CPSC 304 Project Cover Page

Milestone #: 2

Date: 21st of October, 2022

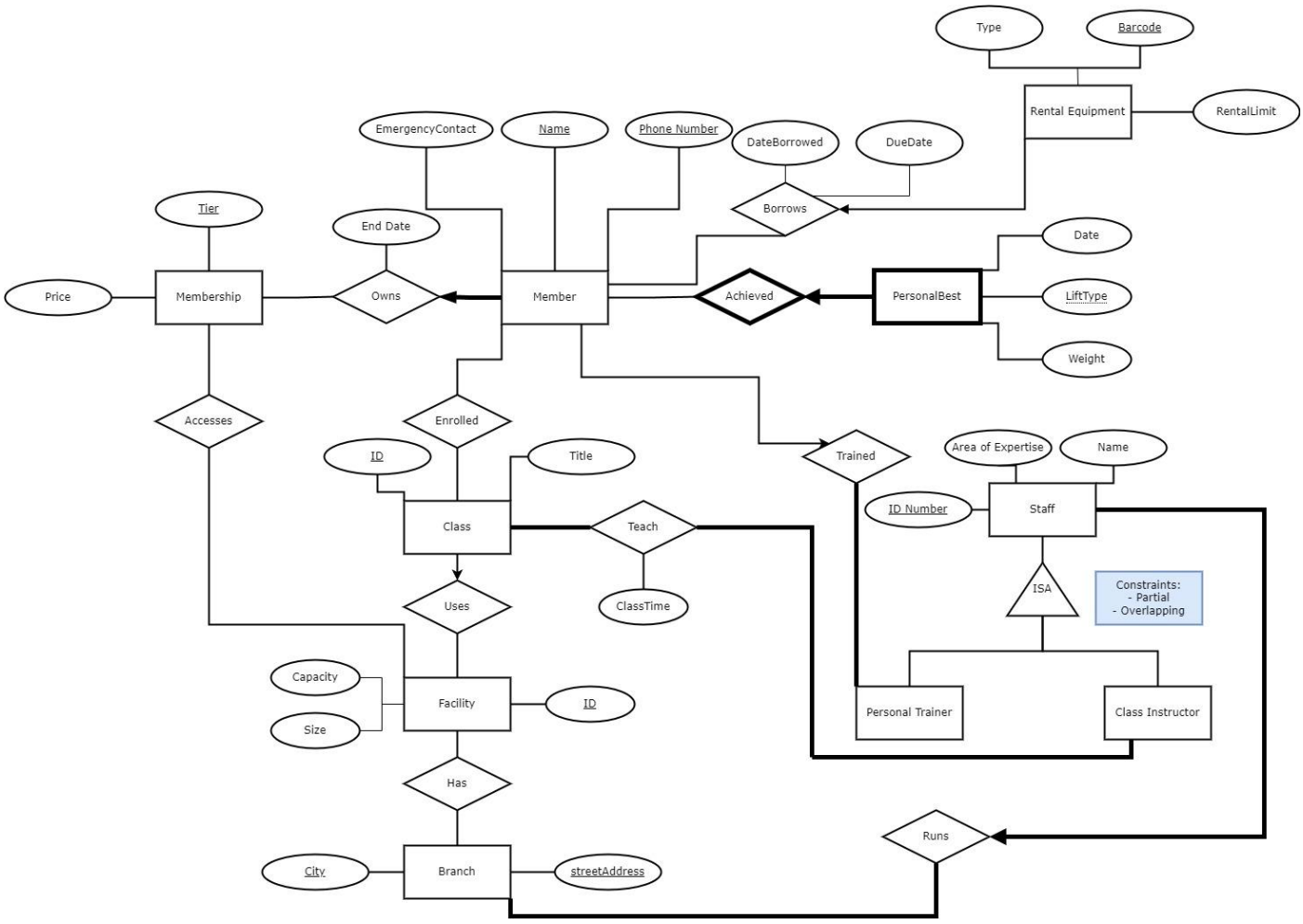
Group Number: 46

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Allison Luna	17158247	e0u4g	allisonmluna@gmail.com
Mariam Ayubi	75748251	s7w2b	mariamayubi7@gmail.com
Akim Ruslanov	52530805	y5t2b	akim.ruslanov@gmail.com

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

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

ER Diagram for Task 3



Changes Made Since Milestone 1:

1. Added *Capacity* and *Size* attributes to *Facility* to add in functional dependencies where the left hand side is not a key and initially have relations not in BCNF or 3NF to decompose.
    a. Resulting functional dependencies added:
        i. *Size → Capacity*
2. Added *DueDate, RentalLimit*, and *DateBorrowed* to *RentalEquipment* to add in functional dependencies where the left hand side is not a key and initially have relations not in BCNF or 3NF to decompose.
    a. RentalLimit represents the maximum amount of time a piece of RentalEquipment can be rented for. It is used to determine DueDate from DateBorrowed.
    b. Resulting new functional dependency:
        i. RentalLimit, DateBorrowed → DueDate
3. Renamed *Has*, the relationship between Facility and Branch, to *LocatedIn*, so the name provides more information.

Task 3: Deriving Schema from our ER Diagram

StaffRuns(AreaOfExpertise: CHAR(30), Name: CHAR(40), IDNumber: INTEGER, **BranchCity**: CHAR(40), **BranchStreetAddress**: CHAR(60))

- Candidate Keys: IDNumber

Personal Trainer(**IDNumber**: INTEGER)

- Candidate Keys: IDNumber

Class Instructor(**IDNumber**: INTEGER)

- Candidate Keys: IDNumber

LocatedIn(**FacilityID**: CHAR(10), **BranchCity**: CHAR(40), **BranchStreetAddress**: CHAR(60))

- Candidate Keys: FacilityID + BranchCity + BranchStreetAddress

Branch(City: CHAR(40), StreetAddress: CHAR(60))

- Candidate Keys: City + StreetAddress

Member(Name: CHAR(40), PhoneNumber: CHAR(10), Emergency Contact: CHAR(10), **MembershipTier**: CHAR(15), **PersonalTrainerId**: INTEGER, EndDate: DATE)

- Candidate Key: Name, Phone Number
- MembershipTier must be not null to include the constraint

Membership(Tier: CHAR(15), Price: CHAR(10))

- Candidate Key: Tier

PersonalBest(**MemberName**: CHAR(40), **MemberPhone**: CHAR(10), Date: DATE, LiftType: CHAR(15), Weight: INTEGER)

- Candidate Key: Member Name, Phone Number, Lift Type
- MembershipName and PhoneNumber must be not null to satis. the constraint

Accesses(**MembershipTier**: CHAR(15), **FacilityID**: CHAR(10))

- CandidateKey: MembershipTier, FacilityID

Class(ID: INTEGER, Title: CHAR(40), **FacilityID**: CHAR(10))

- Candidate Key: ID

Enrolled(**MemberName**: CHAR(40), **MemberPhone**: CHAR(10), **CourseID**: INTEGER)

- Candidate Key: Member Name, Member Phone, Course ID

BorrowsRentalEquipment(Barcode: INTEGER, Type: CHAR(15), **MemberName**: CHAR(40), **MemberPhone**: CHAR(10)**,** RentalLimit: INTEGER, DateBorrowed: DATE, DueDate: DATE)

- Candidate Key: Barcode

Facility(IDNumber: INTEGER, Capacity: INTEGER, Size: CHAR(10)

- Candidate Key: ID Number

Teach(**ClassID**: INTEGER, **ClassInstructorID**: INTEGER, ClassTime: CHAR(11))

- Candidate Key: ClassID, ClassInstructorID

Task 4: Functional Dependencies

StaffRuns:
- IDNumber -> Name
- IDNumber -> AreaOfExpertise
- IDNumber -> BranchCity
- IDNumber -> BranchStreetAddress

PersonalTrainer:
- IDNumber -> IDNumber

ClassInstructor:
- IDNumber -> IDNumber

LocatedIn:
- FacilityID, BranchCity, BranchStreetAddress -> FacilityID, BranchCity, BranchStreetAddress

Branch:
- City, StreetAddress -> City, StreetAddress

Member:
- Name, PhoneNumber -> Emergency Contact
- Name, PhoneNumber -> MembershipTier
- Name, PhoneNumber -> PersonalTrainerID
- Name, PhoneNumber -> EndDate

Membership:
- Tier -> Price

PersonalBest:
- MemberName, MemberPhone, LiftType -> Date
- MemberName, MemberPhone, LiftType -> Weight

