



**Building a Scheduling Database for a Tennis Club**  
**CIS 9340 - UFA [55051]**  
**Principles of Database Management Systems**  
**Data Generation**

**Group members:**

1. Hugo Hasegawa - [Hugo.Hasegawa@baruchmail.cuny.edu](mailto:Hugo.Hasegawa@baruchmail.cuny.edu)
2. LeVietHoa Nguyen - [VietHoa.Nguyen@baruchmail.cuny.edu](mailto:VietHoa.Nguyen@baruchmail.cuny.edu)
3. Shameer Ali - [Shameer.Ali@baruchmail.cuny.edu](mailto:Shameer.Ali@baruchmail.cuny.edu)
4. Vrushank Shah - [Vrushank.Shah@baruchmail.cuny.edu](mailto:Vrushank.Shah@baruchmail.cuny.edu)
5. Duyen Nguyen - [Duyen.Nguyen@baruchmail.cuny.edu](mailto:Duyen.Nguyen@baruchmail.cuny.edu)

Cary Leeds Tennis Center, which was established in 2015 in the Bronx, New York, uses a File-based System with old spreadsheets to keep track of clients, appointments, payments, equipment, and maintenance. However, the old system cannot keep up with the company's success and the high number of demands. To accommodate for this, they have contacted our team to discuss options available. During our first meeting with them, they told us that the main issue with their file-based system is being unable to keep track of scheduled appointments. As a result, they constantly need to cancel appointments because of conflicting schedules. This in turn leads to a loss of revenue for the business, and this needs to be rectified as soon as possible.

After being informed of the issue, our team proposed that the best solution to the problem would be to implement a scheduling database. The database would be primarily focused on their scheduling operation, for which their revenue is derived. This will allow the club to keep track of their business in a more efficient and effective way. During the second meeting, they informed us that they have agreed to have our team "Data Generation" create the scheduling database system for the business.

Our team then enquired about the way in which the business is operated. In their tennis center, tennis players schedule practice sessions with their favorite coach or other coaches who are available at the requested day and time. Practice/Training sessions can either be private sessions, or group sessions. The club needs updated records within the database to show the availability of a coach and time available.

In addition, all coaches have different levels of experience, thus, the price range per hour per coach may vary. The club needs to also keep track of the materials utilized and available such as tennis balls, racket, cones, machine balls per session. This will provide a more accurate record of what materials are available for additional sessions.

Creating a scheduling database application for the business can be a tedious task. Therefore, the team needs to gather all relevant information necessary to complete the task. The information listed below are some of the information that will be required by our team, but it is not limited to only these:

- Number of coaches currently employed by the club
- Coaches level of experience (ratings format, 1-5)
- Materials available per session scheduled
- Timing of the day which sessions can be scheduled (club open hours, weekend, or weekday), also session time limits

During team discussion and meeting, we have decided that the below list of entities will be our foundation for which the database will be created.

Entities:

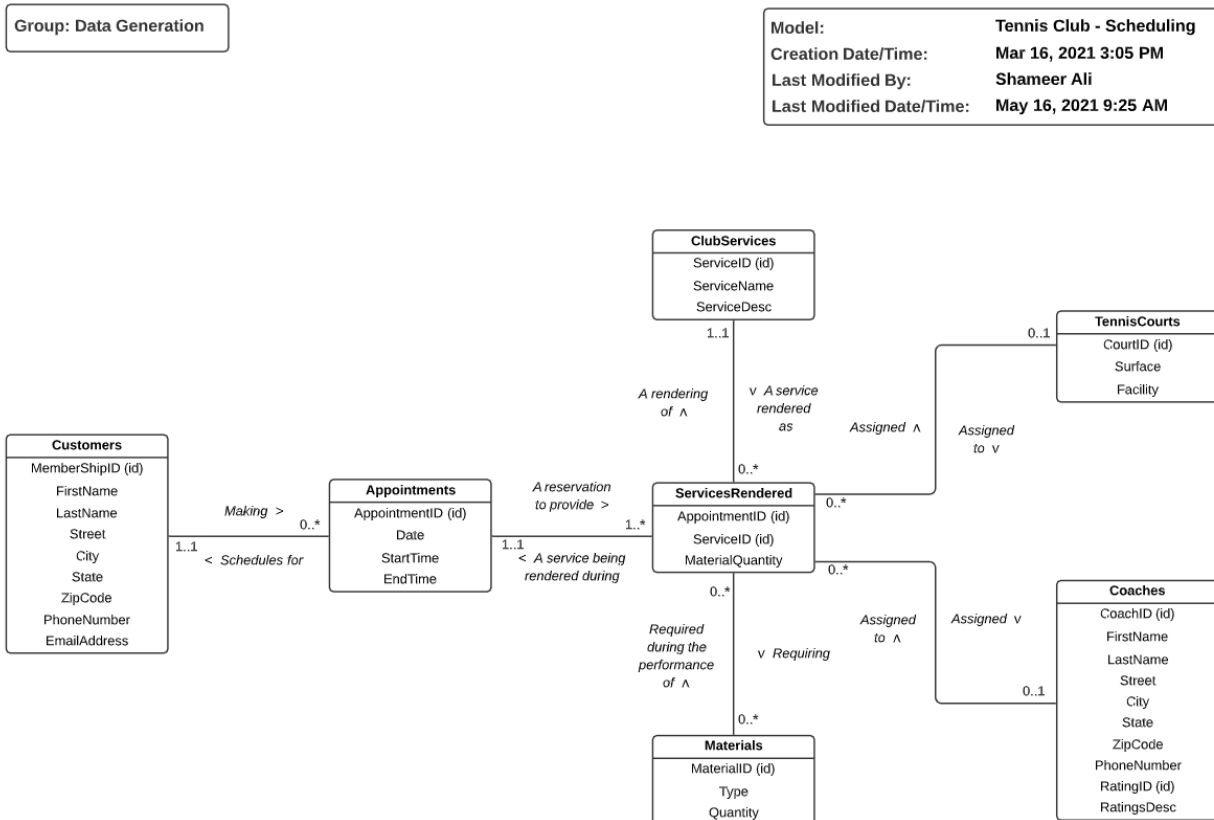
- Customers
- Coaches
- Appointments
- ClubServices

- ServicesRendered
- TennisCourts
- Materials

Since the team will be creating the database for the club, each team member has been assigned their respective roles listed below.

