

CPSC 304 Project Cover Page

Milestone #: 2

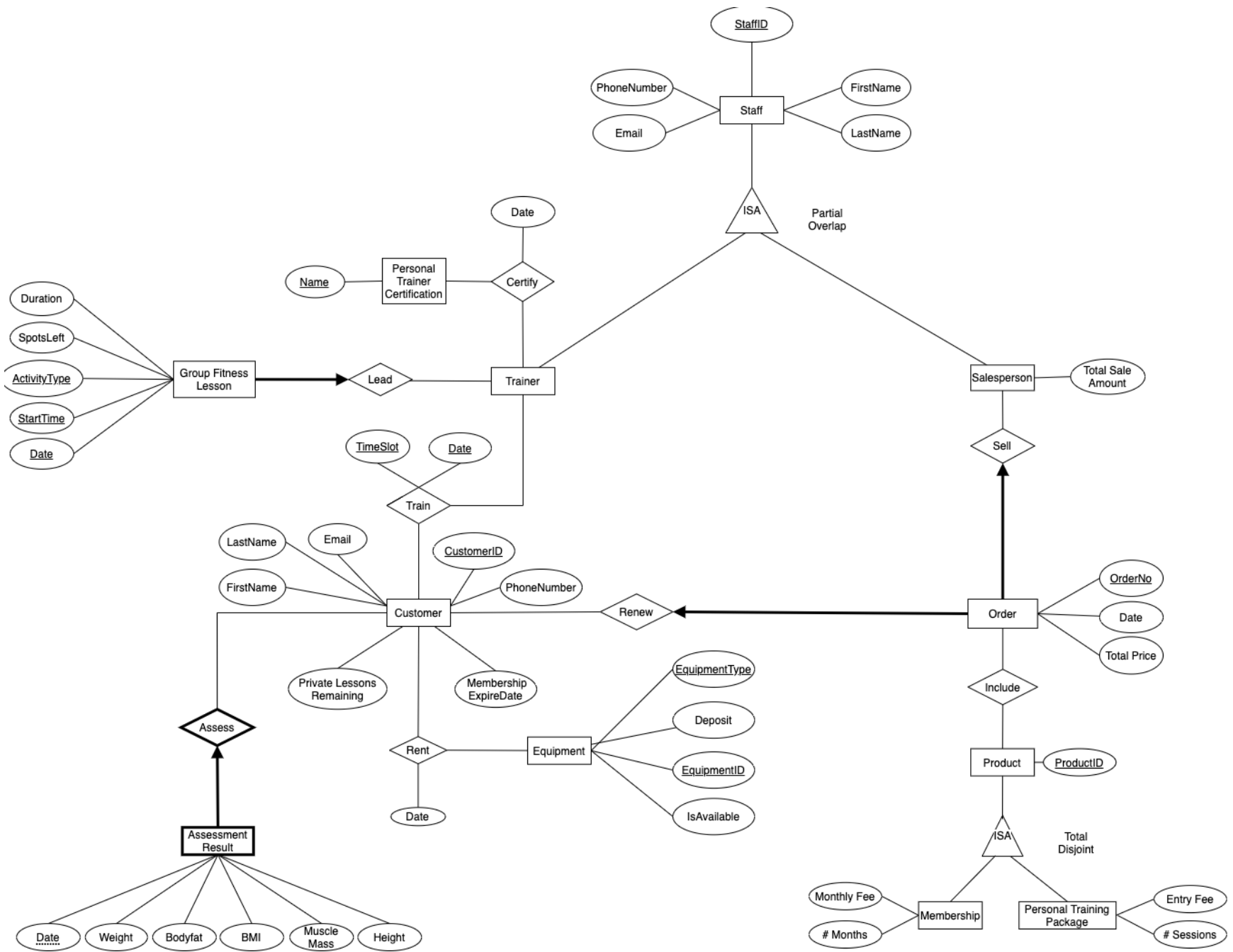
Date: Oct 25th

Group Number: 8

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Kaiqian Yang	56765233	q0d7e	yangkq5596@163.com
Xinye Zhan	67727339	v0y2b	m13819101375@163.com
Chloe Zhang	55689814	g5z2b	chloezhang1030@gmail.com

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

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



2. ER diagram

Changes we made:

- We got rid of the incorrect weak entity “emergency contact” from M1, and added another weak entity “Assessment Result”. The “Assessment Result” belongs to a “Customer”.
- We refactored the way we represent the process of a salesperson helping a customer making an order to renew his/her membership and/or personal package by adding the entity “Order”.
- We changed the way we identify unique equipment. Now a piece of equipment is identified by its type and its ID. For example, “dumbell 1”, “yogamat 1” etc.
- We specified the constraints for the “Staff” ISA and removed a children entity “Manager” because now “Staff” is partial overlap, it is acceptable to not show “Manager”.
- We changed the attribute “certificate” of a “Personal Trainer” into a relation because we want to keep track of the all kinds of certificate a trainer has and what date he/she got certified.