Accesses:
- MembershipTier, FacilityID-> MembershipTier
- MembershipTier, FacilityID-> FacilityID

Class
- ID -> Title
- ID -> FacilityID

Enrolled:
- MemberName, MemberPhone, ClassID -> MemberName, MemberPhone, ClassID

BorrowsRentalEquipment:
- Barcode -> MemberName, MemberPhone
- Barcode -> Type
- Barcode -> RentalLimit
- Barcode -> DueDate
- Barcode -> DateBorrowed
- DateBorrowed, RentalLimit -> DueDate

Facility:
- IDNumber -> IDNumber
- IDNumber -> Size
- IDNumber -> Capacity

- Size -> Capacity

Teach:

- ClassID, ClassInstructorID -> ClassTime

Task 5: Normalization

Normalization 1:

Facility(IDNumber, Size Capacity)

Minimal key is IDNumber

FDs:

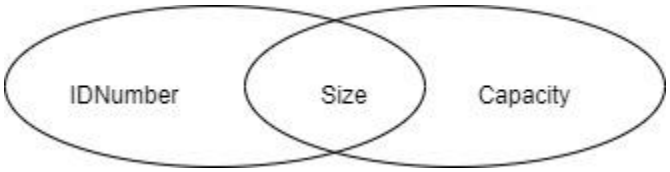
1. IDNumber → Size
2. IDNumber → Capacity
3. Size → Capacity

Closures:

1. $\{IDNumber\}^+ = \{IDNumber, Size, Capacity\}$
2. $\{Size\}^+ = \{Size, Capacity\}$

Size → Capacity violates BCNF since Size is not a superkey (and also violates 3NF as Capacity is not part of a minimal key)

Split by Decomposition



We get:

Facility(IDNumber, **Size**)

FacilitySize(Size, Capacity)

Normalization 2:

BorrowsRentalEquipment(Barcode, Type, **MemberName**, **MemberPhone,** RentalLimit, DateBorrowed, DueDate)

Minimal Key: Barcode

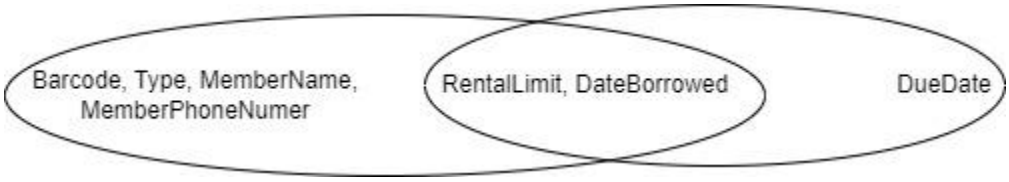
FDs:

1. Barcode → Type
2. Barcode → MemberName
3. Barcode → MemberPhone
4. Barcode → RentalLimit
5. Barcode → DateBorrowed
6. Barcode → DueDate
7. RentalLimit, DateBorrowed → DueDate

Closures:

1. $\{Barcode\}^+ = \{Barcode, Type, MemberName, MemberPhone, RentalLimit, DateBorrowed, DueDate\}$
2. $\{RentalLimit, DateBorrowed\}^+ = \{RentalLimit, DateBorrowed, DueDate\}$

FD 7 violates BCNF since RentalLimit and DateBorrowed are not a superkey and also violate 3NF as DueDate is not part of a minimal key.
Split by decomposition



We get:
BorrowsRentalEquipment(Barcode, Type, **MemberName**, **MemberPhone**, **RentalLimit, DateBorrowed**)
RentalEquipmentDue(RentalLimit, DateBorrowed, DueDate)

Now they are all in BCNF.

Final List of tables:

StaffRuns(AreaOfExpertise: CHAR(30), Name: CHAR(40), IDNumber: INTEGER, **BranchCity**: CHAR(40), **BranchStreetAddress**: CHAR(60))
- Candidate Keys: IDNumber

Personal Trainer(**IDNumber**: INTEGER)
- Candidate Keys: IDNumber

Class Instructor(**IDNumber**: INTEGER)
- Candidate Keys: IDNumber

LocatedIn(**FacilityID**: CHAR(10), **BranchCity**: CHAR(40), **BranchStreetAddress**: CHAR(60))
- Candidate Keys: FacilityID + BranchCity + BranchStreetAddress

Branch(City: CHAR(40), StreetAddress: CHAR(60))
- Candidate Keys: City + StreetAddress