Team Data Generation – Members Role

- Hugo Hasegawa – Project Leader
- LeVietHoa Nguyen – Documentation Specialist
- Duyen Nguyen – Database Designer
- Shameer Ali – System Analyst
- Vrushank Shah – Application Developer

**Entity Relationship Model diagram****Relationship Sentences**

One **Customer** may be making one or more **Appointment**

One **Appointment** must be scheduled for one and only one **Customer**

One **Appointment** must be a reservation to provide one or more **ServicesRendered**

One **ServicesRendered** must be a service being rendered during one and only one **Appointment**

One **ClubServices** may be a service rendered as one or more **ServicesRendered**

One **ServicesRendered** must be a rendering of one and only one **ClubServices**

One **TennisCourt** may be assigned to one or more **ServicesRendered**

One **ServicesRendered** may be assigned to one and only one **TennisCourt**

One **Coach** *may be assigned to* one or more **ServicesRendered**

One **ServicesRendered** *may be assigned to* one and only one **Coach**

One **Material** *may be required during the performance of* one or more **ServicesRendered**

One **ServicesRendered** *may be requiring* one or more **Material**

### **Initial Relations**

**Customers** (MembershipID (**PK**), FirstName, LastName, Street, City, State, ZipCode, PhoneNumber, EmailAddress)

**PRIMARY KEY** MembershipID

**Appointments** (AppointmentID (**PK**), AppointmentDate, StartTime, EndTime, MemberShipID (**FK**))

**PRIMARY KEY** AppointmentID

**FOREIGN KEY** MemberShipID **REFERENCES** Customers (MembershipID)

**ClubServices** (ServiceID (**PK**), ServiceName, ServiceDesc)

**PRIMARY KEY** ServiceID

**TennisCourts** (CourtID (**PK**), Surface, Facility)

**PRIMARY KEY** CourtID

**Coaches** (CoachID (**PK**), FirstName, LastName, Street, City, State, ZipCode, PhoneNumber, RatingID, RatingDesc)

**PRIMARY KEY** CoachID

**Materials** (MaterialID (**PK**), Type, Quantity)

**PRIMARY KEY** MaterialID

**ServicesRendered** (ServiceRenderedID(**PK**), AppointmentID(**FK**), ServiceID(**FK**), CourtID(**FK**), CoachID(**FK**))

**PRIMARY KEY** ServiceRenderedID

**FOREIGN KEY** AppointmentID **REFERENCES** Appointments (AppointmentID)

**FOREIGN KEY** ServiceID **REFERENCES** ClubServices (ServiceID)

**FOREIGN KEY** CourtID **REFERENCES** TennisCourts (CourtID)

**FOREIGN KEY** CoachID **REFERENCES** Coaches (CoachID)

**ServicesRenderedMaterials** (ServiceRenderedID (**PK**)(**FK**), MaterialID (**PK**)(**FK**), MaterialQuantity)

**PRIMARY KEY** ServiceRenderedID, MaterialID

**FOREIGN KEY** ServiceRenderedID **REFERENCES** ServicesRendered (ServiceRenderedID)

**FOREIGN KEY** MaterialID **REFERENCES** Materials (MaterialID)

---

### End of Relations

---

## Normalization

This is the Normalization phase which is the process of minimizing redundancy from a relation or set of relations. Redundancy in relation may cause insertion, deletion and updating anomalies. So, it helps to minimize the redundancy in relations. Normal forms are used to eliminate or reduce redundancy in database tables.

By definition a 3NF has to be in a 2NF, and 1NF. We achieved our final tables by first normalizing our initial tables to 2NF. We got rid of all transitive functional dependencies. After this step, we finally normalized to 3NF by getting rid of all partial functional dependencies. We proceeded by splitting the necessary tables, and by creating new relations shown below.

**Customers** (MemberShipID (**PK**), FirstName, LastName, Street, City, State, ZipCode, PhoneNumber, EmailAddress)

**PRIMARY KEY** MembershipID

**FD1:** MemberShipID  $\rightarrow$  FirstName, LastName, Street, City, State, ZipCode, PhoneNumber, EmailAddress

**FD2:** ZipCode  $\rightarrow$  City, State

**1<sup>st</sup> NF:** Meets the definition of a relation

**2<sup>nd</sup> NF:** No partial key dependencies

**3<sup>rd</sup> NF:** Transitive dependencies exist

- MemberShipID  $\rightarrow$  ZipCode and ZipCode  $\rightarrow$  City, State

**Solution:** Split Customers relation into two new relations named CustomerInfo and ZipCodes.

**CustomersInfo** (MemberShipID (**PK**), FirstName, LastName, Street, ZipCode (**FK**), PhoneNumber, EmailAddress)

**PRIMARY KEY** MembershipID

**FD1:** MemberShipID  $\rightarrow$  FirstName, LastName, Street, ZipCode (**FK**), PhoneNumber, EmailAddress

**1<sup>st</sup> NF:** Meets the definition of a relation

**2<sup>nd</sup> NF:** No partial key dependencies

**3<sup>rd</sup> NF:** No Transitive dependencies exist

**ZipCodes** (ZipCode (**PK**), City, State)  
**PRIMARY KEY** ZipCode

**FD1:** ZipCode  $\rightarrow$  City, State

**1<sup>st</sup> NF:** Meets the definition of a relation

**2<sup>nd</sup> NF:** No partial key dependencies

**3<sup>rd</sup> NF:** No Transitive dependencies exist

---

**Appointments** (AppointmentID (**PK**), AppointmentDate, StartTime, EndTime, MemberShipID (**FK**))

**PRIMARY KEY** AppointmentID

**FD1:** AppointmentID  $\rightarrow$  AppointmentDate, StartTime, EndTime, MemberShipID (**FK**)

**1<sup>st</sup> NF:** Meets the definition of a relation

**2<sup>nd</sup> NF:** No partial key dependencies

**3<sup>rd</sup> NF:** No Transitive dependencies exist

**Appointments** (AppointmentID (**PK**), AppointmentDate, StartTime, EndTime, MemberShipID (**FK**))

**PRIMARY KEY** AppointmentID

---

**ClubServices** (ServiceID (**PK**), ServiceName, ServiceDesc)

**PRIMARY KEY** ServiceID

**FD1:** ServiceID  $\rightarrow$  ServiceName, ServiceDesc

**1<sup>st</sup> NF:** Meets the definition of a relation

**2<sup>nd</sup> NF:** No partial key dependencies

**3<sup>rd</sup> NF:** No transitive dependencies exist

**ClubServices** (ServiceID (**PK**), ServiceName, ServiceDesc)

**PRIMARY KEY** ServiceID

---

**TennisCourts** (CourtID (**PK**), Surface, Facility)  
**PRIMARY KEY** CourtID

**FD1:** TennisCourtID  $\rightarrow$  Surface, Facility

**1<sup>st</sup> NF:** Meets the definition of a relation

**2<sup>nd</sup> NF:** No partial key dependencies

**3<sup>rd</sup> NF:** No Transitive dependencies exist

**TennisCourts** (CourtID (**PK**), Surface, Facility)  
**PRIMARY KEY** CourtID

---

**Coaches** (CoachID (**PK**), FirstName, LastName, Street, City, State, ZipCode, PhoneNumber, RatingID, RatingDesc)  
**PRIMARY KEY** CoachID