3. Schema & 4. Functional Dependencies

1. Staff

Table definition	Staff (StaffID: CHAR(20), FirstName: CHAR(20), LastName: CHAR(20), Email: CHAR(30), PhoneNumber: CHAR(20), SIN: CHAR(20))
Primary keys	StaffID
Candidate keys	{StaffID} {SIN}
Foreign keys	/
Other Constraints	FirstName, LastName, PhoneNumber, Email UNIQUE
Functional Dependency	StaffID → FirstName, LastName, PhoneNumber, Email, SIN SIN → StaffID, FirstName, LastName, PhoneNumber, Email

```
CREATE TABLE Staff (  
    StaffID      CHAR(20)    PRIMARY KEY,  
    FirstName    CHAR(20),  
    LastName     CHAR(20),  
    Email        CHAR(30),  
    PhoneNumber  CHAR(20),  
    SIN          CHAR(20)    UNIQUE  
);
```

2. Trainer

Table definition	Trainer (StaffID: CHAR(20))
Primary keys	StaffID
Candidate keys	{StaffID}
Foreign keys	FOREIGN KEY (StaffID) REFERENCES Staff
Other Constraints	FOREIGN KEY (StaffID) REFERENCES Staff ON DELETE NO ACTION ON UPDATE CASCADE
Functional Dependency	No non-trivial functional dependency.

```
CREATE TABLE Trainer (  
    StaffID          CHAR(20)    PRIMARY KEY,  
    FOREIGN KEY (StaffID) REFERENCES Staff  
        ON DELETE NO ACTION  
        ON UPDATE CASCADE  
);
```

3. Salesperson

Table definition	Salesperson (StaffID: CHAR(20) , TotalSaleAmount: INTEGER)
Primary keys	StaffID
Candidate keys	{StaffID}
Foreign keys	FOREIGN KEY (StaffID) REFERENCES Staff
Other Constraints	Set TotalSaleAmount DEFAULT value 0 FOREIGN KEY (StaffID) REFERENCES Staff ON DELETE NO ACTION ON UPDATE CASCADE
Functional Dependency	StaffID → TotalSaleAmount

```
CREATE TABLE Salesperson (  
    StaffID          CHAR(20)    PRIMARY KEY,  
    TotalSaleAmount  FLOAT       DEFAULT 0,  
    FOREIGN KEY (StaffID) REFERENCES Staff  
        ON DELETE NO ACTION  
        ON UPDATE CASCADE  
);
```

4. Product

Table definition	Product (<u>ProductID</u> : CHAR(20))
Primary keys	ProductID
Candidate keys	{ProductID}
Foreign keys	/
Other Constraints	/
Functional Dependency	No non-trivial functional dependency.

```
CREATE TABLE Product (  
    ProductID      CHAR(20)    PRIMARY KEY  
);
```

5. Membership

Table definition	Membership (ProductID: CHAR(20) , MonthlyFee: FLOAT, NumOfMonths: INTEGER)
Primary keys	ProductID
Candidate keys	{ProductID} {MonthlyFee, NumOfMonths}
Foreign keys	FOREIGN KEY (ProductID) REFERENCES Product
Other Constraints	MonthlyFee, NumOfMonths UNIQUE and NOT NULL , FOREIGN KEY (ProductID) REFERENCES Product ON DELETE CASCADE ON UPDATE CASCADE
Functional Dependency	ProductID → MonthlyFee, NumOfMonths MonthlyFee, NumOfMonths → ProductID

```
CREATE TABLE Membership (  
    ProductID      CHAR(20)    PRIMARY KEY,  
    MonthlyFee     FLOAT       UNIQUE NOT NULL,  
    NumOfMonths    INTEGER     UNIQUE NOT NULL,  
    FOREIGN KEY (ProductID) REFERENCES ProductID  
        ON DELETE CASCADE  
        ON UPDATE CASCADE  
);
```


6. PersonalTrainingPackage

Table definition	PersonalTrainingPackage (ProductID: CHAR(20) , EntryFee: FLOAT, NumOfSessions: INTEGER)
Primary keys	ProductID
Candidate keys	{ProductID} {EntryFee, NumOfSessions}
Foreign keys	FOREIGN KEY (ProductID) REFERENCES Product
Other Constraints	Entry Fee, NumOfSessions UNIQUE and NOT NULL , FOREIGN KEY (ProductID) REFERENCES Product ON DELETE CASCADE ON UPDATE CASCADE
Functional Dependency	ProductID → EntryFee, NumOfSessions EntryFee, NumOfSessions → ProductID

```
CREATE TABLE PersonalTrainingPackage (
    ProductID      CHAR(20)    PRIMARY KEY,
    EntryFee       FLOAT       UNIQUE NOT NULL,
    NumOfSessions  INTEGER     UNIQUE NOT NULL,
    FOREIGN KEY (ProductID) REFERENCES ProductID
        ON DELETE CASCADE
        ON UPDATE CASCADE
);
```

7. Order_Sell_renew

Table definition	Order_Sell_renew (<u>OrderNo</u> : CHAR(30), Date: DATE, TotalPrice: FLOAT, CustomerID : CHAR(20), StaffID : CHAR(20))
Primary keys	OrderNo
Candidate keys	{OrderNo}
Foreign keys	FOREIGN KEY (CustomerID) REFERENCES Customer, FOREIGN KEY (StaffID) REFERENCES Salesperson
Other Constraints	Date, CustomerID, StaffID NOT NULL , Set TotalPrice DEFAULT value 0, FOREIGN KEY (CustomerID) REFERENCES Customer ON DELETE NO ACTION ON UPDATE CASCADE, FOREIGN KEY (StaffID) REFERENCES Salesperson ON DELETE NO ACTION ON UPDATE CASCADE
Functional Dependency	OrderNo → Date, TotalPrice, StaffID, CustomerID

```
CREATE TABLE Order_Sell_Renew (
    OrderNo          CHAR(30) PRIMARY KEY,
    Date             DATE      NOT NULL,
    TotalPrice        FLOAT     DEFAULT 0,
    CustomerID        CHAR(20) NOT NULL,
    StaffID           CHAR(20) NOT NULL,
    FOREIGN KEY (CustomerID) REFERENCES Customer
        ON DELETE NO ACTION
        ON UPDATE CASCADE,
    FOREIGN KEY (StaffID) REFERENCES Salesperson
        ON DELETE NO ACTION
        ON UPDATE CASCADE
);
```