Member(Name: CHAR(40), PhoneNumber: CHAR(10), Emergency Contact: CHAR(10), **MembershipTier**: CHAR(15), **PersonalTrainerId**: INTEGER, EndDate: DATE)
- Candidate Key: Name, Phone Number
- MembershipTier must be not null to include the constraint

Membership(Tier: CHAR(15), Price: CHAR(10))
- Candidate Key: Tier

PersonalBest(**MemberName**: CHAR(40), **MemberPhone**: CHAR(10), Date: DATE, LiftType: CHAR(15), Weight: INTEGER)
- Candidate Key: Member Name, Phone Number, Lift Type
- MembershipName and PhoneNumber must be not null to satis. the constraint

Accesses(**MembershipTier**: CHAR(15), **FacilityID**: CHAR(10))
- CandidateKey: MembershipTier, FacilityID

Class(ID: INTEGER, Title: CHAR(40), **FacilityID**: CHAR(10))
- Candidate Key: ID

Enrolled(**MemberName**: CHAR(40), **MemberPhone**: CHAR(10), **CourseID**: INTEGER)
- Candidate Key: Member Name, Member Phone, Course ID

BorrowsRentalEquipment(Barcode: INTEGER, Type: CHAR(15), **MemberName**: CHAR(40), **MemberPhone**: CHAR(10), **RentalLimit**: INTEGER, **DateBorrowed**: DATE)

- Candidate Key: Barcode, Member Name, Member Phone, RentalLimit, DateBorrowed

RentalEquipmentDue(RentalLimit: INTEGER, DateBorrowed: DATE, DueDate: DATE)

- Candidate Key: RentalLimit, DateBorrowed

Facility(IDNumber: INTEGER, **Size**: CHAR(10))

- Candidate Key: ID Number

FacilitySize(Size: CHAR(10), Capacity: INTEGER)

- Candidate Key: Size

Teach(**ClassID**: INTEGER, **ClassInstructorID**: INTEGER, ClassTime: CHAR(11))

