

## **Arcade Membership Database**

CIS 3400 (EMWA) - Database Management Systems  
Group 5

Esme Gonzalez - [esme.gonzalez@baruchmail.cuny.edu](mailto:esme.gonzalez@baruchmail.cuny.edu)

Julie Margolin - [julie.margolin@baruchmail.cuny.edu](mailto:julie.margolin@baruchmail.cuny.edu)

Jason Warm - [jason.warm@baruchmail.cuny.edu](mailto:jason.warm@baruchmail.cuny.edu)

Jacky Chen - [jacky.chen5@baruchmail.cuny.edu](mailto:jacky.chen5@baruchmail.cuny.edu)

Rafiuzzaman - [fnu.rafiuzzaman@baruchmail.cuny.edu](mailto:fnu.rafiuzzaman@baruchmail.cuny.edu)

## 1. Narrative

This project aims to track the operations at a local arcade. Our arcade requires that all customers will reserve beforehand to avoid capacity overload, thus collecting necessary demographic information. The customers will download an app on their phone or receive a game card upon entry that is assigned to their unique user. Customers will then have to scan a **barcode** (this is identical to their CustomerIDs) on their app or swipe their game card on each machine they intend to use to activate it. This will then **track** which games were played and by who, how long the game was played for, and how many points were scored. The database achieved will help the arcade be more efficient in its operations and track demand for games. For example, if data shows a game is not really being played, the arcade will remove it. If a game is really popular, the arcade will purchase a second one so multiple people can play at once. It will also be easier to figure out if a machine is broken (i.e. **data shows 0 people/hours played**). Lastly, a section of the database will track **employee information** to optimize billing. Employees will also handle maintenance requests, especially for broken games.

## 2. Information Needs

The information that would help the arcade's efficiency would be the information that the customer would provide when they sign-up for a membership. This includes name, date of birth, address, phone number, gender, etc. The information about the customers will help us collect the necessary demographic information and also help our loyalty reward system. Information about the arcade games that are being played is also crucial, information such as points, price, the average daily plays (updated in the database once a month), how long the game is played for shows the games' popularity. We would need our employees' information such as address, phone number, email, pay-rate, etc. We also want to track each employees' shifts and how often/long they work to reward them (with bonuses, for example) accordingly. Finally, we will track the maintenance requests, giving them unique identifiers and collect information such as category, date, estimated cost, actual cost, description, status, etc.

## 3. Initial List of Entities

### Customer:

- CustomerID
- FirstName
- LastName
- MiddleInitial
- Street
- City
- State

- PostalCode
- PhoneNumber
- Email
- DOB
- Gender
- CreditCardNo
- CreditCardExpirationDate
- CreditCardSecurityCode
- PointTotal

#### Employee

- EmployeeID
- FirstName
- LastName
- MiddleInitial
- SSN
- Street
- City
- State
- PostalCode
- PayRate
- Email
- PhoneNumber
- DOB
- Shifts

#### Reservation

- ReservationID
- ReservationDate
- ReservationTime

#### Arcade Games

- GameID
- DurationPerDay
- DateOfInstallation
- Highscore
- Price

#### GameStats

- TimePlayed

- GameScore
- HighScore
- TotalPlays

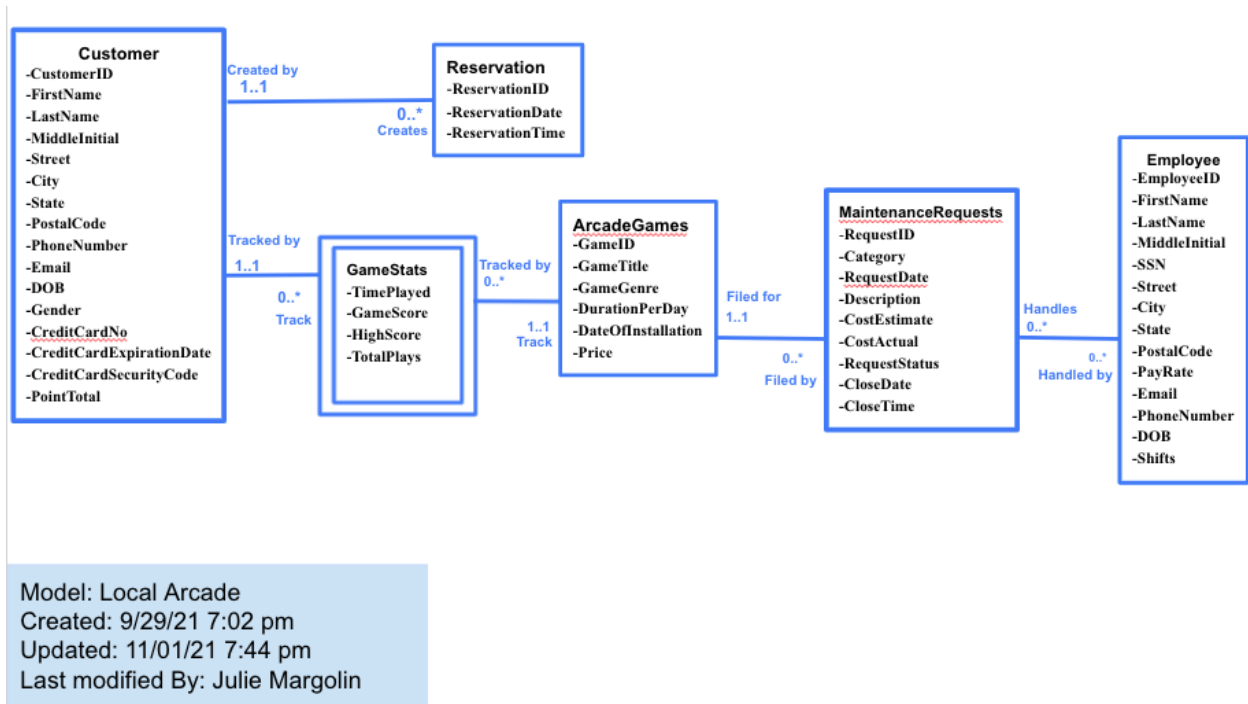
#### Maintenance Requests

- RequestID
- Category
- RequestDate
- Description
- CostEstimate
- CostActual
- RequestStatus
- CloseDate
- CloseTime

#### 4. Distribution of duties

- ❖ System Analysis - **Esme Gonzalez**
- ❖ Logical and Physical Modeling - **Rafiuzzaman**
- ❖ Database Implementation - **Jason Warm**
- ❖ Application Implementation - **Jacky Chen**
- ❖ Write up Final Report - **Julie Margolin**