8. Includes

Table definition	Include (ProductID: CHAR(20), OrderNo: CHAR(30))
Primary keys	ProductID, OrderNo
Candidate keys	{ProductID, OrderNo}
Foreign keys	FOREIGN KEY (ProductID) REFERENCES Product, FOREIGN KEY (OrderNo) REFERENCES Order
Other Constraints	FOREIGN KEY (ProductID) REFERENCES Product ON DELETE NO ACTION ON UPDATE NO ACTION, FOREIGN KEY (OrderNo) REFERENCES Order ON DELETE CASCADE ON UPDATE CASCADE
Functional Dependency	No non-trivial functional dependency.

```
CREATE TABLE Includes (
    ProductID      CHAR(20),
    OrderNo        CHAR(30),
    PRIMARY KEY (ProductID, OrderNo),
    FOREIGN KEY (ProductID) REFERENCES Product,
        ON DELETE NO ACTION
        ON UPDATE NO ACTION
    FOREIGN KEY (OrderNo) REFERENCES Order
        ON DELETE CASCADE
        ON UPDATE CASCADE
);
```

9. AssessmentResult_Assess

Table definition	AssessmentResult_Assess (CustomerID: CHAR(20) , <u>Date: DATE</u> , Weight: FLOAT, Height: FLOAT, Bodyfat: FLOAT, BMI: FLOAT, MuscleMass: FLOAT)
Primary keys	CustomerID, Date
Candidate keys	{CustomerID, Date}
Foreign keys	FOREIGN KEY (CustomerID) REFERENCES Customer
Other Constraints	Set Weight, Height, Bodyfat, BMI, MuscleMass DEFAULT value 0 FOREIGN KEY (CustomerID) REFERENCES Customer ON DELETE CASCADE ON UPDATE CASCADE
Functional Dependency	CustomerID, Date → Weight, Bodyfat, BMI, MuscleMass, Height

```
CREATE TABLE AssessmentResult_Assess (
    CustomerID    CHAR(20),
    Date          DATE,
    Weight        FLOAT DEFAULT 0,
    Height        FLOAT DEFAULT 0,
    BMI           FLOAT DEFAULT 0,
    BodyFatPct    FLOAT DEFAULT 0,
    MuscleMass    FLOAT DEFAULT 0,
    PRIMARY KEY (CustomerID, Date),
    FOREIGN KEY (CustomerID) REFERENCES CustomerID
        ON DELETE CASCADE
        ON UPDATE CASCADE
);
```

10. Customer

Table definition	Customer (<u>CustomerID</u> : CHAR(20), FirstName: CHAR(20), LastName: CHAR(20), Email: CHAR(30), PhoneNumber: CHAR(20), MembershipExpireDate: DATE, PrivateLessonRemaining: INTEGER)
Primary keys	CustomerID
Candidate keys	{CustomerID} {Email, PhoneNumber, FirstName, LastName}
Foreign keys	/
Other Constraints	Set Email, PhoneNumber, FirstName, LastName UNIQUE , Set PrivateLessonRemaining DEFAULT value 0
Functional Dependency	CustomerID → PhoneNumber, MembershipExpireDate, FirstName, LastName, Email, PrivateLessonRemaining FirstName, LastName, PhoneNumber, Email → CustomerID, MembershipExpireDate, PrivateLessonRemaining

```
CREATE TABLE Customer (
    CustomerID      CHAR(20)    PRIMARY KEY,
    FirstName       CHAR(20)    UNIQUE,
    LastName        CHAR(20)    UNIQUE,
    Email           CHAR(30)    UNIQUE,
    PhoneNumber     CHAR(20)    UNIQUE,
    MembershipExpireDate DATE,
    PrivateLessonRemaining INTEGER DEFAULT 0
);
```

11. Rent

Table definition	Rent (CustomerID: CHAR(20) , EqID: CHAR(20) , EqType: CHAR(20) , Date: DATE)
Primary keys	CustomerID, EqID, EqType
Candidate keys	{CustomerID, EqID, EqType}
Foreign keys	FOREIGN KEY (CustomerID) REFERENCES Customer, FOREIGN KEY (EqID, EqType) REFERENCES Equipment_Inventory
Other Constraints	FOREIGN KEY (CustomerID) REFERENCES Customer ON DELETE CASCADE ON UPDATE CASCADE, FOREIGN KEY (EqID, EqType) REFERENCES Equipment_Inventory ON DELETE CASCADE ON UPDATE CASCADE
Functional Dependency	CustomerID, EqID, EqType → Date

```
CREATE TABLE Rent (
    CustomerID    CHAR(20),
    EqID          CHAR(20),
    EqType        CHAR(20),
    Date          DATE,
    PRIMARY KEY (CustomerID, EqID, EqType),
    FOREIGN KEY (CustomerID) REFERENCES Customer
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    FOREIGN KEY (EqID, EqType) REFERENCES Equipment_Inventory
        ON DELETE CASCADE
        ON UPDATE CASCADE
);
```