- Candidate Key: ClassID, ClassInstructorID

Task 6: Table Creation

```
CREATE TABLE StaffRuns(
        AreaOfExpertise CHAR(30),
        Name CHAR(40) NOT NULL,
        IDNumber INTEGER PRIMARY KEY,
        BranchCity CHAR(40) NOT NULL,
        BranchStreetAddress CHAR(60) NOT NULL,
        FOREIGN KEY(BranchCity, BranchStreetAdress),
                REFERENCES Branch(City, StreetAddress)
                ON DELETE CASCADE
                ON UPDATE CASCADE

);

CREATE TABLE PersonalTrainer(
        IDNumber INTEGER PRIMARY KEY,
        FOREIGN KEY(IDNumber),
                REFERENCES StaffRuns(IDNumber)
                ON DELETE CASCADE
                ON UPDATE CASCADE
);

CREATE TABLE ClassInstructor(
        IDNumber INTEGER PRIMARY KEY,
        FOREIGN KEY(IDNumber),
                REFERENCES StaffRuns(IDNumber)
                ON DELETE CASCADE
                ON UPDATE CASCADE
);

CREATE TABLE LocatedIn(
        FacilityID CHAR(10),
        BranchCity CHAR(40),
        BranchStreetAddress(60),
        PRIMARY KEY(FacilityID, BranchCity, BranchStreetAddress),
        FOREIGN KEY(FacilityID) REFERENCES Facility(IDNumber)
                ON DELETE CASCADE
                ON UPDATE CASCADE,
        FOREIGN KEY(BranchCity, BranchStreetAddress) REFERENCES Branch(City,
        StreetAddress)
                ON DELETE CASCADE
                ON UPDATE CASCADE
);
```

```
CREATE TABLE Branch(
        City CHAR(40),
        StreetAddress CHAR(60),
        PRIMARY KEY (City, StreetAddress)
);


CREATE TABLE Enrolled(
        MemberName CHAR(40),
        MemberPhone CHAR(10),
        ClassID INTEGER,
        PRIMARY KEY (MemberName, MemberPhone, ClassID),
        FOREIGN KEY(MemberName, MemberPhone) REFERENCES Member(Name,
PhoneNumber)
                ON DELETE CASCADE
                ON UPDATE CASCADE,
        FOREIGN KEY(ClassID) REFERENCES Class(ID)
                ON DELETE CASCADE
                ON UPDATE CASCADE
);
CREATE TABLE BorrowsRentalEquipment(
        Barcode INTEGER PRIMARY KEY,
        Type CHAR(15) NOT NULL,
        MemberName CHAR(40),
        MemberPhone CHAR(10),
        RentalLimit INTEGER,
        DateBorrowed DATE,
        FOREIGN KEY(MemberName, MemberPhone) REFERENCES Member(Name,
PhoneNumber)
                ON DELETE CASCADE
                ON UPDATE CASCADE,
        FOREIGN KEY (RentalLimit, DateBorrowed) REFERENCES
        RentalEquipmentDue(RentalLimit, DateBorrowed)
                ON DELETE SET NULL
                ON UPDATE CASCADE
);

CREATE TABLE RentalEquipmentDue(
        RentalLimit INTEGER,
        DateBorrowed DATE,
        DueDate DATE NOT NULL,
        PRIMARY KEY (RentalLimit, DateBorrowed)
);
```

```
CREATE TABLE Facility(
	ID CHAR(10) PRIMARY KEY,
	Size CHAR(10),
	FOREIGN KEY (Size) REFERENCES FacilitySize(Size)
		ON DELETE SET DEFAULT
		ON UPDATE CASCADE
);

CREATE TABLE FacilitySize(
	Size CHAR(10) PRIMARY KEY,
	Capacity INTEGER NOT NULL
);

CREATE TABLE Teach(
	ClassID INTEGER,
	ClassInstructorID INTEGER,
	ClassTime CHAR(11),
	PRIMARY KEY (ClassID, ClassInstructorID),
	FOREIGN KEY (ClassID) REFERENCES Class(ID)
		ON DELETE CASCADE
		ON UPDATE CASCADE,
	FOREIGN KEY (ClassInstructorID) REFERENCES ClassInstructor(IDNumber)
		ON DELETE CASCADE
		ON UPDATE CASCADE
);

CREATE TABLE Member(
	Name CHAR(40),
	PhoneNumber CHAR(10),
	EmergencyContact CHAR(10) NOT NULL,
	EndDate DATE NOT NULL,
	MembershipTier CHAR(15) NOT NULL,
	PersonalTrainerID INTEGER,
	PRIMARY KEY(Name, PhoneNumber),
	FOREIGN KEY (MembershipTier) REFERENCES Membership(Tier)
		ON DELETE CASCADE
		ON UPDATE CASCADE,
	FOREIGN KEY (PersonalTrainerID) REFERENCES PersonalTrainer(IDNumber)
		ON DELETE SET NULL
		ON UPDATE CASCADE
);

CREATE TABLE Membership(
	Tier CHAR(15),
```

```
        Price CHAR(10) NOT NULL,
        PRIMARY KEY (Tier)
);

CREATE TABLE PersonalBest(
        MemberName CHAR(40),
        MemberPhoneCHAR(10),
        Date DATE,
        LiftType CHAR(15),
        Weight INTEGER NOT NULL,
        PRIMARY KEY (MemberName, MemberPhone, LiftType),
        FOREIGN KEY (MemberName, MemberPhone) REFERENCES  Member(Name,
PhoneNumber)
                ON DELETE CASCADE
                ON UPDATE CASCADE
);

CREATE TABLE Accesses(
        MembershipTier CHAR(15),
        FacilityID CHAR(10),
        PRIMARY KEY (MembershipTier, FacilityID),
        FOREIGN KEY (MembershipTier) REFERENCES Membership(Tier) ON DELETE
CASCADE ON UPDATE CASCADE,
        FOREIGN KEY (FacilityID) REFERENCES Facility (ID)
                ON DELETE CASCADE
                ON UPDATE CASCADE
);


CREATE TABLE Class (
        ID INTEGER PRIMARY KEY,
        Title CHAR(40) NOT NULL,
        FacilityID CHAR(10),
        FOREIGN KEY (FacilityID) REFERENCES Facility (ID)
                ON DELETE CASCADE
                ON UPDATE CASCADE
);
```