## ER Model:



### 1. Customer and Reservation

One customer (0) may create by one or more (\*) reservations

One reservation (1) must be created by for one and only one (1) customer

### 2. Customer and GameStats

One Customer (0) may be tracked by one or more (\*) gamestats

One gamestat (1) must track one and only one (1) customer

### 3. GameStats and Arcade Games

One arcade game (0) may be tracked by one or more (\*) gamestats

One gamestat (1) must track one and only one (1) arcade game

### 4. Arcade Games and Maintenance

One arcade game (0) may have one or more (\*) maintenance requests

One maintenance request (1) must be filed for one and only one (1) arcade game

## **5. Maintenance and Employees**

One maintenance request (0) may be handled by one or more (\*) employees

One employee (0) may handle one or more (\*) maintenance requests

**Relational Model:**

Customer(CustomerID (key), FirstName, LastName, MiddleInitial, Street, City, State, PostalCode, PhoneNumber, Email, DOB, Gender, CreditCardNo, CreditCardExpirationDate, CreditCardSecurityCode, PointTotal)

GameStats(CustomerID (fk)(key), GameID (fk)(key), TimePlayed, GameScore, HighScore, TotalPlays)

Reservation(ReservationID (key), ReservationDate, ReservationTime, CustomerID (fk) )

ArcadeGames(GameID (key), GameTitle, GameGenre, DurationPerPlay, DateOfInstallation, Price)

MaintenanceRequests(RequestID (key), Category, RequestDate, Description, CostEstimate, CostActual, RequestStatus, CloseDate, CloseTime, GameID (fk) )

Employee(EmployeeID (key), FirstName, LastName, MiddleInitial, SSN, Street, City, State, PostalCode, PayRate, Email, PhoneNumber, DOB, Shifts)

MaintenanceRequests\_Employee(RequestID (fk)(key), EmployeeID (fk)(key) )

## Normalization:

Lavender = Original relation from relational model

Blue = Relation after split

---

Customer(CustomerID (key), FirstName, LastName, MiddleInitial, Street, City, State, PostalCode, PhoneNumber, Email, DOB, Gender, CreditCardNo, CreditCardExpirationDate, CreditCardSecurityCode, PointTotal)

Key: CustomerID

FD1: CustomerID  $\rightarrow$  FirstName, LastName, MiddleInitial, Street, City, State, PostalCode, PhoneNumber, Email, DOB, Gender, CreditCardNo, CreditCardExpirationDate, CreditCardSecurityCode, PointTotal

FD2: PostalCode  $\rightarrow$  City, State

**Customer is in 1NF because it is given as a relation. Customer is in 2NF because the key attribute determines all of the non-key attributes. Customer is not in 3NF because FD2 is a transitive dependency.**

Solution: Split the relation

Split 1:

Customer(CustomerID (key), FirstName, LastName, MiddleInitial, Street, REMOVE City, REMOVE State, COPY PostalCode, PhoneNumber, Email, DOB, Gender, CreditCardNo, CreditCardExpirationDate, CreditCardSecurityCode, PointTotal)

### New relation 1:

Customers(CustomerID (key), FirstName, LastName, MiddleInitial, Street, PostalCode, PhoneNumber, Email, DOB, Gender, CreditCardNo, CreditCardExpirationDate, CreditCardSecurityCode, PointTotal)

Key: CustomerID

FD1: CustomerID  $\rightarrow$  FirstName, LastName, MiddleInitial, Street, PostalCode, PhoneNumber, Email, DOB, Gender, CreditCardNo, CreditCardExpirationDate, CreditCardSecurityCode, PointTotal

**Customers is in 1NF because it was split from a relation. Customers is in 2NF because the key attribute determines all of the non-key attributes. Customers is in 3NF because there are no transitive dependencies.**



New relation 2:

**CustomerLocations**(PostalCode, City, State)

Key: PostalCode

FD1: PostalCode  $\rightarrow$  City, State

**Locations is in 1NF because it was split from a relation. Locations is in 2NF because the key attribute determines all of the non-key attributes. Locations is in 3NF because there are no transitive dependencies.**

**NOTE\*\*: We have chosen to de-normalize the Customer relation for simplicity in physical modeling. The new relations shown above will not be included as final relations.**

**GameStats**(CustomerID (fk)(key), GameID (fk)(key), TimePlayed, GameScore, HighScore, TotalPlays)

Key: CustomerID, GameID

FD1: CustomerID, GameID  $\rightarrow$  TimePlayed, GameScore, HighScore, TotalPlays

**GameStats is in 1NF because it was split from a relation. GameStats is in 2NF because the key attribute determines all of the non-key attributes. GameStats is in 3NF because there are no transitive dependencies. No need to normalize.**

**Reservation**(ReservationID, ReservationDate, ReservationTime, CustomerID (fk))

Key: ReservationID, CustomerID

FD1: ReservationID  $\rightarrow$  ReservationDate, ReservationTime

**Reservation is in 1NF because it was given as a relation. Reservation is in 2NF because the key attribute determines all of the non-key attributes. Reservation is in 3NF because there are no transitive dependencies. Therefore, Reservation is fully normalized to 3NF. There is no need for any splits.**

**ArcadeGames**(GameID, GameTitle, GameGenre, DurationPerPlay, DateOfInstallation, Price)

Key: GameID

FD1: GameID  $\rightarrow$  GameTitle, GameGenre, DurationPerPlay, DateOfInstallation, Price

**ArcadeGames is in 1NF because it was split from a relation. ArcadeGames is in 2NF because the key attribute determines all of the non-key attributes. ArcadeGames is in 3NF because there are no transitive dependencies. No normalization is needed.**

MaintenanceRequests(RequestID, Category, RequestDate, Description, CostEstimate, CostActual, RequestStatus, CloseDate, CloseTime, GameID(fk) )

Key: RequestID, GameID

FD1: RequestID, GameID → Category, RequestDate, Description, CostEstimate, CostActual, RequestStatus, CloseDate, CloseTime

FD2: GameID, Description → CostEstimate

**MaintenanceRequests is in 1NF because it was given as a relation. MaintenanceRequests is not in 2NF because GameID on its own determines CostEstimate. MaintenanceRequests is not in 3NF because FD2 is a transitive dependency. Solution is to split.**

MaintenanceRequests(RequestID, Category, RequestDate, COPY Description, REMOVE CostEstimate, CostActual, RequestStatus, CloseDate, CloseTime, COPY GameID)

**New Relation 3:**

Maintenance(RequestID, Category, RequestDate, Description, CostActual, RequestStatus, CloseDate, CloseTime, GameID)

Key: RequestID, GameID

FD1: RequestID, GameID → Category, RequestDate, Description, CostActual, RequestStatus, CloseDate, CloseTime

**Maintenance is in 1NF because it was given as a relation. Maintenance is in 2NF because the key attribute determines all of the non-key attributes. MaintenanceRequests is in 3NF because there are no transitive dependencies.**

**New Relation 4:**

MaintenanceDescription(GameID, Description, CostEstimate)