12. Equipment (violate BCNF, 3NF, see decomposition below)

Table definition	Equipment (<u>EqID</u> : CHAR(20), <u>EqType</u> : CHAR(20), Deposit:FLOAT, IsAvailable: CHAR(3))
Primary keys	EqID, EqType
Candidate keys	{EqID, EqType}
Foreign keys	/
Other Constraints	Set IsAvailable DEFAULT value 'YES', Set Deposit DEFAULT value 0
Functional Dependency	EqID, EqType → Deposit, IsAvailable EqType → Deposit

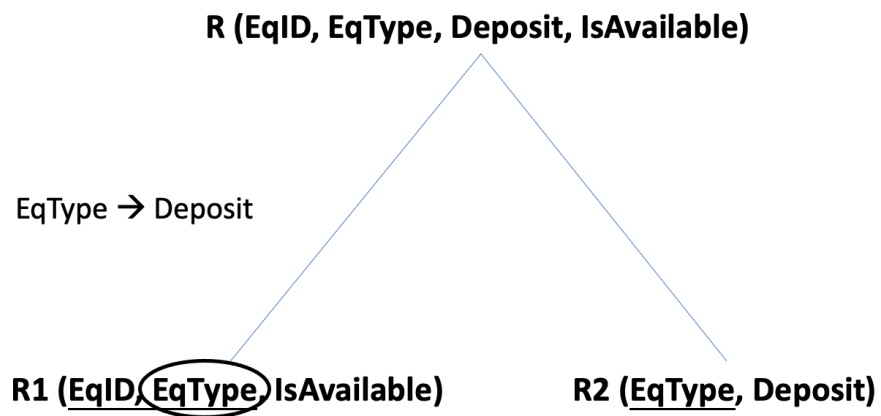
12.1 Equipment_Inventory(decomposed from Equipment)

Table definition	Equipment_Inventory (<u>EqID</u> : CHAR(20), <u>EqType</u> : CHAR(20), IsAvailable: CHAR(3))
Primary keys	EqID, EqType
Candidate keys	{EqID, EqType}
Foreign keys	FOREIGN KEY (EqType) REFERENCES Equipment_Deposit
Other Constraints	Set IsAvailable DEFAULT value 'YES', Set Deposit DEFAULT value 0
Functional Dependency	EqID, EqType → IsAvailable

12.2 Equipment_Deposit(decomposed from Equipment)

Table definition	Equipment_Inventory(EqType : CHAR(20), Deposit: FLOAT)
Primary keys	EqType
Candidate keys	{EqType}
Foreign keys	/
Other Constraints	Set Deposit DEFAULT value 0
Functional Dependency	EqType → Deposit

Decompose TABLE Equipment



```
CREATE TABLE Equipment_Inventory (  
    EqID          CHAR(20),  
    EqType        CHAR(20),  
    IsAvailable   CHAR(3)    DEFAULT 'YES',  
    PRIMARY KEY (EqID, EqType),  
    FOREIGN KEY (EqType) REFERENCES Equipment_Deposit  
);  
  
CREATE TABLE Equipment_Deposit (  
    EqType        CHAR(20)    PRIMARY KEY,  
    Deposit       FLOAT       DEFAULT 0  
);
```


13. Train

Table definition	Train (TimeSlot: INTEGER, Date: DATE, CustomerID: CHAR(20), StaffID: CHAR(20))
Primary keys	TimeSlot, Date, CustomerID, StaffID
Candidate keys	{TimeSlot, Date, CustomerID, StaffID}
Foreign keys	FOREIGN KEY (CustomerID) REFERENCES Customer, FOREIGN KEY (StaffID) REFERENCES Trainer
Other Constraints	FOREIGN KEY (CustomerID) REFERENCES Customer ON DELETE CASCADE ON UPDATE CASCADE, FOREIGN KEY (StaffID) REFERENCES Trainer ON DELETE CASCADE ON UPDATE CASCADE
Functional Dependency	No non-trivial functional dependency.

```
CREATE TABLE Train(
    TimeSlot      INTEGER,
    Date          DATE,
    CustomerID    CHAR(20),
    StaffID       CHAR(20),
    PRIMARY KEY (TimeSlot, Date, CustomerID, StaffID),
    FOREIGN KEY (CustomerID) REFERENCES Customer
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    FOREIGN KEY (StaffID) REFERENCES Trainer
        ON DELETE CASCADE
        ON UPDATE CASCADE
);
```

14. Certify

Table definition	Certify (Name: CHAR(20) , StaffID: CHAR(20) , Date: DATE)
Primary keys	Name, StaffID
Candidate keys	{Name, StaffID}
Foreign keys	FOREIGN KEY (Name) REFERENCES PersonalTrainerCertification, FOREIGN KEY (StaffID) REFERENCES Trainer
Other Constraints	FOREIGN KEY (Name) REFERENCES PersonalTrainerCertification ON DELETE NO ACTION ON DELETE NO ACTION, FOREIGN KEY (StaffID) REFERENCES Trainer ON DELETE CASCADE ON UPDATE CASCADE
Functional Dependency	Name, StaffID → Date

```
CREATE TABLE Certify (
    Name          CHAR(20),
    StaffID       CHAR(20),
    Date          DATE,
    PRIMARY KEY (Name, StaffID),
    FOREIGN KEY (Name) REFERENCES PersonalTrainerCertification
        ON DELETE NO ACTION
        ON UPDATE NO ACTION,
    FOREIGN KEY (StaffID) REFERENCES Trainer
        ON DELETE CASCADE
        ON UPDATE CASCADE
);
```

