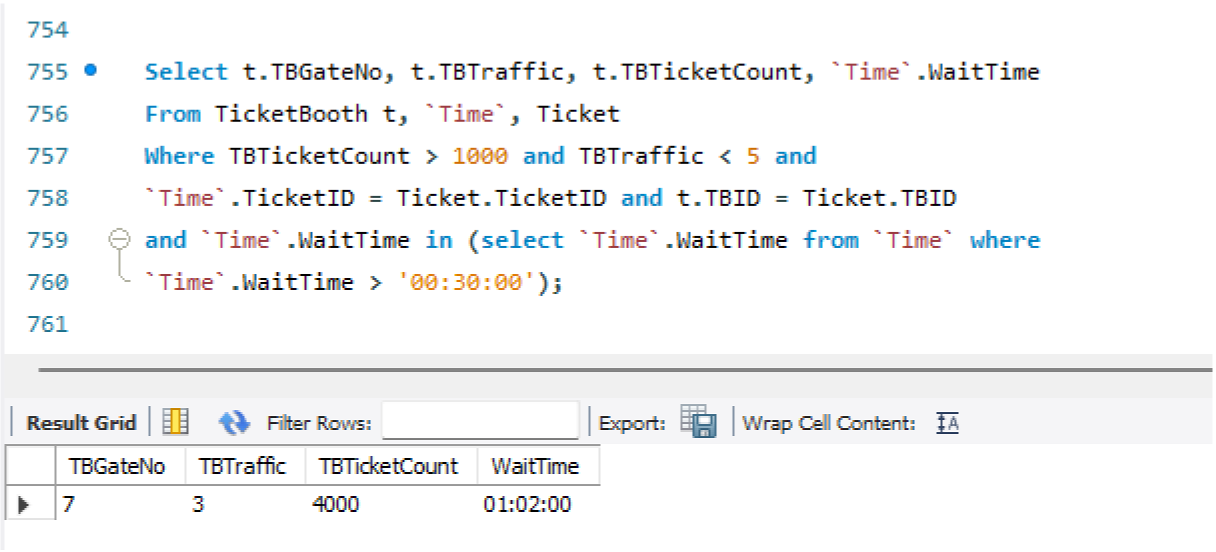
FROM TicketBooth t, ` Time` , Ticket

WHERE TBTicketCount > 1000 and TBTraffic < 5 and

` Time` .TicketID = Ticket.TicketID and t.TBID = Ticket.TBID

And ` Time` .WaitTime in (Select ` Time` .WaitTime from ` Time` Where

` Time` .WaitTime > ‘00:30:00’);



```
Select t.TBGateNo, t.TBTraffic, t.TBTicketCount, `Time`.WaitTime
From TicketBooth t, `Time`, Ticket
Where TBTicketCount > 1000 and TBTraffic < 5 and
`Time`.TicketID = Ticket.TicketID and t.TBID = Ticket.TBID
and `Time`.WaitTime in (select `Time`.WaitTime from `Time` where
`Time`.WaitTime > '00:30:00');
```

TBGateNo	TBTraffic	TBTicketCount	WaitTime
7	3	4000	01:02:00

Purpose: This query helps identify the flow of traffic on problem Ticket Booths. By showing both the amount of tickets that go through the Ticket Booth, and the amount of tickets that can go through the Ticket booth in a minute (TBTraffic). This helps identify which Ticket Booths are slow and why they may be going slow. This helps ISU both understand the amount of traffic that goes through each Ticket Gate and what needs to be done to either increase the flow, or to decrease the number of tickets going through said ticket booth.

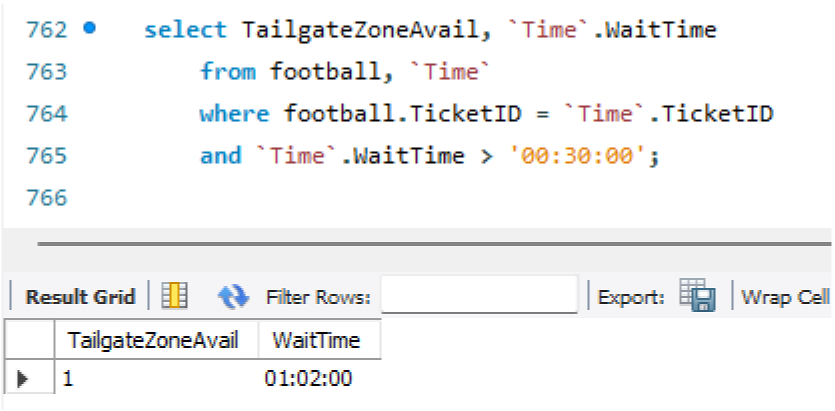
Query 4:

Select TailgateZoneAvail, ` Time` .WaitTime

From football, ` Time`

Where football.TicketID = ` Time` .TicketID

And ` Time` .WaitTime > '00:30:00';



```
select TailgateZoneAvail, `Time`.WaitTime
    from football, `Time`
    where football.TicketID = `Time`.TicketID
    and `Time`.WaitTime > '00:30:00';
```

TailgateZoneAvail	WaitTime
1	01:02:00

Purpose: This query helps identify whether Tailgate Zones can affect wait times. As can be seen from this sample of data, the only football ticket that had a wait time over 30:00 was a Tailgate zone ticket. This means that being in a Tailgate zone could negatively affect wait times. This helps ISU understand whether Tailgate zones are a problem with wait times and can take necessary action to either decrease the activity going through ticket booths that tailgate zone tickets tend to go through, or to implement changes to the tailgating zone to make the wait times shrink.

Query 5:

SELECT Customer.CustomerName, Customer.CustomerState, f1FeedbackRating

FROM Customer, Feedback f1

WHERE f1.CustomerID = Customer.CustomerID

AND f1.FeedbackRating < ALL

(SELECT FeedbackRating

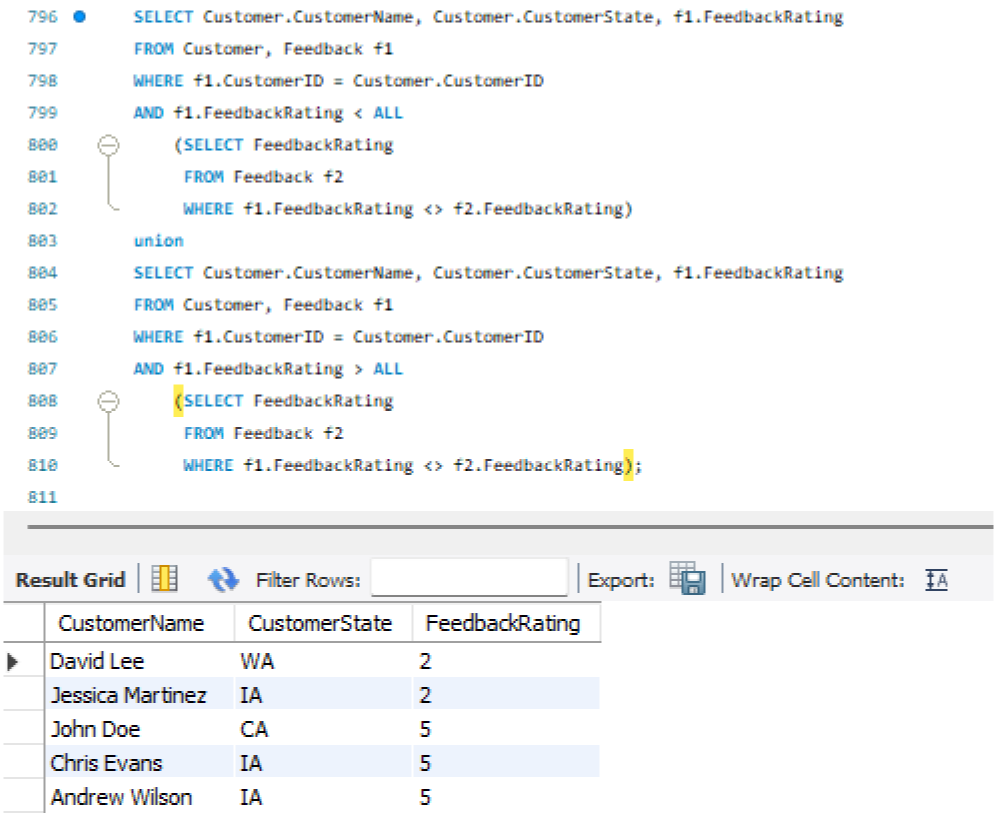
FROM Feedback f2

WHERE f1.FeedbackRating <> f2.FeedbackRating)

Union

SELECT Customer.CustomerName, Customer.CustomerState, f1.FeedbackRating

```
From Customer, Feedback f1
WHERE f1.CustomerID = Customer.CustomerID
AND f1.FeedbackRating > ALL
        (SELECT FeedbackRating
         FROM Feedback f2
         WHERE f1.FeedbackRating <> f.2FeedbackRating);
```



```
SELECT Customer.CustomerName, Customer.CustomerState, f1.FeedbackRating
FROM Customer, Feedback f1
WHERE f1.CustomerID = Customer.CustomerID
AND f1.FeedbackRating < ALL
    (SELECT FeedbackRating
     FROM Feedback f2
     WHERE f1.FeedbackRating <> f2.FeedbackRating)
union
SELECT Customer.CustomerName, Customer.CustomerState, f1.FeedbackRating
FROM Customer, Feedback f1
WHERE f1.CustomerID = Customer.CustomerID
AND f1.FeedbackRating > ALL
    (SELECT FeedbackRating
     FROM Feedback f2
     WHERE f1.FeedbackRating <> f2.FeedbackRating);
```