**FD1:** CoachID  $\rightarrow$  FirstName, LastName, Street, City, State, ZipCode, PhoneNumber, RatingID, RatingDesc

**FD2:** ZipCode  $\rightarrow$  City, State

**FD3:** RatingID  $\rightarrow$  RatingDesc

**1<sup>st</sup> NF:** Meets the definition of a relation

**2<sup>nd</sup> NF:** No partial key dependencies

**3<sup>rd</sup> NF:** Transitive dependencies exist

- CoachID  $\rightarrow$  ZipCode and ZipCode  $\rightarrow$  City, State
- CoachID  $\rightarrow$  RatingID and RatingID  $\rightarrow$  RatingDesc

**Solution:** Split Coaches relation into three new relations named, CoachesInfo, ZipCodes and Ratings

**Note:** We already created a ZipCodes when the Customer table was normalized. Therefore, we will reuse the ZipCodes relation already created.

**CoachesInfo** (CoachID (**PK**), FirstName, LastName, Street, ZipCode (**FK**), PhoneNumber)  
**PRIMARY KEY** CoachID

**FD1:** CoachID  $\rightarrow$  FirstName, LastName, Street, ZipCode (**FK**), PhoneNumber

**1<sup>st</sup> NF:** Meets the definition of a relation

**2<sup>nd</sup> NF:** No partial key dependencies

**3<sup>rd</sup> NF:** No transitive dependencies exist



**Ratings** (RatingID (**PK**), RatingDesc)  
**PRIMARY KEY** RatingID

**FD1:** RatingID  $\rightarrow$  RatingDesc

**1<sup>st</sup> NF:** Meets the definition of a relation

**2<sup>nd</sup> NF:** No partial key dependencies

**3<sup>rd</sup> NF:** No transitive dependencies exist

---

**Materials** (MaterialID (**PK**), Type, Quantity)  
**PRIMARY KEY** MaterialID

**FD1:** MaterialID  $\rightarrow$  Type, Quantity

**1<sup>st</sup> NF:** Meets the definition of a relation

**2<sup>nd</sup> NF:** No partial key dependencies

**3<sup>rd</sup> NF:** No transitive dependencies exist

**Materials** (MaterialID (**PK**), Type, Quantity)  
**PRIMARY KEY** MaterialID

---

**ServicesRendered** (ServiceRenderedID (**PK**), AppointmentID (**FK**), ServiceID (**FK**), CourtID (**FK**), CoachID (**FK**))  
**PRIMARY KEY** ServiceRenderedID

**1<sup>st</sup> NF:** Meets the definition of a relation

**2<sup>nd</sup> NF:** No partial key dependencies

**3<sup>rd</sup> NF:** No transitive dependencies exist

**ServicesRendered** (ServiceRenderedID (**PK**), AppointmentID (**FK**), ServiceID (**FK**), CourtID (**FK**), CoachID (**FK**))  
**PRIMARY KEY** ServiceRenderedID

---

**ServicesRenderedMaterials** (ServiceRenderedID (**PK**)(**FK**), MaterialID (**PK**)(**FK**), MaterialQuantity)  
**PRIMARY KEY** ServiceRenderedID, MaterialID

**1<sup>st</sup> NF:** Meets the definition of a relation

**2<sup>nd</sup> NF:** No partial key dependencies

**3<sup>rd</sup> NF:** No transitive dependencies exist

**ServicesRenderedMaterials** (ServiceRenderedID (**PK**)(**FK**), MaterialID (**PK**)(**FK**),  
MaterialQuantity)

**PRIMARY KEY** ServiceRenderedID, MaterialID

---

**End of Normalization**

---

**Tables after normalized to 3<sup>rd</sup> NORMAL FORM**

**CustomersInfo** (MembershipID (**PK**), FirstName, LastName, Street, ZipCode (**FK**),  
PhoneNumber, EmailAddress)

**PRIMARY KEY** MembershipID

**ZipCodes** (ZipCode (**PK**), City, State)

**PRIMARY KEY** ZipCode

**Appointments** (AppointmentID (**PK**), AppointmentDate, StartTime, EndTime, MemberShipID  
(**FK**))

**PRIMARY KEY** AppointmentID

**ClubServices** (ServiceID (**PK**), ServiceName, ServiceDesc)

**PRIMARY KEY** ServiceID

**TennisCourts** (TennisCourtID (**PK**), Surface, Facility)

**PRIMARY KEY** TennisCourtID

**CoachesInfo** (CoachID (**PK**), FirstName, LastName, Street, ZipCode (**FK**), PhoneNumber

**PRIMARY KEY** CoachID

**Ratings** (RatingID (**PK**), RatingDesc)

**PRIMARY KEY** RatingID

**Materials** (MaterialID (**PK**), Type, Quantity)

**PRIMARY KEY** MaterialID

**ServicesRendered** (ServiceRenderedID (**PK**), AppointmentID (**FK**), ServiceID (**FK**), CourtID  
(**FK**), CoachID (**FK**))

**PRIMARY KEY** ServiceRenderedID

**ServicesRenderedMaterials** (ServiceRenderedID (**PK**)(**FK**), MaterialID (**PK**)(**FK**),  
MaterialQuantity)

**PRIMARY KEY** ServiceRenderedID, MaterialID

---

**End of Total Tables**

---

**Physical Design****CustomersInfo**

Attribute Name	Data Type	MS Access Type	Required?	Length/Size	Constraints	Description/Purpose
MemberShipID	Integer	AUTOINCREMENT (X)	Yes	100,1	PK NOT NULL	Unique identifier for MemberShipID instance
FirstName	Variable Character	VARCHAR(X)	Yes	20	NOT NULL	Customer first name
LastName	Variable Character	VARCHAR(X)	Yes	20	NOT NULL	Customer last name
Street	Variable Character	VARCHAR(X)	Yes	40	NOT NULL	Customer street address
ZipCode	Character	CHAR(X)	Yes	5	FK NOT NULL	Unique identifier for ZipCode instance
PhoneNumber	Variable Character	VARCHAR(X)	Yes	20	N/A	Customer mobile number
EmailAddress	Variable Character	VARCHAR(X)	Yes	50	N/A	Customer email address