15. PersonalTrainerCertification

Table definition	PersonalTrainerCertification (<u>Name</u> : CHAR(20))
Primary keys	Name
Candidate keys	{Name}
Foreign keys	/
Other Constraints	/
Functional Dependency	No non-trivial functional dependency.

```
CREATE TABLE PersonalTrainerCertification(  
    Name          CHAR(20)    PRIMARY KEY  
);
```

16. GroupFitnessLesson_Lead (violate BCNF, 3NF, see decomposition below)

Table definition	GroupFitnessLesson_Lead (<u>StartTime</u> : TIME, <u>Date</u> : DATE, <u>ActivityType</u> : CHAR(20), NumOfSpotsLeft: INTEGER, Duration: INTEGER, StaffID : CHAR(20))
Primary keys	StartTime, Date, ActivityType
Candidate keys	{StartTime, Date, ActivityType}
Foreign keys	FOREIGN KEY (StaffID) REFERENCES Trainer
Other Constraints	Set Duration DEFAULT value 45, Set NumOfSpotsLeft DEFAULT value 0, Set StaffID DEFAULT value 'TBA', FOREIGN KEY (StaffID) REFERENCES Trainer ON DELETE NO ACTION ON UPDATE CASCADE
Functional Dependency	StartTime, Date, ActivityType → NumOfSpotsLeft, Duration, StaffID ActivityType → Duration

16.1 GroupLesson_Details (decomposed from GroupFitnessLesson_Lead)

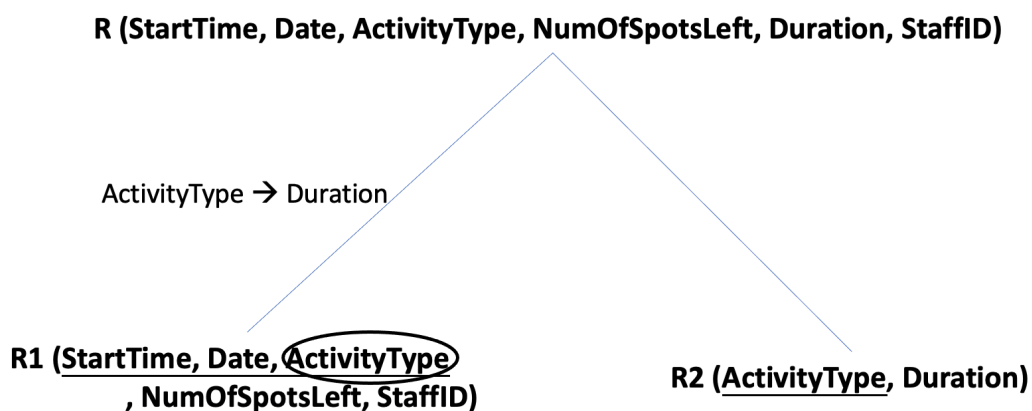
Table definition	Activity_Details (<u>StartTime</u> : TIME, <u>Date</u> : DATE, <u>ActivityType</u> : CHAR(20), NumOfSpotsLeft: INTEGER, StaffID : CHAR(20))
Primary keys	StartTime, Date, ActivityType
Candidate keys	{StartTime, Date, ActivityType}
Foreign keys	FOREIGN KEY (StaffID) REFERENCES Trainer, FOREIGN KEY (ActivityType) REFERENCES GroupLesson_Duration
Other Constraints	Set NumOfSpotsLeft DEFAULT value 0, Set StaffID DEFAULT value 'TBA', FOREIGN KEY (StaffID) REFERENCES Trainer ON DELETE NO ACTION ON UPDATE CASCADE
Functional Dependency	StartTime, Date, ActivityType → NumOfSpotsLeft, Duration, StaffID StartTime, Date, StaffID → NumOfSpotsLeft, Duration, ActivityType

16.2 GroupLesson_ActivityDuration (decomposed from GroupFitnessLesson_Lead)

Table definition	Activity_Duration (<u>ActivityType</u> : CHAR(20), Duration: INTEGER)
Primary keys	ActivityType

Candidate keys	{ActivityType}
Foreign keys	
Other Constraints	Set Duration DEFAULT value 45
Functional Dependency	ActivityType → Duration

Decompose TABLE GroupFitnessLesson_Lead



```
CREATE TABLE GroupLesson_ActivityDuration(
    ActivityType CHAR(20) PRIMARY KEY,
    Duration INTEGER DEFAULT 45
);

CREATE TABLE GroupLesson_Details(
    StartTime TIME,
    Date DATE,
    ActivityType CHAR(20),
    NumOfSpotsLeft INTEGER DEFAULT 20,
    StaffID CHAR(20) NOT NULL,
    PRIMARY KEY (StartTime, Date, ActivityType),
    FOREIGN KEY (StaffID) REFERENCES Trainer
        ON DELETE NO ACTION
        ON UPDATE CASCADE,
    FOREIGN KEY (ActivityType) REFERENCES GroupLesson_ActivityDuration
);
```