Key: Description, GameID

FD1: GameID, Description → CostEstimate

**MaintenanceDescription is in 1NF because it was given as a relation. MaintenanceDescription is in 2NF because the key attribute determines all of the non-key attributes. MaintenanceDescription is in 3NF because there are no transitive dependencies.**

**Employee**(EmployeeID , FirstName, LastName, MiddleInitial, SSN, Street, City, State, PostalCode, PayRate, Email, PhoneNumber, DOB, Shifts)

Key: EmployeeID

FD1: EmployeeID  $\rightarrow$  FirstName, LastName, MiddleInitial, SSN, Street, City, State, PostalCode, PayRate, Email, PhoneNumber, DOB, Shifts

FD2: PostalCode  $\rightarrow$  City, State

FD3: SSN  $\rightarrow$  FirstName, LastName, MiddleInitial, DOB

**Employee is in 1NF because it was split from a relation. Employee is in 2NF because the key attribute determines all of the non-key attributes. Employee is not in 3NF because FD2 and FD3 are transitive dependencies.**

Solution:

Employee(EmployeeID, FirstName, LastName, MiddleInitial, SSN, Street, REMOVE City, REMOVE State, COPY PostalCode , PayRate, Email, PhoneNumber, DOB, Shifts)

**New Relation 5:**

**Staff**(EmployeeID , FirstName, LastName, MiddleInitial, SSN, Street, PostalCode , PayRate, Email, PhoneNumber, DOB, Shifts)

Key: EmployeeID

FD1: EmployeeID  $\rightarrow$  FirstName, LastName, MiddleInitial, SSN, Street, PostalCode, PayRate, Email, PhoneNumber, DOB, Shifts

FD2: SSN  $\rightarrow$  FirstName, LastName, MiddleInitial, DOB

**Staff is in 1NF because it was split from a relation. Staff is in 2NF because the key attribute determines all of the non-key attributes. Staff is not in 3NF because FD2 is a transitive dependency.**

**New Relation 6:**

**EmployeeLocation**(PostalCode,City,State)

Key: PostalCode

FD1: PostalCode  $\rightarrow$  City, State

**EmployeeLocation is in 1NF because it was given as a relation. EmployeeLocation n is in 2NF because the key attribute determines all of the non-key attributes. EmployeeLocation is in 3NF because there are no transitive dependencies.**

**New Relation 7:**

**StaffInfo**(SSN, FirstName, LastName, MiddleInitial, DOB)

Key: SSN

FD1: SSN  $\rightarrow$  FirstName, LastName, MiddleInitial, DOB

**StaffInfo is in 1NF because it was given as a relation. StaffInfo is in 2NF because the key attribute determines all of the non-key attributes. StaffInfo is in 3NF because there are no transitive dependencies.**

#### New Relation 8:

**EmployeeInfo**(EmployeeID SSN, Street, PostalCode, PayRate, Email, PhoneNumber, Shifts)

Key: EmployeeID

FD1: EmployeeID  $\rightarrow$  SSN, Street, PostalCode, PayRate, Email, PhoneNumber, Shifts

**EmployeeInfo is in 1NF because it was given as a relation. EmployeeInfo is in 2NF because the key attribute determines all of the non-key attributes. EmployeeInfo is in 3NF because there are no transitive dependencies.**

**NOTE\*\*: We have chosen to de-normalize the Employee relation for simplicity in physical modeling. The new relations shown above will not be included as final relations.**

**MaintenanceRequests\_Employee**(RequestID (fk)(key), EmployeeID (fk)(key) )

Key: RequestID, EmployeeID

**MaintenanceRequests\_Employee is in 3NF because all attributes are key attributes, thus there are no functional dependencies. In other words, it is an “All Key” relation. It does not need to be normalized further.**

## FINAL SET OF RELATIONS:

**Customer**(CustomerID, FirstName, LastName, MiddleInitial, Street, City, State, PostalCode, PhoneNumber, Email, DOB, Gender, CreditCardNo, CreditCardExpirationDate, CreditCardSecurityCode, PointTotal)

Key: CustomerID

FD1: CustomerID  $\rightarrow$  FirstName, LastName, MiddleInitial, Street, City, State, PostalCode, PhoneNumber, Email, DOB, Gender, CreditCardNo, CreditCardExpirationDate, CreditCardSecurityCode, PointTotal

FD2: PostalCode  $\rightarrow$  City, State

**GameStats**(CustomerID, GameID, TimePlayed, GameScore, HighScore, TotalPlays)

Key: CustomerID, GameID

FD1: CustomerID, GameID  $\rightarrow$  TimePlayed, GameScore, HighScore, TotalPlays

**Reservation**(ReservationID, ReservationDate, ReservationTime, CustomerID)

Key: ReservationID, CustomerID

FD1: ReservationID  $\rightarrow$  ReservationDate, ReservationTime

**ArcadeGames**(GameID, GameTitle, GameGenre, DurationPerPlay, DateOfInstallation, Price)

Key: GameID

FD1: GameID  $\rightarrow$  GameTitle, GameGenre, DurationPerPlay, DateOfInstallation, Price

**Maintenance**(RequestID, Category, RequestDate, Description, CostActual, RequestStatus, CloseDate, CloseTime, GameID)

Key: RequestID, GameID

FD1: RequestID, GameID  $\rightarrow$  Category, RequestDate, Description, CostActual, RequestStatus, CloseDate, CloseTime

**MaintenanceDescription**(GameID, Description, CostEstimate)

Key: Description, GameID

FD1: GameID, Description  $\rightarrow$  CostEstimate

**Employee**(EmployeeID, FirstName, LastName, MiddleInitial, SSN, Street, City, State, PostalCode, PayRate, Email, PhoneNumber, DOB, Shifts)

Key: EmployeeID

FD1: EmployeeID  $\rightarrow$  FirstName, LastName, MiddleInitial, SSN, Street, City, State, PostalCode, PayRate, Email, PhoneNumber, DOB, Shifts