**ZipCodes**

Attribute Name	Data Type	MS Access Type	Required?	Length/Size	Constraints	Description/Purpose
ZipCode	Character	CHAR(X)	Yes	5	PK NOT NULL	Unique identifier for ZipCode instance
City	Variable Character	VARCHAR(X)	Yes	20	NOT NULL	City with respected ZipCode
State	Character	CHAR(X)	Yes	2	NOT NULL	State with respected ZipCode

**Appointments**

Attribute Name	Data Type	MS Access Type	Required?	Length/Size	Constraints	Description/Purpose
AppointmentID	Integer	AUTOINCREMENT (X)	Yes	200,1	PK NOT NULL	Unique identifier for AppointmentID instance
AppointmentDate	Date	DATE	Yes	N/A	NOT NULL	Appointment scheduled date
StartTime	Time	TIME	Yes	N/A	NOT NULL	Appointment start time
EndTime	Time	TIME	Yes	N/A	NOT NULL	Appointment end time
MemberShipID	Integer	LONG	Yes	N/A	FK NOT NULL	Unique identifier for MemberShipID instance

**ClubServices**

Attribute Name	Data Type	MS Access Type	Required?	Length/Size	Constraints	Description/Purpose
ServiceID	Integer	AUTOINCREMENT (X)	Yes	300,1	PK NOT NULL	Unique identifier for ServiceID instance
ServiceName	Variable Character	VARCHAR(X)	Yes	50	NOT NULL	Service name
ServiceDesc	Variable Character	VARCHAR(X)	Yes	100	NOT NULL	Service description

**TennisCourts**

Attribute Name	Data Type	MS Access Type	Required?	Length/Size	Constraints	Description/ Purpose
CourtID	Integer	AUTOINCREMENT (X)	Yes	400,1	PK NOT NULL	Unique identifier for CourtID instance
Surface	Variable Character	VARCHAR(X)	Yes	30	NOT NULL	Tennis court surface type
Facility	Variable Character	VARCHAR(X)	Yes	10	NOT NULL	Tennis court indoor/outdoor

**CoachesInfo**

Attribute Name	Data Type	MS Access Type	Required?	Length/Size	Constraints	Description/ Purpose
CoachID	Integer	AUTOINCREMENT (X)	Yes	500,1	PK NOT NULL	Unique identifier for CoachID instance
FirstName	Variable Character	VARCHAR(X)	Yes	20	NOT NULL	Coach first name
LastName	Variable Character	VARCHAR(X)	Yes	20	NOT NULL	Coach last name
Street	Variable Character	VARCHAR(X)	Yes	40	NOT NULL	Coach street address
ZipCode	Character	CHAR(X)	Yes	5	FK NOT NULL	Coach zipcode
PhoneNumber	Variable Character	VARCHAR(X)	Yes	20	NOT NULL	Coach mobile number
RatingID	Integer	NUMBER	Yes	N/A	FK NOT NULL	Unique identifier for RatingID instance

**Ratings**

Attribute Name	Data Type	MS Access Type	Required?	Length/Size	Constraints	Description/ Purpose
RatingID	Integer	NUMBER	Yes	N/A	PK NOT NULL	Unique identifier for RatingID instance
RatingDesc	Variable Character	VARCHAR(X)	Yes	20	NOT NULL	Rating description

**Materials**

Attribute Name	Data Type	MS Access Type	Required?	Length/Size	Constraints	Description/ Purpose
MaterialID	Integer	AUTOINCREMENT (X)	Yes	600,1	PK NOT NULL	Unique identifier for MaterialID instance
Type	Variable Character	VARCHAR(X)	Yes	30	NOT NULL	Material name
Quantity	Integer	NUMBER	Yes	N/A	NOT NULL	Material quantity

**ServicesRendered**

Attribute Name	Data Type	MS Access Type	Required?	Length/Size	Constraints	Description/ Purpose
ServiceRenderedID	Integer	AUTOINCREMENT (X)	Yes	700,1	PK NOT NULL	Unique identifier for ServiceRenderedID instance
AppointmentID	Integer	LONG	Yes	N/A	FK NOT NULL	Unique identifier for AppointmentID instance
ServiceID	Integer	LONG	Yes	N/A	FK NOT NULL	Unique identifier for ServiceID instance

CourtID	Integer	LONG	Yes	N/A	FK	Unique identifier for CourtID instance
CoachID	Integer	LONG	Yes	N/A	FK	Unique identifier for CoachID instance

**ServicesRenderedMaterials**

Attribute Name	Data Type	MS Access Type	Required?	Length/Size	Constraints	Description/Purpose
ServiceRenderedID	Integer	LONG	Yes	N/A	PK (FK) NOT NULL	Unique identifier for ServiceRenderedID instance
MaterialID	Integer	LONG	Yes	N/A	PK (FK) NOT NULL	Unique identifier for MaterialID instance
MaterialQuantity	Integer	NUMBER	Yes	N/A	NOT NULL	Material quantity to be assigned to service

---

**End of Physical Design**

---



**SQL DDL used to create the tables, primary key and foreign key constraints****CREATE TABLE STATEMENTS**

---

CREATE TABLE CustomersInfo

```
(
  MemberShipID AUTOINCREMENT(100,1) NOT NULL,
  FirstName VARCHAR(20) NOT NULL,
  LastName VARCHAR(20) NOT NULL,
  Street VARCHAR(40) NOT NULL,
  ZipCode CHAR(5) NOT NULL,
  PhoneNumber VARCHAR(20),
  EmailAddress VARCHAR(50)
);
```

CREATE TABLE ZipCodes

```
(
  ZipCode CHAR (5) NOT NULL,
  City VARCHAR(20) NOT NULL,
  State CHAR (2) NOT NULL
);
```

CREATE TABLE Appointments

```
(
  AppointmentID AUTOINCREMENT(200,1) NOT NULL,
  AppointmentDate DATE NOT NULL,
  StartTime TIME NOT NULL,
  EndTime TIME NOT NULL,
  MemberShipID LONG NOT NULL
);
```

CREATE TABLE ClubServices

```
(
  ServiceID AUTOINCREMENT(300,1) NOT NULL,
  ServiceName VARCHAR(50) NOT NULL,
  ServiceDesc VARCHAR(100) NOT NULL
);
```