5. Normalization

12. Equipment (violate BCNF, 3NF, see decomposition below)

Table definition	Equipment (<u>EqID</u> : CHAR(20), <u>EqType</u> : CHAR(20), Deposit:FLOAT, IsAvailable: CHAR(3))
Primary keys	EqID, EqType
Candidate keys	{EqID, EqType}
Foreign keys	/
Other Constraints	Set IsAvailable DEFAULT value 'YES', Set Deposit DEFAULT value 0
Functional Dependency	EqID, EqType → Deposit, IsAvailable EqType → Deposit

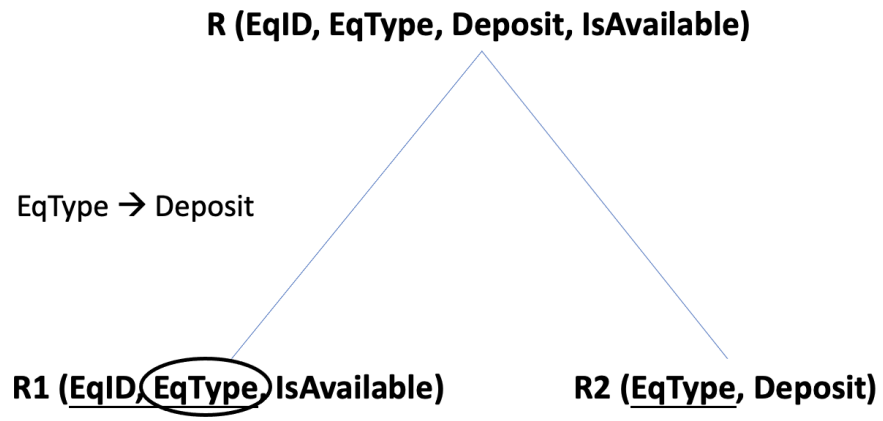
12.1 Equipment_Inventory(decomposed from Equipment)

Table definition	Equipment_Inventory (<u>EqID</u> : CHAR(20), <u>EqType</u> : CHAR(20), IsAvailable: CHAR(3))
Primary keys	EqID, EqType
Candidate keys	{EqID, EqType}
Foreign keys	FOREIGN KEY (EqType) REFERENCES Equipment_Deposit
Other Constraints	Set IsAvailable DEFAULT value 'YES', Set Deposit DEFAULT value 0
Functional Dependency	EqID, EqType → IsAvailable

12.2 Equipment_Deposit(decomposed from Equipment)

Table definition	Equipment_Deposit(<u>EqType</u> : CHAR(20), Deposit: FLOAT)
Primary keys	EqType
Candidate keys	{EqType}
Foreign keys	/
Other Constraints	Set Deposit DEFAULT value 0
Functional Dependency	EqType → Deposit

Decompose TABLE Equipment



16. GroupFitnessLesson_Lead (violate BCNF, 3NF, see decomposition below)

Table definition	GroupFitnessLesson_Lead (<u>StartTime</u> : TIME, <u>Date</u> : DATE, <u>ActivityType</u> : CHAR(20), NumOfSpotsLeft: INTEGER, Duration: INTEGER, StaffID : CHAR(20))
Primary keys	StartTime, Date, ActivityType
Candidate keys	{StartTime, Date, ActivityType}
Foreign keys	FOREIGN KEY (StaffID) REFERENCES Trainer
Other Constraints	Set Duration DEFAULT value 45, Set NumOfSpotsLeft DEFAULT value 0, Set StaffID DEFAULT value 'TBA', FOREIGN KEY (StaffID) REFERENCES Trainer ON DELETE NO ACTION ON UPDATE CASCADE
Functional Dependency	StartTime, Date, ActivityType → NumOfSpotsLeft, Duration, StaffID ActivityType → Duration

16.1 GroupLesson_Details (decomposed from GroupFitnessLesson_Lead)

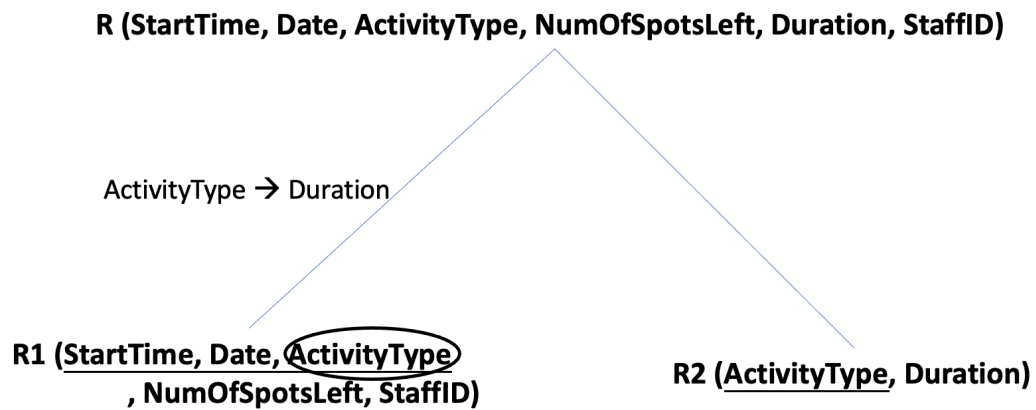
Table definition	Activity_Details (<u>StartTime</u> : TIME, <u>Date</u> : DATE, <u>ActivityType</u> : CHAR(20), NumOfSpotsLeft: INTEGER, StaffID : CHAR(20))
Primary keys	StartTime, Date, ActivityType
Candidate keys	{StartTime, Date, ActivityType}
Foreign keys	FOREIGN KEY (StaffID) REFERENCES Trainer, FOREIGN KEY (ActivityType) REFERENCES GroupLesson_Duration
Other Constraints	Set NumOfSpotsLeft DEFAULT value 0, Set StaffID DEFAULT value 'TBA', FOREIGN KEY (StaffID) REFERENCES Trainer ON DELETE NO ACTION ON UPDATE CASCADE
Functional Dependency	StartTime, Date, ActivityType → NumOfSpotsLeft, Duration, StaffID StartTime, Date, StaffID → NumOfSpotsLeft, Duration, ActivityType