FD2: PostalCode  $\rightarrow$  City, State

FD3: SSN  $\rightarrow$  FirstName, LastName, MiddleInitial, DOB

**MaintenanceRequests\_Employee**(RequestID, EmployeeID)

Key: RequestID, EmployeeID

## **Physical Database Implementation:**

### **CREATE TABLES:**

```
CREATE TABLE Customer
(
  CustomerID VARCHAR(10) NOT NULL,
  FirstName VARCHAR(25) NOT NULL,
  LastName VARCHAR(35) NOT NULL,
  MiddleInitial VARCHAR(1),
  Street VARCHAR(25),
  City VARCHAR(20),
  State VARCHAR(2),
  PostalCode VARCHAR(10),
  PhoneNumber VARCHAR(21),
  Email VARCHAR(35),
  DOB DATE,
  Gender VARCHAR(10),
  CreditCardNo NUMBER,
  CreditCardExpirationDate DATE,
  CreditCardSecurityCode NUMBER,
  PointTotal NUMBER,
  CONSTRAINT pk_customer
    PRIMARY KEY (CustomerID)
)

CREATE TABLE GameStats
(
  CustomerID VARCHAR(10) NOT NULL,
  GameID VARCHAR(10) NOT NULL ,
  TimePlayed TIME,
  GameScore NUMBER,
  HighScore NUMBER,
  TotalPlays NUMBER,
  CONSTRAINT pk_GameStats
    PRIMARY KEY (CustomerID, GameID)
)

CREATE TABLE Reservation
(
  ReservationID VARCHAR(10) NOT NULL,
```

```
ReservationDate DATE,  
ReservationTime TIME,  
CustomerID VARCHAR(10),  
CONSTRAINT pk_Reservation  
    PRIMARY KEY (ReservationID)  
)
```

```
CREATE TABLE ArcadeGames  
(  
GameID VARCHAR(10) NOT NULL,  
GameTitle VARCHAR(35),  
GameGenre VARCHAR(35),  
DurationPerPlay VARCHAR(20),  
DateOfInstallation DATE,  
Price CURRENCY,  
CONSTRAINT pk_ArcadeGames  
    PRIMARY KEY (GameID)  
)
```

```
CREATE TABLE Maintenance  
(  
RequestID VARCHAR(10) NOT NULL,  
Category VARCHAR(35),  
RequestDate DATE,  
Description VARCHAR(75),  
CostActual NUMBER,  
RequestStatus VARCHAR(20),  
CloseDate DATE,  
CloseTime TIME,  
GameID VARCHAR(10),  
CONSTRAINT pk_Maintenance  
    PRIMARY KEY (RequestID)  
)
```

```
CREATE TABLE MaintenanceDescription  
(  
GameID VARCHAR(10) NOT NULL,  
Description VARCHAR(75),  
CostEstimate NUMBER,  
CONSTRAINT pk_MaintenanceDescription  
    PRIMARY KEY (GameID, Description)
```

)

```
CREATE TABLE Employee
(
EmployeeID VARCHAR(10) NOT NULL ,
FirstName VARCHAR(25) NOT NULL,
LastName VARCHAR(35) NOT NULL,
MiddleInitial VARCHAR(1),
SSN VARCHAR(15) NOT NULL,
Street VARCHAR(25),
City VARCHAR(20),
State VARCHAR(2),
PostalCode VARCHAR(10),
PayRate CURRENCY,
Email VARCHAR(35),
PhoneNumber VARCHAR(20),
DOB DATE,
Shifts NUMBER,
CONSTRAINT pk_Employee
    PRIMARY KEY (EmployeeID)
)
```

```
CREATE TABLE MaintenanceRequests_Employee
(
RequestID VARCHAR(10) NOT NULL,
EmployeeID VARCHAR(10) NOT NULL,
CONSTRAINT pk_MaintenanceRequests_Employee
    PRIMARY KEY (RequestID, EmployeeID)
)
```

### ALTER TABLES:

```
ALTER TABLE GameStats
ADD CONSTRAINT fk_GameStats_Customer
FOREIGN KEY (CustomerID)
REFERENCES Customer
(CustomerID)
```

```
ALTER TABLE GameStats
ADD CONSTRAINT fk_GameStats_ArcadeGames
FOREIGN KEY (GameID)
```



```
REFERENCES ArcadeGames  
(GameID)
```

```
ALTER TABLE Reservation  
ADD CONSTRAINT fk_Reservation_Customer  
FOREIGN KEY (CustomerID)  
REFERENCES Customer  
(CustomerID)
```

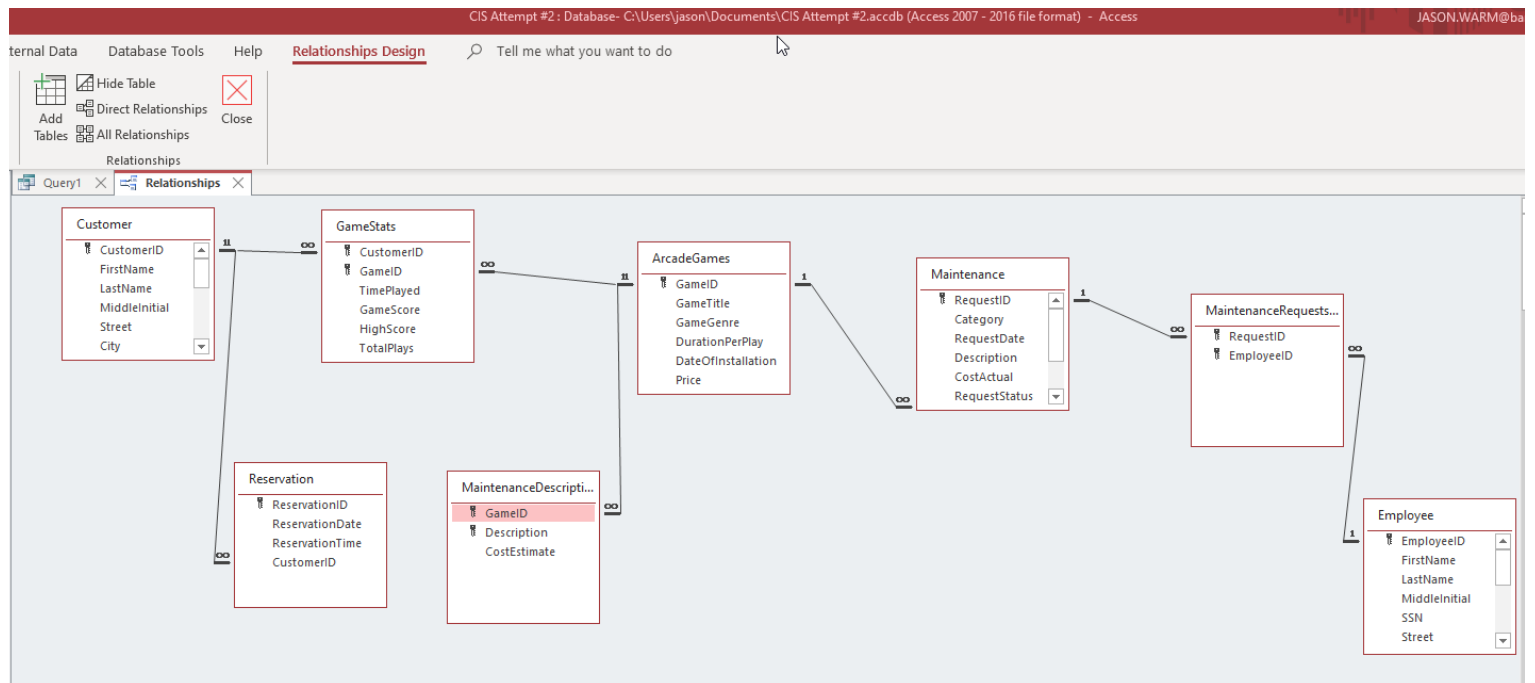
```
ALTER TABLE Maintenance  
ADD CONSTRAINT fk_Maintenance_ArcadeGames  
FOREIGN KEY (GameID)  
REFERENCES ArcadeGames  
(GameID)
```

```
ALTER TABLE MaintenanceRequests_Employee  
ADD CONSTRAINT fk_MaintenanceRequests_Employee_Maintenance  
FOREIGN KEY (RequestID)  
REFERENCES Maintenance  
(RequestID)
```

```
ALTER TABLE MaintenanceRequests_Employee  
ADD CONSTRAINT fk_MaintenanceRequests_Employee_Employee  
FOREIGN KEY (EmployeeID)  
REFERENCES Employee  
(EmployeeID)
```

```
ALTER TABLE MaintenanceDescription  
ADD CONSTRAINT fk_MaintenanceDescription_ArcadeGames  
FOREIGN KEY (GameID)  
REFERENCES ArcadeGames  
(GameID)
```