CREATE TABLE TennisCourts

```
(
  CourtID AUTOINCREMENT(400,1) NOT NULL,
```

```
Surface VARCHAR(30) NOT NULL,  
Facility VARCHAR(10) NOT NULL  
);
```

```
CREATE TABLE CoachesInfo
```

```
(  
CoachID AUTOINCREMENT(500,1) NOT NULL,  
FirstName VARCHAR(20) NOT NULL,  
LastName VARCHAR(20) NOT NULL,  
Street VARCHAR(40) NOT NULL,  
ZipCode CHAR(5) NOT NULL,  
PhoneNumber VARCHAR(20) NOT NULL,  
RatingID NUMBER NOT NULL  
);
```

```
CREATE TABLE Ratings
```

```
(  
RatingID NUMBER NOT NULL,  
RatingDesc VARCHAR(20) NOT NULL  
);
```

```
CREATE TABLE Materials
```

```
(  
MaterialID AUTOINCREMENT(600,1) NOT NULL,  
Type VARCHAR(30) NOT NULL,  
Quantity NUMBER NOT NULL  
);
```

```
CREATE TABLE ServicesRendered
```

```
(  
ServiceRenderedID AUTOINCREMENT(700,1) NOT NULL,  
AppointmentID LONG NOT NULL,  
ServiceID LONG NOT NULL,  
CourtID LONG,  
CoachID LONG  
);
```

```
CREATE TABLE ServicesRenderedMaterials
```

```
(  
ServiceRenderedID LONG NOT NULL,  
MaterialID LONG NOT NULL,  
MaterialQuantity NUMBER NOT NULL  
);
```

**ALTER TABLE STATEMENTS TO ADD PRIMARY KEYS**

---

```
ALTER TABLE CustomersInfo
    ADD CONSTRAINT PK_Customer
        PRIMARY KEY (MembershipID);
```

```
ALTER TABLE ZipCodes
    ADD CONSTRAINT PK_ZipCode
        PRIMARY KEY (ZipCode);
```

```
ALTER TABLE Appointments
    ADD CONSTRAINT PK_Appointment
        PRIMARY KEY (AppointmentID);
```

```
ALTER TABLE ClubServices
    ADD CONSTRAINT PK_ClubService
        PRIMARY KEY (ServiceID);
```

```
ALTER TABLE TennisCourts
    ADD CONSTRAINT PK_TennisCourt
        PRIMARY KEY (CourtID);
```

```
ALTER TABLE CoachesInfo
    ADD CONSTRAINT PK_Coaches
        PRIMARY KEY (CoachID);
```

```
ALTER TABLE Ratings
    ADD CONSTRAINT PK_Rating
        PRIMARY KEY (RatingID);
```

```
ALTER TABLE Materials
    ADD CONSTRAINT PK_Material
        PRIMARY KEY (MaterialID);
```

```
ALTER TABLE ServicesRendered
    ADD CONSTRAINT PK_ServiceRendered
        PRIMARY KEY (ServiceRenderedID);
```

```
ALTER TABLE ServicesRenderedMaterials
    ADD CONSTRAINT PK_ServiceRenderedMaterial
        PRIMARY KEY (ServiceRenderedID, MaterialID);
```

### **ALTER TABLE STATEMENTS TO ADD FOREIGN KEYS**

---

```
ALTER TABLE CustomersInfo
    ADD CONSTRAINT FK_CustomerZipCode
        FOREIGN KEY (ZipCode) REFERENCES ZipCodes (ZipCode);
```

No Foreign Key in ZipCodes table

```
ALTER TABLE Appointments
    ADD CONSTRAINT FK_Customer
        FOREIGN KEY (MemberShipID) REFERENCES CustomersInfo
            (MemberShipID);
```

No Foreign Key in ClubServices table

No Foreign Key in TennisCourts table

```
ALTER TABLE CoachesInfo
    ADD CONSTRAINT FK_CoachZipCode
        FOREIGN KEY (ZipCode) REFERENCES ZipCodes (ZipCode);
```

```
ALTER TABLE CoachesInfo
    ADD CONSTRAINT FK_Rating
        FOREIGN KEY (RatingID) REFERENCES Ratings (RatingID);
```

No Foreign Keys in Ratings table

No Foreign Keys in Materials table

```
ALTER TABLE ServicesRendered
    ADD CONSTRAINT FK_Appointment
        FOREIGN KEY (AppointmentID) REFERENCES Appointments
            (AppointmentID);
```

```
ALTER TABLE ServicesRendered
  ADD CONSTRAINT FK_ClubService
    FOREIGN KEY (ServiceID) REFERENCES ClubServices (ServiceID);
```

```
ALTER TABLE ServicesRendered
  ADD CONSTRAINT FK_TennisCourt
    FOREIGN KEY (CourtID) REFERENCES TennisCourts (CourtID);
```

```
ALTER TABLE ServicesRendered
  ADD CONSTRAINT FK_Coach
    FOREIGN KEY (CoachID) REFERENCES CoachesInfo (CoachID);
```

```
ALTER TABLE ServicesRenderedMaterials
  ADD CONSTRAINT FK_ServiceRenderedMaterial
    FOREIGN KEY (ServiceRenderedID) REFERENCES ServicesRendered
      (ServiceRenderedID);
```

```
ALTER TABLE ServicesRenderedMaterials
  ADD CONSTRAINT FK_MaterialServiceRendered
    FOREIGN KEY (MaterialID) REFERENCES Materials (MaterialID);
```

### Forms, reports and queries

Below shows the Customer Information Form. This form can be used to look up existing customers in the database and also add new customers into the database.

 Customers Info Form

MemberShipID

110

▼

New Customer

FirstName

Elizabeth

Save

LastName

Yang

Close

Street

4215 Madison Ave

ZipCode

10301

▼

City

Staten Island

State

NY


PhoneNumber

(317) 372-6222

EmailAddress

Liz.Yang@gmail.com

Below shows the Appointment Form. This form can be used to look up existing appointments using AppointmentID, and also create new appointments based on request.

 Appointments Form

AppointmentID

215

▼

New Appointment

AppointmentDate

6/9/2021

Save

StartTime

2:00:00 PM

Delete

EndTime

3:30:00 PM

Close

MemberShipID

108

FirstName

Alex

LastName

Macdonald

Below shows the Services Rendered Form. This form is used to specify the services being rendered during an existing appointment. It is also used for the creation of new ServicesRendered.