Task 7: Table Population

```
INSERT INTO StaffRuns                VALUES(“Yoga”, “Andy Smythe”, “193756”, “Burnaby”,
“1673   Macdonald St.”);
INSERT INTO StaffRuns                VALUES(“Kickboxing”, “Sam Proud”, “739038”,
“Vancouver”, “13 Main St.”);
```

```
INSERT INTO StaffRuns VALUES(“Reception”, “Sasha Williams”,”178033”, “Coquitlam”, “4474 Maple Ave.”);
INSERT INTO StaffRuns VALUES(“Kickboxing”, “Michelle Quaker”,”777977”, “Vancouver”, “13 Main St.”);
INSERT INTO StaffRuns VALUES(“Olympic Weightlifting”, “Charlie Poth”, “872536”, “Vancouver”, “994 Marine Dr.”);
INSERT INTO StaffRuns VALUES(“Bodybuilding”, “Arnold Schwartz”, “125367”, “Toronto”, “924 Joust St.”);
INSERT INTO StaffRuns VALUES(“Pilates”, “Jordan Polo”, “045732”, “Toronto”, “924 Joust St.”);
INSERT INTO StaffRuns VALUES(“Bodybuilding”, “Kiana Keith”, “019253”, “Vancouver”, “994 Marine Dr.”);
INSERT INTO StaffRuns VALUES(“Water Aerobics”, “Vanna Blanche”, “555678”, “Coquitlam”, “4474 Maple Ave.”);
INSERT INTO StaffRuns VALUES(“Yoga”, “Frank Rivers”,“006370”, “Vancouver”, “13 Main St.”);

INSERT INTO Personal Trainer VALUES(“872536”);
INSERT INTO Personal Trainer VALUES(“739038”);
INSERT INTO Personal Trainer VALUES(“125367”);
INSERT INTO Personal Trainer VALUES(“045732”);
INSERT INTO Personal Trainer VALUES(“006370”);

INSERT INTO Class Instructor VALUES(“193756”);
INSERT INTO Class Instructor VALUES(“739038”);
INSERT INTO Class Instructor VALUES(“777977”);
INSERT INTO Class Instructor VALUES(“019253”);
INSERT INTO Class Instructor VALUES(“555678”);

INSERT INTO LocatedIn VALUES(“YOGSTD”, “Burnaby”, “1673 Macdonald St.”);
INSERT INTO LocatedIn VALUES(“POOL”, “Coquitlam”, “4474 Maple Ave.”);
INSERT INTO LocatedIn VALUES(“BOXRNG”, “Vancouver”, “13 Main St.”);
INSERT INTO LocatedIn VALUES(“GYM”, “Vancouver”, “994 Marine Dr.”);
INSERT INTO LocatedIn VALUES(“BOXRNG”, “Toronto”, “924 Joust St.”);

INSERT INTO Branch VALUES(“Burnaby”, “1673 Macdonald St.”);
INSERT INTO Branch VALUES(“Coquitlam”, “4474 Maple Ave.”);
INSERT INTO Branch VALUES(“Vancouver”, “13 Main St.”);
INSERT INTO Branch VALUES(“Vancouver”, “994 Marine Dr.”);
INSERT INTO Branch VALUES(“Toronto”, “924 Joust St.”);
```

```
INSERT INTO Member VALUES(“Terminator”, “+1234555”, “+9000955”, “Gold”, “872536”, “2022-12-12”);
INSERT INTO Member VALUES(“Bucky”, “+12355”, “+000000”, “Bronze”, “739038”, “2022-12-12);
INSERT INTO Member VALUES(“Captain Canada”, “+2352”, “+5551555”, “Silver”, “125367”, “2022-12-12);
INSERT INTO Member VALUES(“Iron Dog”, “+54252424”, “+655555”, “Platinum”, “045732”, “2022-12-12);
INSERT INTO Member VALUES(“Guy Fiery”, “+5152424”, “+6969696”, “Basic”, “006370”, “2022-12-12);

INSERT INTO Membership VALUES(“Gold”, “129.99”);
INSERT INTO Membership VALUES(“Bronze”, “59.99”);
INSERT INTO Membership VALUES(“Silver”, “79.99”);
INSERT INTO Membership VALUES(“Platinum”, “200”);
INSERT INTO Membership VALUES(“Basic”, “39.99”);