## RELATIONSHIPS AFTER ALTER TABLES:



## INSERT STATEMENTS:

```
INSERT INTO Customer VALUES
('C101', 'Malcom', 'Reid', 'A', '8989 Smith Rd', 'Brooklyn',
'NY', '11348', '347-219-8841', 'malcom.reid@gmail.com',
'07-17-1986', 'M', '9054356794033821', '07-01-2025', 123, 100);
```

```
INSERT INTO Customer VALUES
('C102', 'Eliot', 'Rez', 'L', '8129 Weyn Rd', 'Brooklyn', 'NY',
'11348', '432-123-4341', 'Eliot.rez@gmail.com', '06-20-1970',
'M', '3453674568123456', '06-01-2028', 345, 160);
```

```
INSERT INTO Customer VALUES
('C103', 'Anna', 'Webb', 'R', '2100 Baker Place', 'New York',
'NY', '11746', '574-345-8291', 'Anna.Webb@gmail.com',
'12-19-2000', 'F', '9538463728402845', '09-01-2026', 771, 189);
```

```
INSERT INTO Employee VALUES
('E101', 'Alice', 'Yee', 'M', '104-80-9941', '149 Price Ave',
'Staten Island', 'NY', '11824', 16, 'Alice.Yee@gmail.com',
'917-462-9714', '06-19-2000', 9);
```

```

INSERT INTO Employee VALUES
('E102', 'Benjamin', 'Forster', 'B', '506-78-3456', '4567
Clover Rd', 'Brooklyn', 'NY', '11348', '15',
'Benjamin.Forster@gmail.com', '678-435-76776', '06-20-1990',
'7');

INSERT INTO Employee VALUES
('E103', 'Garrett', 'Stein', 'T', '701-224-9174', '7598 Lake
Pond Rd', 'New Hyde Park', 'NY', '11746', '17',
'Garrett.Stein@gmail.com', '771-484-9997', '04-18-2002', '8');

INSERT INTO ArcadeGames VALUES
('G101', 'Pacman', 'Retro', '30 seconds', '01/21/2020', 0.50);

INSERT INTO ArcadeGames VALUES
('G102', 'Zombies', 'Horror', '45 seconds', '9/01/2021', .80);

INSERT INTO ArcadeGames VALUES
('G103', 'Guitar Hero', 'Music', '300 seconds', '03/15/2021',
2.00);

INSERT INTO GameStats VALUES
('C101', 'G101', '120 seconds', 1200, 6000, 10);

INSERT INTO GameStats VALUES
('C102', 'G102', '315 seconds', 5200, 10000, 23);

INSERT INTO GameStats VALUES
('C103', 'G103', '2400 seconds', 12948, 25000, 8);

INSERT INTO Reservation VALUES
('R101', '11/12/2021', '12pm', 'C101');

INSERT INTO Reservation VALUES
('R102', '11/12/2021', '10:24am', 'C102');

INSERT INTO Reservation VALUES
('R103', '11/16/2021', '6:30pm', 'C103');

INSERT INTO Maintenance VALUES

```

```
('M101', 'Screen repair', '9/12/2020', 'The screen is cracked at  
the top right corner', '200', 'Fixed', '9/18/2020', '9:52am',  
'G101');
```

```
INSERT INTO Maintenance VALUES  
('M102', 'Electrical Repair', '10/18/2021', 'Broken electrical  
wire', '65', 'Pending fix', NULL, NULL, 'G102');
```

```
INSERT INTO Maintenance VALUES  
('M103', 'Controller repair', '11/9/2021', 'Guitar button jammed  
on the red button', '145', 'Pending fix', NULL, NULL, 'G103');
```

```
INSERT INTO MaintenanceDescription VALUES  
('G101', 'The screen appears to be cracked at the top right  
corner', '120');
```

```
INSERT INTO MaintenanceDescription VALUES  
('G102', 'The Game appears to not power on due to either the  
electrical wire or the power button', '110');
```

```
INSERT INTO MaintenanceDescription VALUES  
('G103', 'The guitar button jammed on the red button', '145');
```