 Services Rendered Form

▶

ServiceRenderedID

710

▼

AppointmentID

209

▼

ServiceID

301

▼

ServiceName

Training

CourtID

400

▼

CoachID

505

▼

CoachFirstName

Briana

CoachLastName


Francis

New ServicesRendered

Save

Close

Below shows the Club Services Form. This form is used to identify the services currently being offered by the Tennis Club. Additional Services can also be needed based on the Club expansion of services.

 Club Services Form

▶

ServiceID

301

▼


ServiceName

Training

ServiceDesc

Renting court and require coach for tranning

Below Shows the Tennis Courts Form. This form is used to display the Tennis Courts that the club currently possess. These courts can also vary by surface, and facility.

 Tennis Courts Form

CourtID

400

▼


Surface

Grass

Facility

Outdoor

Below shows the Coaches Information Form. This form is used to show all coaches employed by the club. It can also be used to add newly employed coaches to the database. Furthermore, each coach is then assigned a rating based on experience level.

 Coaches Info Form

CoachID

501

▼

FirstName

Steven

LastName

Cornor

Street

52 E 2th St

ZipCode

10002

▼

City

Mahattan

State

NY

PhoneNumber

(917) 342-2373

RatingID

5


New Coach

Save

Delete



Below shows the Ratings Form. This form is used to create the ratings level which will be assigned to the coaches based on their level of experience. Note: Currently there is no need for this to be updated as 5 is considered the highest rating assigned to any coach.

 Ratings Form

RatingID


5

▼

RatingDesc

Five Stars

Below shows the ZipCodes Form. This form is used to identify existing zip code in the database with their respected city and state. In addition, it is also used to add new zip code information.

 ZipCodes Form

ZipCode

10005

▼

New ZipCode

City


Mahattan

Save

State

NY

Below shows the Materials Form. This form identifies all the existing materials utilized by the club to perform its Club Services. It also included the quantity of said materials.

 Materials Form

MaterialID

600

▼

Type

Ball

Quantity

2000

Below shows the Services Rendered Materials Form. This form can be used to identify the type of material assigned to a specific ServicesRendered, and also the quantity of said assigned material.

## Services Rendered Materials Form

ServiceRenderedID

710

▼

MaterialID

600

▼

Type

Ball

MaterialQuantity

2

Below shows the Appointments Report. This report is used for business purposes, to better keep track of the appointments schedule for specific dates and their respected time.

Appointments Report					Monday, May 17, 2021 5:20:40 PM
AppointmentID	MemberShipID	AppointmentDate	StartTime	EndTime	
201	111	6/7/2021	12:30:00 PM	1:30:00 PM	
202	101	6/7/2021	4:30:00 PM	5:00:00 PM	
203	109	6/7/2021	4:00:00 PM	5:00:00 PM	
204	113	6/7/2021	2:00:00 PM	3:30:00 PM	
200	100	6/7/2021	9:00:00 AM	11:00:00 AM	
211	118	6/8/2021	4:00:00 PM	5:00:00 PM	
205	107	6/8/2021	9:30:00 AM	10:30:00 AM	
206	103	6/8/2021	1:00:00 PM	2:00:00 PM	
207	118	6/8/2021	2:00:00 PM	3:00:00 PM	
208	112	6/8/2021	2:00:00 PM	3:30:00 PM	
209	114	6/8/2021	3:00:00 PM	4:30:00 PM	
210	119	6/8/2021	4:00:00 PM	5:00:00 PM	
218	107	6/9/2021	9:00:00 AM	10:30:00 AM	
217	115	6/9/2021	4:00:00 PM	4:30:00 PM	

Below shows the Services Rendered Report. This report is also for business use and it displays what services will be performed during each appointment. It also highlights whether a court and coach are needed, if yes, their respected identifier.

Services Rendered Report					Monday, May 17, 2021 5:23:29 PM	
ServiceRenderedID	AppointmentID	ServiceID	CourtID	CoachID		
700	200	302				
701	201	300	400			
702	209	301	400	505		
703	203	301	402	502		
704	204	302				
705	205	300	401			
706	206	300	401			
707	207	301	402	502		
708	208	301	401	503		
709	209	302				
710	210	300	402			
711	211	301	401	501		
712	212	302				

Below is the Services Rendered Materials Report. This report is also an addition to the previously mentioned report used for business purposes. It shows the materials that is assigned to a specific Service Rendered and its quantity.

Services Rendered Materials Report			Monday, May 17, 2021 5:26:22 PM	
ServiceRenderedID	MaterialID	MaterialQuantity		
700	604	2		
701	601	1		
701	602	1		
701	600	200		
702	603	5		
703	603	5		
704	604	1		
707	603	5		
708	603	5		
709	604	3		
710	600	2		
710	601	2		
711	603	5		

## Navigation Form

Below shows the Navigation Form. This user interface allows for easy access to all forms available for scheduling. In addition, it also consists of reports created for business uses. Each form can be viewed by clicking on their expected names on the far left.

The screenshot displays a web application interface. At the top, a light blue header bar contains a small icon and the text "Navigation Form". Below this, a red sidebar on the left lists various menu items in white text: CustomersInfo, ZipCodes, Appointments, ServicesRendered, ClubServices, TennisCourts, CoachesInfo, Ratings, ServicesRenderedMaterials, Materials, Appointments Report, ServicesRendered Report, and ServicesRenderedMaterials Report. The main content area has a light blue header bar with a small icon and the text "Customers Info Form". Below this header, the form contains several input fields with labels and values: MemberShipID (110), FirstName (Elizabeth), LastName (Yang), Street (4215 Madison Ave), ZipCode (10301), City (Staten Island), State (NY), PhoneNumber ((317) 372-6222), and EmailAddress (Liz.Yang@gmail.com). To the right of the MemberShipID field is a "New Customer" button. To the right of the FirstName field is a "Save" button. To the right of the LastName field is a "Close" button.