INSERT INTO PersonalBest VALUES(“Terminator”, “+1234555”, “2022-12-12”, “Bench Press”, 405);
INSERT INTO PersonalBest VALUES(“Bucky”, “+12355”, “2022-02-12”, “Squat”, 405);
INSERT INTO PersonalBest VALUES(“Captain Canada”, “+2352”, “2021-12-12”, “Dead Lift”, 455);
INSERT INTO PersonalBest VALUES(“Iron Dog”, “+54252424”, “2020-12-12”, “Bench Press”, 420);
INSERT INTO PersonalBest VALUES(“Guy Fiery”, “+5152424”, “2022-12-10”, “Front Squat”, 205);

INSERT INTO Accesses VALUES(“Gold”, “YOGSTD”);
INSERT INTO Accesses VALUES(“Silver”, “POOL”);
INSERT INTO Accesses VALUES(“Platinum”, “BOXRNG”);
INSERT INTO Accesses VALUES(“Bronze”,“GYM”);
INSERT INTO Accesses VALUES(“Basic”, “GYM”);

INSERT INTO Class VALUES(1, “Yoga for Seniors”, “YOGSTD”);
INSERT INTO Class VALUES(2, “Kickboxing for Beginners”, “BOXRNG”);
INSERT INTO Class VALUES(3, “Mua Thai”, “BOXRNG”);
INSERT INTO Class VALUES(4, “Body Building II”, “GYM”);
INSERT INTO Class VALUES(5, “Water Aerobics I”, “POOL”);

INSERT INTO Enrolled VALUES(“Terminator”, “+1234555”, 1);
INSERT INTO Enrolled VALUES(“Terminator”, “+1234555”, 5);
INSERT INTO Enrolled VALUES(“Bucky”, “+12355”, 2);
INSERT INTO Enrolled VALUES(“Captain Canada”, “+2352”, 4);
INSERT INTO Enrolled VALUES(“Iron Dog”, “+54252424”, 5);
```

```
INSERT INTO BorrowsRentalEquipment VALUES(11111, “30lbs dumbbells”, “Iron Dog”, “+54252424”, 5, “2022-09-12”);
INSERT INTO BorrowsRentalEquipment VALUES(11112, “jump rope”, “Terminator”, “+1234555”, 14, “2022-10-15”);
INSERT INTO BorrowsRentalEquipment VALUES(11113, “boxing gloves”, “Bucky”, “+12355”, 5, “2022-10-15”);
INSERT INTO BorrowsRentalEquipment VALUES(11114, “yoga mat”, “Captain Canada”, “+2352”, 7, “2022-10-15”);
INSERT INTO BorrowsRentalEquipment VALUES(11115, “boxing gloves”, “Guy Fiery”, “+5152424”, 5, “2022-09-03”);

INSERT INTO RentalEquipmentDue VALUES(5, “2022-09-12”, “2022-09-17”);
INSERT INTO RentalEquipmentDue VALUES(14, “2022-10-15”, “2022-10-29”);
INSERT INTO RentalEquipmentDue VALUES(5, “2022-10-15”, “2022-10-20”);
INSERT INTO RentalEquipmentDue VALUES(7, “2022-09-03”, “2022-09-10”);
INSERT INTO RentalEquipmentDue VALUES(5, “2022-09-13”, “2022-09-18”);

INSERT INTO Facility VALUES(“POOL”, “mid-large”);
INSERT INTO Facility VALUES(“YOGSTD”, “small”);
INSERT INTO Facility VALUES(“BOXRNG”, “medium”);
INSERT INTO Facility VALUES(“GYM”, “large”);
INSERT INTO Facility VALUES(“RUNTRK”, “small-mid”);

INSERT INTO FacilitySize VALUES(“small”, 20);
INSERT INTO FacilitySize VALUES(“medium”, 100);
INSERT INTO FacilitySize VALUES(“large”, 300);
INSERT INTO FacilitySize VALUES(“mid-large”, 200);
INSERT INTO FacilitySize VALUES(“small-mid”, 50);

INSERT INTO TEACH VALUES(1, “193756”, “2:30-3:30”);
INSERT INTO TEACH VALUES(2, “739038”, “11:00-12:30”);
INSERT INTO TEACH VALUES(2, “777977”, “11:00-12:30”);
INSERT INTO TEACH VALUES(3, “777977”, “9:00-10:00”);
INSERT INTO TEACH VALUES(4, “019253”, “9:00-10:00”);
INSERT INTO TEACH VALUES(5, “555678”, “14:00-16:00”);
```