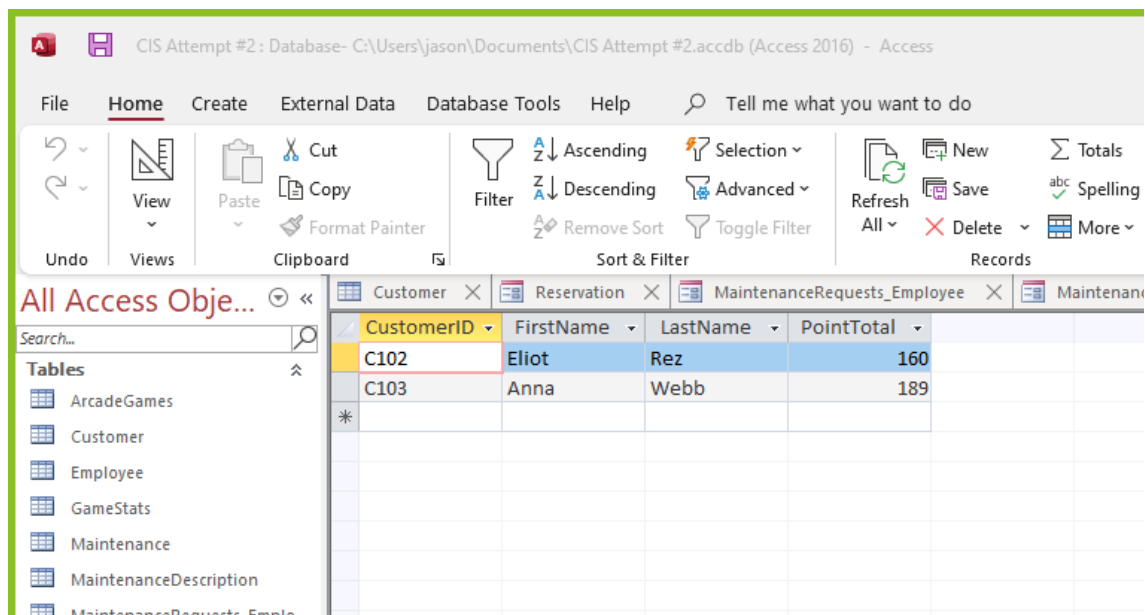
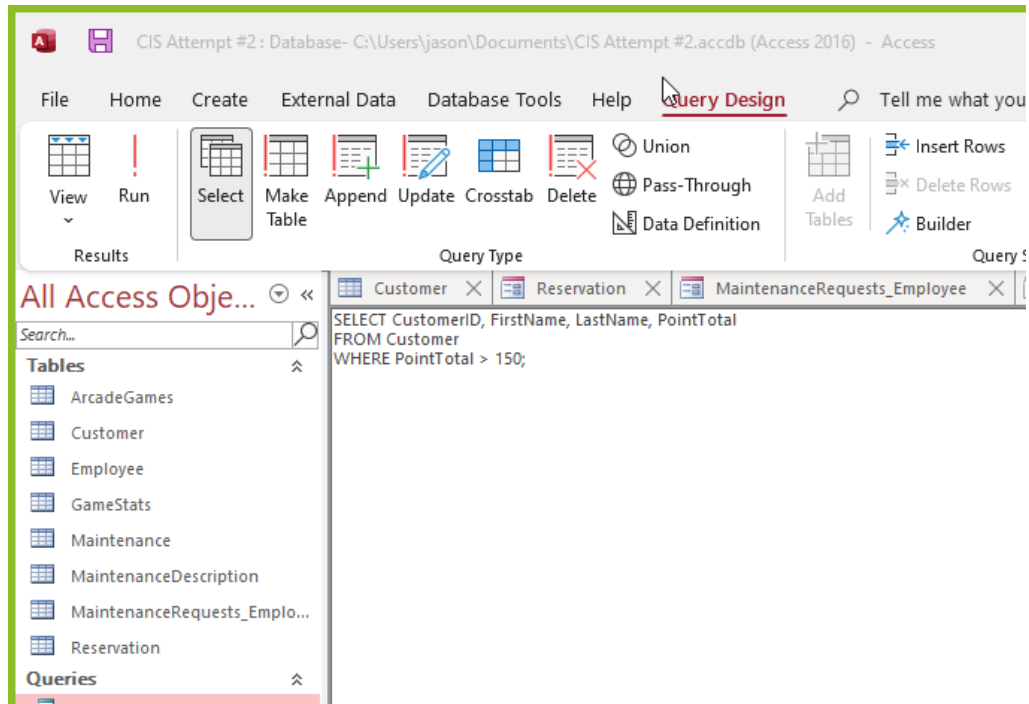
```
INSERT INTO MaintenanceRequests_Employee VALUES  
('M101', 'E101');
```

```
INSERT INTO MaintenanceRequests_Employee VALUES  
('M102', 'E102');
```

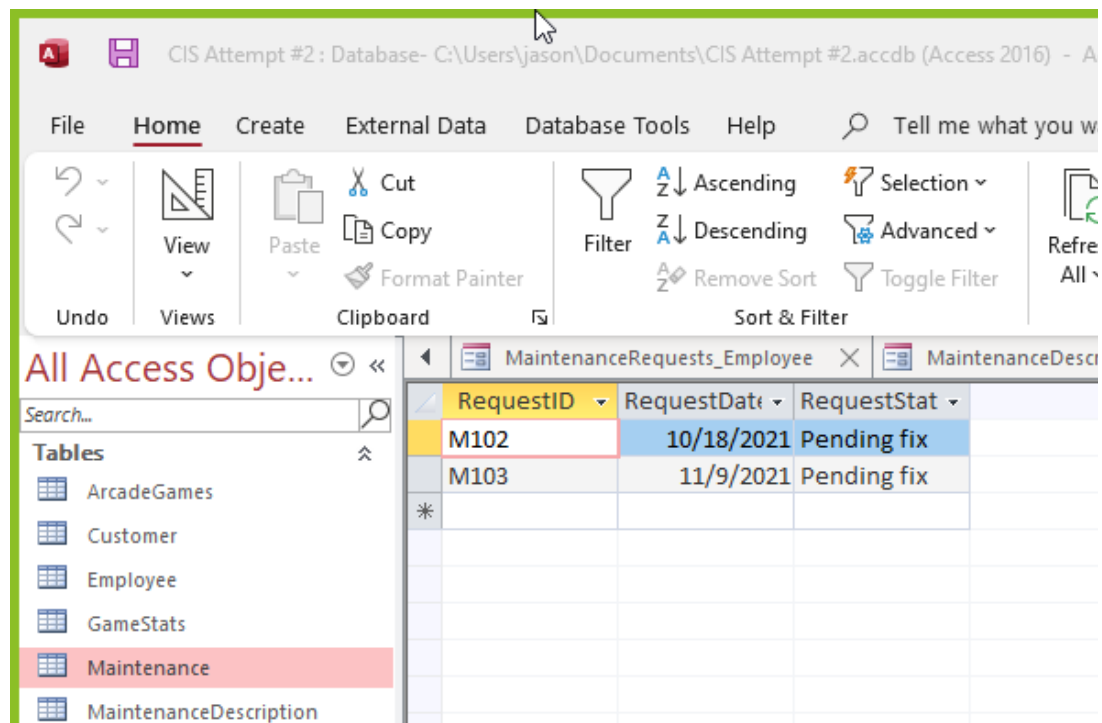
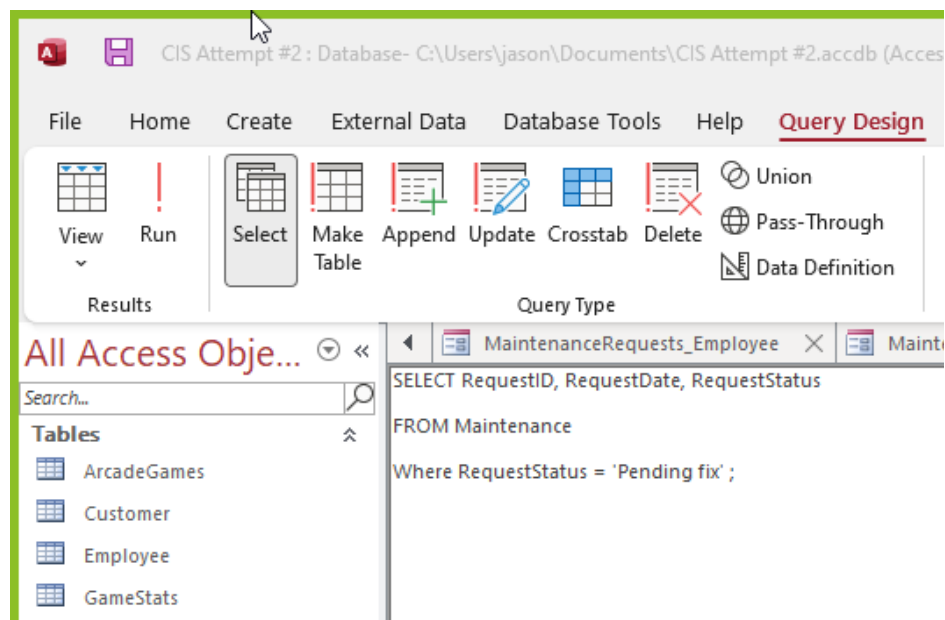
```
INSERT INTO MaintenanceRequests_Employee VALUES  
('M103', 'E103');
```