16.2 GroupLesson_ActivityDuration (decomposed from GroupFitnessLesson_Lead)

Table definition	Activity_Duration (<u>ActivityType</u> : CHAR(20), Duration: INTEGER)
Primary keys	ActivityType

Candidate keys	{ActivityType}
Foreign keys	
Other Constraints	Set Duration DEFAULT value 45
Functional Dependency	ActivityType → Duration

Decompose TABLE GroupFitnessLesson_Lead



6. SQL CREATE TABLE

https://github.students.cs.ubc.ca/CPSC304-2021W-T1/project_g5z2b_q0d7e_v0y2b

File named M2.sql

```
CREATE TABLE Staff (  
    StaffID          CHAR(20)    PRIMARY KEY,  
    FirstName        CHAR(20),  
    LastName         CHAR(20),  
    Email            CHAR(30),  
    PhoneNumber      CHAR(20),  
    SIN              CHAR(20)    UNIQUE  
);  
  
CREATE TABLE Trainer (  
    StaffID          CHAR(20)    PRIMARY KEY,  
    FOREIGN KEY (StaffID) REFERENCES Staff  
        ON DELETE NO ACTION  
        ON UPDATE CASCADE  
);  
  
CREATE TABLE Salesperson (  
    StaffID          CHAR(20)    PRIMARY KEY,  
    TotalSaleAmount  FLOAT       DEFAULT 0,  
    FOREIGN KEY (StaffID) REFERENCES Staff  
        ON DELETE NO ACTION  
        ON UPDATE CASCADE  
);  
  
CREATE TABLE Product (  
    ProductID       CHAR(20)    PRIMARY KEY  
);  
  
CREATE TABLE Membership (  
    ProductID       CHAR(20)    PRIMARY KEY,  
    MonthlyFee      FLOAT       UNIQUE NOT NULL,  
    NumOfMonths     INTEGER     UNIQUE NOT NULL,  
    FOREIGN KEY (ProductID) REFERENCES ProductID  
        ON DELETE CASCADE  
        ON UPDATE CASCADE  
);  
  
CREATE TABLE PersonalTrainingPackage (  
    StaffID          CHAR(20)    PRIMARY KEY,  
    ProductID       CHAR(20)    PRIMARY KEY,  
    MonthlyFee      FLOAT       UNIQUE NOT NULL,  
    NumOfMonths     INTEGER     UNIQUE NOT NULL,  
    FOREIGN KEY (StaffID) REFERENCES Staff  
        ON DELETE NO ACTION  
        ON UPDATE CASCADE  
    FOREIGN KEY (ProductID) REFERENCES ProductID  
        ON DELETE CASCADE  
        ON UPDATE CASCADE  
);
```

```

ProductID      CHAR(20)  PRIMARY KEY,
EntryFee       FLOAT     UNIQUE NOT NULL,
NumOfSessions  INTEGER   UNIQUE NOT NULL,
FOREIGN KEY (ProductID) REFERENCES ProductID
    ON DELETE CASCADE
    ON UPDATE CASCADE
);

```

```

CREATE TABLE Order_Sell_Renew (
    OrderNo      CHAR(30) PRIMARY KEY,
    Date         DATE     NOT NULL,
    TotalPrice   FLOAT     DEFAULT 0,
    CustomerID   CHAR(20) NOT NULL,
    StaffID      CHAR(20) NOT NULL,
    FOREIGN KEY (CustomerID) REFERENCES Customer
        ON DELETE NO ACTION
        ON UPDATE CASCADE,
    FOREIGN KEY (StaffID) REFERENCES Salesperson
        ON DELETE NO ACTION
        ON UPDATE CASCADE
);

```

```

CREATE TABLE Includes (
    ProductID    CHAR(20),
    OrderNo      CHAR(30),
    PRIMARY KEY (ProductID, OrderNo),
    FOREIGN KEY (ProductID) REFERENCES Product,
        ON DELETE NO ACTION
        ON UPDATE NO ACTION
    FOREIGN KEY (OrderNo) REFERENCES Order
        ON DELETE CASCADE
        ON UPDATE CASCADE
);

```

```

CREATE TABLE AssessmentResult_Assess (
    CustomerID   CHAR(20),
    Date         DATE,
    Weight       FLOAT  DEFAULT 0,
    Height       FLOAT  DEFAULT 0,
    BMI          FLOAT  DEFAULT 0,
    BodyFatPct   FLOAT  DEFAULT 0,
    MuscleMass   FLOAT  DEFAULT 0,