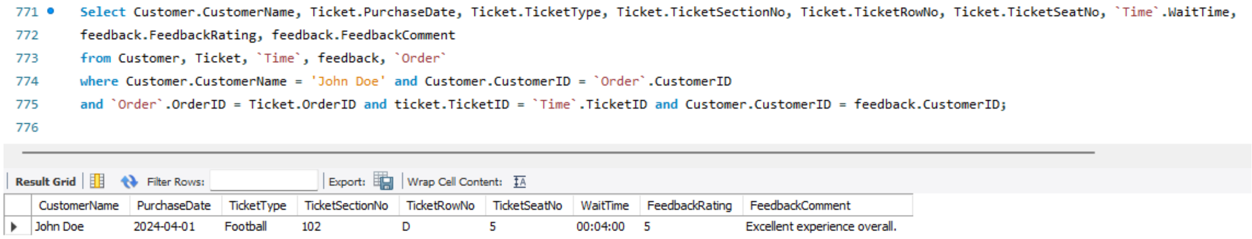
CustomerName	CustomerState	FeedbackRating
David Lee	WA	2
Jessica Martinez	IA	2
John Doe	CA	5
Chris Evans	IA	5
Andrew Wilson	IA	5

Purpose:

The purpose of this query is to provide the list of customer names, the state they are from or have traveled from to see the events at Iowa state football or basketball games and the ratings they are providing. It's important to understand the ratings from customer feedback because it allows us to create and develop a new system that generates a newer and more efficient wait time.

Query 6:

```
Select Customer.CustomerName, Ticket.PurchaseDate, Ticket.TicketType, Ticket.TicketSectionNo, Ticket.TicketRowNo, Ticket.TicketSeatNo, ` Time` .WaitTime

From Customer, Ticket, ` Time` , feedback, ` Order`

Where Customer.CustomerName = ‘John Doe’ and Customer.CustomerID = ` Order` .CustomerID

And ` Order` .OrderID = Ticket.OrderID and ticket.TicketID = ` Time` .TicketID and Customer.CustomerID = feedback.CustomerID;
```



```
Select Customer.CustomerName, Ticket.PurchaseDate, Ticket.TicketType, Ticket.TicketSectionNo, Ticket.TicketRowNo, Ticket.TicketSeatNo, `Time`.WaitTime,
feedback.FeedbackRating, feedback.FeedbackComment
from Customer, Ticket, `Time`, feedback, `Order`
where Customer.CustomerName = 'John Doe' and Customer.CustomerID = `Order`.CustomerID
and `Order`.OrderID = Ticket.OrderID and ticket.TicketID = `Time`.TicketID and Customer.CustomerID = feedback.CustomerID;
```

CustomerName	PurchaseDate	TicketType	TicketSectionNo	TicketRowNo	TicketSeatNo	WaitTime	FeedbackRating	FeedbackComment
John Doe	2024-04-01	Football	102	D	5	00:04:00	5	Excellent experience overall.

Purpose: This query helps ISU be able to go and see what specific customers had to say for feedback, and it shows their seat and feedback. This allows ISU to look at both positive and negative individual feedback to see what their experience was and try to point at why their feedback might be negative overall.

Conclusion & Issues

The purpose of this research on Iowa States ticketing system was to analyze how customers and students favor the updated system that Iowa State implemented. We looked at factors as to why there are still complications about wait times during football and basketball games. These factors concluded that entrance points, weather conditions, ticket processing times, and consumer check-ins to why wait times are still a problem.

Our team developed a new system to view all the problems occurring when it narrows down to waiting times. We created a rational database diagram to analyze all entities, attributes, and relationships to identify every aspect we would need to focus on when creating our new system for Iowa State. These relationships are crucial to decreasing wait times, some include customer feedback understand if there's on going issues with wait times, viewing and pinpointing different entrance points that take longer to process customers and most importantly how weather conditions are affecting these times. We also developed queries to track customer and student feedback, able to identify different wait times among customers and their gate entrance, seeing if weather conditions are affecting wait times more than they should be.

Creating a new Iowa State ticketing system allowed an efficient and reliable way to increase customer and student satisfaction. Rather than having repetitive issues occurring we were able to develop a system that increases wait time which is convenient for every party involved such as the event itself, employees, students, and customers.