## Database Application:

Simple Query 1: Which customer(s) have a PointTotal greater than 150?



Simple Query 2: Show the request ID and dates of requests whose statuses are “Pending fix”



Simple Query 3: Identify the employees with pay rates less than 17 and at least 8 shifts

The image shows two screenshots of Microsoft Access. The top screenshot is in Query Design view, and the bottom screenshot is in Query Results view.

**Top Screenshot: Query Design View**

The title bar shows: CIS Attempt #2 : Database- C:\Users\jason\Documents\CIS Attempt #2.accdb (Access 2016) - Access

The ribbon includes: File, Home, Create, External Data, Database Tools, Help, and **Query Design**.

The Query Type section shows: View, Run, Select, Make Table, Append, Update, Crosstab, Delete, Union, Pass-Through, and Data Definition.

The left pane shows the **All Access Objects** task pane with a search bar. Under **Tables**, the following tables are listed: ArcadeGames, Customer, **Employee** (selected), GameStats, Maintenance, MaintenanceDescription, MaintenanceRequests\_Emplo..., and Reservation. Under **Queries**, Simple\_Query\_1 is listed.

The main area shows the SQL statement:

```
SELECT EmployeeID, FirstName, LastName, PayRate, Shifts
FROM Employee
WHERE PayRate < 17 AND Shifts >= 8 ;
```

**Bottom Screenshot: Query Results View**

The title bar shows: CIS Attempt #2 : Database- C:\Users\jason\Documents\CIS Attempt #2.accdb (Access 2016) - Access

The ribbon includes: File, **Home**, Create, External Data, Database Tools, Help, and Tell me what you want to do.

The ribbon sections include: Undo, Views, Clipboard (Cut, Copy, Paste, Format Painter), Sort & Filter (Filter, Ascending, Descending, Remove Sort, Selection, Advanced, Toggle Filter), and Records (Refresh, New, Save, Delete, Totals, Spelling, More).

The left pane shows the **All Access Objects** task pane with a search bar. Under **Tables**, the following tables are listed: ArcadeGames, Customer, **Employee** (selected), GameStats, Maintenance, MaintenanceDescription, MaintenanceRequests\_Emplo..., and Reservation. Under **Queries**, Simple\_Query\_1 is listed.

The main area shows the results of the query in a table:

EmployeeID	FirstName	LastName	PayRate	Shifts
E101	Alice	Yee	16	9

### Join Query 1: Identify the customer with the highest HighScore

```
SELECT c.CustomerID, c.FirstName, c.LastName, g.GameID, MAX(g.HighScore) AS MaxHighScore
FROM Customer AS c
INNER JOIN GameStats AS g
ON c.CustomerID = g.CustomerID
WHERE HighScore IN (
SELECT MAX(HighScore) AS MaxHighScore
FROM GameStats
)

GROUP BY c.CustomerID, c.FirstName, c.LastName, g.GameID;
```

CustomerID	FirstName	LastName	GameID	MaxHighScore
C103	Anna	Webb	G103	25000

### Customer With Highest Score

GameID	CustomerID	FirstName	LastName	MaxHighScore
G103	C103	Anna	Webb	25000



Join Query 2: Identify the most expensive game whose request status is “pending fix”

```
SELECT a.GameID, MAX(a.Price) AS MaxPrice, m.RequestStatus
FROM ArcadeGames AS a
INNER JOIN Maintenance AS m
ON a.GameID = m.GameID
WHERE m.RequestStatus = 'Pending Fix' AND Price IN (
SELECT MAX(Price)
FROM ArcadeGames
)
GROUP BY a.GameID, m.RequestStatus;
```

GameID	MaxPrice	RequestStatus
G103	\$2.00	Pending fix

## Highest Price And Pending Fix

GameID	MaxPrice	RequestStatus
G103	\$2.00	Pending fix

Join Query 3: Identify the customer and reservation ID of the customer who lives in Queens

```
SELECT c.CustomerID, c.FirstName, c.LastName, c.City, r.ReservationID
FROM Customer AS c
INNER JOIN Reservation AS r
ON c.CustomerID = r.CustomerID
WHERE c.City = 'Queens';
```

CustomerID	FirstName	LastName	City	Reservation
C103	Anna	Webb	Queens	R103
*				

### Customer City is Queens

CustomerID	FirstName	LastName	City	ReservationID
C103	Anna	Webb	Queens	R103

## ArcadeGames Data Entry Form

ArcadeGames	
GameID	<input type="text" value="G101"/>
GameTitle	<input type="text" value="Pacman"/>
GameGenre	<input type="text" value="Retro"/>
DurationPerPlay	<input type="text" value="30 seconds"/>
DateOfInstallation	<input type="text" value="1/21/2020"/>
Price	<input type="text" value="\$0.50"/>

## ArcadeGames Lookup Form

ArcadeGames Lookup	
GameID	<input type="text" value="G102"/>
GameTitle	<input type="text" value="Zombies"/>
GameGenre	<input type="text" value="Horror"/>
DurationPerPlay	<input type="text" value="45 seconds"/>
DateOfInstallation	<input type="text" value="9/1/2021"/>
Price	<input type="text" value="\$0.80"/>

## Customer Data Entry Form

Customer	
CustomerID	<input type="text" value="C101"/>
FirstName	<input type="text" value="Malcom"/>
LastName	<input type="text" value="Reid"/>
MiddleInitial	<input type="text" value="A"/>
Street	<input type="text" value="8989 Smith Rd"/>
City	<input type="text" value="Brooklyn"/>
State	<input type="text" value="NY"/>
PostalCode	<input type="text" value="11348"/>
PhoneNumber	<input type="text" value="347-219-8841"/>
Email	<input type="text" value="malcom.reid@gmail.com"/>
DOB	<input type="text" value="7/17/1986"/>
Gender	<input type="text" value="M"/>
CreditCardNo	<input type="text" value="9054356794033820"/>
CreditCardExpirationDate	<input type="text" value="7/1/2025"/>
CreditCardSecurityCode	<input type="text" value="123"/>
PointTotal	<input type="text" value="100"/>