Navigation Form	
CustomersInfo	Customers Info Form
ZipCodes	
Appointments	
ServicesRendered	
ClubServices	
TennisCourts	
CoachesInfo	
Ratings	
ServicesRenderedMaterials	
Materials	
Appointments Report	
ServicesRendered Report	
ServicesRenderedMaterials Report	

Customers Info Form	
MemberShipID	110
FirstName	Elizabeth
LastName	Yang
Street	4215 Madison Ave
ZipCode	10301
City	Staten Island
State	NY
PhoneNumber	(317) 372-6222
EmailAddress	Liz.Yang@gmail.com

## EXTRA CREDIT

For the purpose of the extra credit, we migrated our database to an SQL Server using Gearhost. Please see below for process:

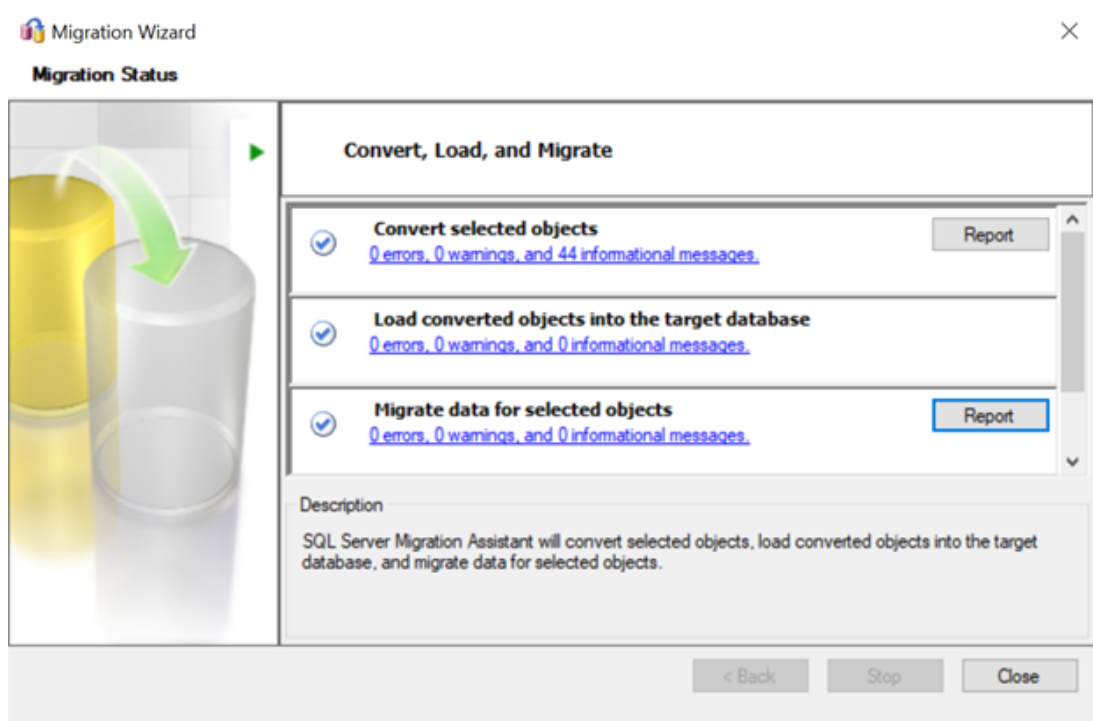
Creation of a database on Gearhost to migrate out MS Access database. Server Login credentials are also included

The screenshot shows the CloudSites dashboard. On the left is a sidebar with navigation links: CloudSites, Databases (selected), Domains, Certificates, Billing, Settings, API, Referrals, and Support. The main content area has a header with 'tennisclubdb' and an 'Add Database User' button. Below this is a 'Free Database' banner with a database icon and text: 'You are using free database which is limited. It's recommended to upgrade to Standard.' with an 'Upgrade Now' button. A table titled 'database users' is displayed, showing the following data:

Username	Password	Permission
tennisclubdb	Bj22kl7GR0f_	Read & Write

At the bottom left, it says 'Made with ❤️ in Arizona Build: 5.0.1.40151'. At the bottom right, there is a 'Show/Hide Password' toggle.

Data being Migrated onto the Gearhost Server showing section status of completion



Report showing tables that were completely migrated to the server

Data Migration Report

Status	From	To	Total Rows	Migrated Rows	Success Rate
ⓘ	Data_Generation_Tennis Club-Database: Appointments	tennisclubdb.dbo.[Appointments]	49	49	100.00%
ⓘ	Data_Generation_Tennis Club-Database: ClubServices	tennisclubdb.dbo.[ClubServices]	3	3	100.00%
ⓘ	Data_Generation_Tennis Club-Database: CoachesInfo	tennisclubdb.dbo.[CoachesInfo]	6	6	100.00%
ⓘ	Data_Generation_Tennis Club-Database: CustomersInfo	tennisclubdb.dbo.[CustomersInfo]	20	20	100.00%
ⓘ	Data_Generation_Tennis Club-Database: Materials	tennisclubdb.dbo.[Materials]	5	5	100.00%
ⓘ	Data_Generation_Tennis Club-Database: Ratings	tennisclubdb.dbo.[Ratings]	5	5	100.00%
ⓘ	Data_Generation_Tennis Club-Database: ServicesRendered	tennisclubdb.dbo.[ServicesRendered]	49	49	100.00%
ⓘ	Data_Generation_Tennis Club-Database: ServicesRenderedMaterials	tennisclubdb.dbo.[ServicesRenderedMat]	47	47	100.00%
ⓘ	Data_Generation_Tennis Club-Database: TennisCourts	tennisclubdb.dbo.[TennisCourts]	5	5	100.00%
ⓘ	Data_Generation_Tennis Club-Database: ZipCodes	tennisclubdb.dbo.[ZipCodes]	25	25	100.00%

Performing SQL querying on the database after fully migrating to the gear host server. Note: All 10 Tables was successfully migrated

SQLQuery1.sql - den1.mssql7.gear.host.tennisclubdb (tennisclubdb (189)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

tennisclubdb

Object Explorer

- Connect
- den1.mssql7.gear.host (SQL Server 14)
  - Databases
    - System Databases
    - Database Snapshots
    - tennisclubdb
      - Database Diagrams
      - Tables
        - System Tables
        - FileTables
        - External Tables
        - Graph Tables
        - dbo.Appointments
        - dbo.ClubServices
        - dbo.CoachesInfo
        - dbo.CustomersInfo
        - dbo.Materials
        - dbo.Ratings
        - dbo.ServicesRendered
        - dbo.ServicesRenderedMaterials
        - dbo.TennisCourts
        - dbo.ZipCodes
      - Views
      - External Resources
      - Synonyms
      - Programmability

SQLQuery1.sql - de...tennisclubdb (189)

```

USE [tennisclubdb]
GO

SELECT [MembershipID]
,[FirstName]
,[LastName]
,[Street]
,[ZipCode]
,[PhoneNumber]
,[EmailAddress]
FROM [dbo].[CustomersInfo]
GO
  