## Customer Lookup Form

Customer	
CustomerID	<input type="text" value="C102"/> <input type="button" value="v"/>
FirstName	<input type="text" value="Eliot"/>
LastName	<input type="text" value="Rez"/>
MiddleInitial	<input type="text" value="L"/>
Street	<input type="text" value="8129 Weyn Rd"/>
City	<input type="text" value="Brooklyn"/>
State	<input type="text" value="NY"/>
PostalCode	<input type="text" value="11348"/>
PhoneNumber	<input type="text" value="432-123-4341"/>
Email	<input type="text" value="Eliot.rez@gmail.com"/>
DOB	<input type="text" value="6/20/1970"/>
Gender	<input type="text" value="M"/>
CreditCardNo	<input type="text" value="3453674568123456"/>
CreditCardExpirationDate	<input type="text" value="6/1/2028"/>
CreditCardSecurityCode	<input type="text" value="345"/>
PointTotal	<input type="text" value="160"/>

## Employee Data Entry Form

### Employee

EmployeeID	<input type="text" value="E101"/>
FirstName	<input type="text" value="Alice"/>
LastName	<input type="text" value="Yee"/>
MiddleInitial	<input type="text" value="M"/>
SSN	<input type="text" value="104-80-9941"/>
Street	<input type="text" value="149 Price Ave"/>
City	<input type="text" value="Staten Island"/>
State	<input type="text" value="NY"/>
PostalCode	<input type="text" value="11824"/>
PayRate	<input type="text" value="16"/>
Email	<input type="text" value="Alice.Yee@gmail.om"/>
PhoneNumber	<input type="text" value="917-462-9714"/>
DOB	<input type="text" value="6/19/2000"/>
Shifts	<input type="text" value="9"/>

## Employee lookup Form

### Employee Lookup

EmployeeID	<input type="text" value="E102"/>
FirstName	<input type="text" value="Benjamin"/>
LastName	<input type="text" value="Forster"/>
MiddleInitial	<input type="text" value="B"/>
SSN	<input type="text" value="506-78-4556"/>
Street	<input type="text" value="4567 Clover Rd"/>
City	<input type="text" value="Brooklyn"/>
State	<input type="text" value="NY"/>
PostalCode	<input type="text" value="11348"/>
PayRate	<input type="text" value="15"/>
Email	<input type="text" value="Benjamin.Forster@gmail.com"/>
PhoneNumber	<input type="text" value="678-435-7677"/>
DOB	<input type="text" value="6/20/1990"/>
Shifts	<input type="text" value="7"/>

## GameStats Data Entry Form

GameStats

GameStats

GameStats

### GameStats

CustomerID	<input type="text" value="C101"/>
GameID	<input type="text" value="G101"/>
TimePlayed	<input type="text" value="120 seconds"/>
GameScore	<input type="text" value="1200"/>
HighScore	<input type="text" value="6000"/>
TotalPlays	<input type="text" value="10"/>

## Maintenance Data Entry Form

Maintenance	
RequestID	M101
Category	Screen repair
RequestDate	9/12/2020
Description	The screen is cracked at the top right corner
CostActual	\$200.00
RequestStatus	Fixed
CloseDate	9/18/2020
CloseTime	9:52:00 AM
GameID	G101

## Maintenance Lookup Form

Maintenance Lookup	
RequestID	M103
Category	Controller Repair
RequestDate	11/9/2021
Description	Guitar button jammed on the red button
CostActual	\$145.00
RequestStatus	Pending fix
CloseDate	
CloseTime	
GameID	G103

## MaintenanceDescription Data Entry Form

MaintenanceDescription	
GameID	G101
Description	The screen appears to be cracked at the top right corner
CostEstimate	\$120.00

## MaintenanceRequests\_Employee Data Entry Form


MaintenanceRequests_Employee	
RequestID	M101
EmployeeID	E101



## Maintenance Assignment Form

[illegible]

## Reservation Data Entry Form



Reservation

ReservationID R101

ReservationDate 11/12/2021

ReservationTime 12:00:00 PM

CustomerID C101

## Reservation Data Entry Form With Customer Details

## Reservation Data Entry

CustomerID	C101
FirstName	Malcolm
LastName	Reid
PhoneNumber	347-219-8841
Email	malcom.reid@gmail.com

### Reservation

ReservationID	ReservationDate	ReservationTime
R101	11/12/2021	12:00:00 PM
*		

Record: 1 of 1
No Filter
Search

## Reservation with Lookup

## Reservation lookup

CustomerID	C102
FirstName	Eliot
LastName	Rez
PhoneNumber	432-123-4341
Email	Eliot.rez@gmail.com

## Reservation

[illegible]

## Customer Master Form

### CustomerMasterForm

CustomerID

First Name

Last Name

GameStats

C101

Malcom

Reid

GameID	TimePlayed	GameScore
G101	120 seconds	1200
*		

Record: 1 of 1

## Navigation form:

Navigation Form

Navigation Form

ArcadeGames

ArcadeGames Report

Customer

CustomerMasterForm

Customer With Highest Score

City Is Queens

Employee

GameStats

GameStats Subform

Reservation

Maintenance

MaintenanceDescription

Highest Price And Pending Fix

Main

CustomerMasterForm

CustomerID

First Name

Last Name

GameStats

C102

Eliot

Rez

GameID	TimePlayed	GameScore
G102	1035 seconds	5200
*		

Record: 1 of 1

## **Conclusion:**

For this project, we coordinated our activities by setting up meeting times and designating a member of the group to share screen over Zoom. While we understood that it may be less work individually to divide up tasks and do them on our own time, we found that we work better together and elected to complete each portion of the project as a group. We created the ER model through a shared PowerPoint and wrote all SQL code on MS Access. For the Access portion, one group member created and worked directly on the file while sharing screen so the rest of the group could participate.

Overall, our group had a good experience with the project. We appreciated that it was split up across the semester and followed what we were learning in the curriculum. This structure made it more doable than it would have been as a regular 2-3 week project. The most difficult steps were setting up the Access file and creating all the forms/code. Some of our issues came with syntax as DataCamp uses a different version of SQL, and none of us were used to Access. We also didn't have much experience creating forms so it took us a little while to figure that out. The easiest part of the project was creating entities and attributes once we had a proposal approved.

We didn't imagine we would be able to create an entire database from scratch. We learned that any business model can be made into a database if the steps we learned this semester are followed, but we didn't consider that even we, as students, could do it too. If we did it all over again, we probably would have added a few more entities and done more insert statements so we would have more to work with when doing the SQL code. In general, we are satisfied with our results and learned a great deal from this project.