```

100 %

Results

	MembershipID	FirstName	LastName	Street	ZipCode	PhoneNumber	EmailAddress
1	100	Duyen	Nguyen	45 Fort Washington Ave	10451	(543) 654-3454	duyen.nguyen@gmail.com
2	101	Hana	Lexis	21 Crown St	10452	(543) 333-7787	hanabala@gmail.com
3	102	Dal	Miller	1225 S Grove St	10453	(646) 112-6444	dal.miller@gmail.com
4	103	Milo	Bloomer	25 Fort Hamlington Ave	11224	(912) 622-8431	m.bloomer12345@gmail.com
5	104	Fisher	Miami	55 King St	10302	(565) 122-4544	fisher.miami@gmail.com
6	105	John	Cena	62nd E 53rd St	11104	(236) 843-1267	John.Cenaaa@gmail.com
7	106	Strong	Lily	666 N Mississippi	10305	(72) 455-5433	lily_1005@gmail.com
8	107	Blake	James	232 Fort St	11225	(345) 222-1342	james_hero@gmail.com
9	108	Alex	Macdonald	925 Centre St	10453	(361) 126-7212	Alex.Mac@gmail.com
10	109	Blus	Ilan	123 Beer Ave	10001	(543) 345-5444	fly_blue@gmail.com
11	110	Elzabeth	Yang	4215 Madison Ave	10301	(317) 372-6222	Liz.Yang@gmail.com
12	111	Hoa	Ho	123 Row St	11222	(456) 234-6787	hoa.hoa123@hotmail.com

Query executed successfully.

den1.mssql7.gear.host (14.0... tennisclubdb (189)) : tennisclubdb : 00:00:00 : 20 rows

Ready Ln 14 Col 1 Ch 1 INS

MS Access after migration to SQL server was completed. Note: Credentials needed to access the database are highlighted in above “Gearhost Server” Snip.

**All Access Objects**

Search...

Tables

- SSMA\$Appointments\$local
- SSMA\$ClubServices\$local
- SSMA\$CoachesInfo\$local
- SSMA\$CustomersInfo\$local
- SSMA\$Materials\$local
- SSMA\$Ratings\$local
- SSMA\$ServicesRendered\$local
- SSMA\$ServicesRenderedMaterials\$local
- SSMA\$TennisCourts\$local
- SSMA\$ZipCodes\$local
- Appointments
- ClubServices
- CoachesInfo
- CustomersInfo
- Materials
- Ratings
- ServicesRendered
- ServicesRenderedMaterials

**CustomersInfo**

MemberShipID	FirstName	LastName	Street	ZipCode	PhoneNumber	EmailAddress
100	Duyen	Nguyen	45 Fort Washington Ave	10451	(543) 654-3454	duyen.nguyen@gmail.com
101	Hana	Lexis	21 Crown St	10452	(543) 333-7787	hanalala@gmail.com
102	Dat	Miller	1225 S Grove St	10453	(646) 112-6444	dat.miller@gmail.com
103	Milo	Bloomer	25 Fort Hamillington Ave	11224	(912) 622-8431	m.bloomer12345@gmail.com
104	Fisher	Miami	55 King St	10302	(565) 122-4544	fisher.miami@gmail.com
105	John	Cena	62nd E 53rd St	11104	(236) 843-1267	John.Cenaaa@gmail.com
106	Strong	Lily	666 N Mississippi	10305	(72) 455-5433	lily_1005@gmail.com
107	Blake	James	232 Fort St	11225	(345) 222-1342	james_hero@gmail.com
108	Alex	Macdonald	925 Centre St	10453	(361) 126-7212	Alex.Mac@gmail.com
109	Blue	Ilan	123 Beer Ave	10001	(543) 345-5444	fly_blue@gmail.com
110	Elizabeth	Yang	4215 Madison Ave	10301	(317) 372-6222	Liz.Yang@gmail.com
111	Hoa	He	123 Bay St	11222	(455) 234-6767	hoa.hoa123@gmail.com
112	May	Bach	412 58th St	11105	(162) 236-1834	lloveMercedes@gmail.com
113	Blie	Nguyen	4545 Flowers St	11223	(433) 541-5432	nguyenbliee@gmail.com
114	Amy	Lau	235 President St	10003	(362) 572-3537	amyspretty@gmail.com
115	Shutting	Bleach	5656 Cat St	11101	(555) 343-0987	lilyflow_bleach@gmail.com
116	Tyler	Ngo	24 W 135 St	10303	(152) 573-3223	followtylerngo@gmail.com
117	Hanna	Baby	765 Lion Ave	11104	(1222) 875-5656	hannah_1996@gmail.com
118	James	Tran	555 Lexington Ave	10455	(344) 655-0988	tranjamesmafia@gmail.com
119	Cindy	Yu	3362 78th St	11102	(125) 667-4477	cylater@gmail.com
(New)						

Record: 14 of 21 of 21

No Filter Search

Datasheet View

## Conclusion

Data Generation's strength was communication. In the first meeting, we already decided on a specific task for each team member, which made each milestone easy to execute and submit. Before each milestone deadline, the team always met up and reviewed all the works to make sure that everyone is on track and understand the project thoroughly.

One of the most complicated tasks was to create a functional ER diagram related to our business case. We had entities and attributes that we realized were irrelevant to our business case, and it was difficult to manage the adjustments as we needed to create new tables that were conflicting with some others. However, the task was too difficult to do without affecting the other tables. Therefore, we ended up creating a new ER diagram. Another daunting task was the data implementation part. The dependencies between tables made every modification complicated as we needed to go back and forth to make it work.

The easiest part for our team was making relations and normalization as it was clear from the beginning which attributes would be primary keys and foreign keys. We enjoyed the trouble shooting aspect of the project as everything didn't go according to plan. This allowed us to research and obtain more knowledge about the database process and where to find resources.

We did not imagine being able to produce a functional database. The most stressful part was to implement data and to hope that everything works. It was fascinating to see how well our project has come out after the implementation and creation of forms.

If we had to do it all over again, we would have chosen to have it completed in PostgreSQL, as it is more updated than MS Access. In addition, it allows for better creativity required in the new age of technology.

Our project has helped to solve Cary Leeds Tennis Center's scheduling issues and to deliver all the solutions that were proposed in the beginning. For example, the managers will be able to control and organize employees' daily tasks. Furthermore, by using our database system, they can easily keep track of their scheduling appointments and the revenue flow. The new system makes sure to record every important information, such as customers info, appointment date, and time, in the right order to avoid data redundancy.

We believe that the project is a game-changer in which we can replace a manual method of tracking customers' and employees' information with an advanced database. Therefore, the operation will run smoothly and professionally. It would be out of date if we continued to use the old method. We will try to seek new methods to help the operation better and keep up with trends. In the future, we plan to go deeper and keep the database updated by combining SQL and Python for an improved data workflow.