```

```

        PRIMARY KEY (CustomerID, Date),
        FOREIGN KEY (CustomerID) REFERENCES CustomerID
            ON DELETE CASCADE
            ON UPDATE CASCADE
    );

CREATE TABLE Customer (
    CustomerID      CHAR(20)    PRIMARY KEY,
    FirstName       CHAR(20)    UNIQUE,
    LastName        CHAR(20)    UNIQUE,
    Email           CHAR(30)    UNIQUE,
    PhoneNumber     CHAR(20)    UNIQUE,
    MembershipExpireDate DATE,
    PrivateLessonRemaining INTEGER DEFAULT 0
);

CREATE TABLE Rent (
    CustomerID      CHAR(20),
    EqID            CHAR(20),
    EqType          CHAR(20),
    Date            DATE,
    PRIMARY KEY (CustomerID, EqID, EqType),
    FOREIGN KEY (CustomerID) REFERENCES Customer
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    FOREIGN KEY (EqID, EqType) REFERENCES Equipment_Inventory
        ON DELETE CASCADE
        ON UPDATE CASCADE
);

CREATE TABLE Equipment_Inventory (
    EqID            CHAR(20),
    EqType          CHAR(20),
    IsAvailable     CHAR(3)     DEFAULT 'YES',
    PRIMARY KEY (EqID, EqType),
    FOREIGN KEY (EqType) REFERENCES Equipment_Deposit
);

CREATE TABLE Equipment_Deposit (
    EqType          CHAR(20)    PRIMARY KEY,
    Deposit         FLOAT       DEFAULT 0
);

```

```

);

CREATE TABLE Train(
    TimeSlot      INTEGER,
    Date          DATE,
    CustomerID    CHAR(20),
    StaffID       CHAR(20),
    PRIMARY KEY (TimeSlot, Date, CustomerID, StaffID),
    FOREIGN KEY (CustomerID) REFERENCES Customer
                        ON DELETE CASCADE
                        ON UPDATE CASCADE,
    FOREIGN KEY (StaffID) REFERENCES Trainer
                        ON DELETE CASCADE
                        ON UPDATE CASCADE
);

CREATE TABLE Certify (
    Name          CHAR(20),
    StaffID       CHAR(20),
    Date          DATE,
    PRIMARY KEY (Name, StaffID),
    FOREIGN KEY (Name) REFERENCES PersonalTrainerCertification
                        ON DELETE NO ACTION
                        ON UPDATE NO ACTION,
    FOREIGN KEY (StaffID) REFERENCES Trainer
                        ON DELETE CASCADE
                        ON UPDATE CASCADE
);

CREATE TABLE PersonalTrainerCertification(
    Name          CHAR(20)    PRIMARY KEY
);

CREATE TABLE GroupLesson_ActivityDuration(
    ActivityType  CHAR(20)    PRIMARY KEY,
    Duration      INTEGER     DEFAULT 45
);

CREATE TABLE GroupLesson_Details(
    StartTime     TIME,
    Date          DATE,
    ActivityType  CHAR(20),

```

```
NumOfSpotsLeft INTEGER      DEFAULT 20,  
StaffID        CHAR(20)     NOT NULL,  
PRIMARY KEY (StartTime, Date, ActivityType),  
FOREIGN KEY (StaffID) REFERENCES Trainer  
            ON DELETE NO ACTION  
            ON UPDATE CASCADE,  
FOREIGN KEY (ActivityType) REFERENCES GroupLesson_ActivityDuration  
);
```

7. SQL INSERT TUPLES

https://github.students.cs.ubc.ca/CPSC304-2021W-T1/project_g5z2b_q0d7e_v0y2b

File named M2InsertTuples.sql

```
insert into Staff values ('101', 'Bob', 'Smith', 'Bobssmith@gmail.com', '6238762202',
'123456654321117');

insert into Staff values ('102', 'Kayln', 'Walter', 'Kaylnwwalter@gmail.com',
'4232345552', '123456654321251');

insert into Staff values ('103', 'Alex', 'Mark', 'Alexmmark@gmail.com', '5524442345',
'123456654328811');

insert into Staff values ('104', 'Kara', 'Rops', 'Kararrops@gmail.com', '332559022',
'123456654321123');

insert into Staff values ('105', 'Candace', 'Conne', 'Candaceconne@gmail.com',
'4545232357', '123456654324511');

insert into Staff values ('201', 'Liam', 'Jones', 'Liamjjones@gmail.com', '6238762202',
'123456654321111');

insert into Staff values ('202', 'Elijah', 'Browns', 'Elijahbbrowns@gmail.com',
'4232345552', '123456654321131');

insert into Staff values ('203', 'Lucas', 'Taylor', 'Lucasttaylor@gmail.com',
'5524442345', '123456654321111');

insert into Staff values ('204', 'Charlotte', 'Clark', 'CharlottteccClark@gmail.com',
'332559022', '123456654321112');

insert into Staff values ('205', 'Emma', 'Scott', 'Emmasscott@gmail.com', '4545232357',
'123456654321111');

insert into Trainer values ('101');
insert into Trainer values ('102');
insert into Trainer values ('103');
insert into Trainer values ('104');
insert into Trainer values ('105');

insert into Salesperson values ('201');
insert into Salesperson values ('202');
insert into Salesperson values ('203');
insert into Salesperson values ('204');
insert into Salesperson values ('205');

insert into Product values ('11111111');
insert into Product values ('11111112');
insert into Product values ('11111113');
insert into Product values ('11111114');
insert into Product values ('11111115');
```

```

insert into Membership values ('11111111', 75.00, 1);
insert into Membership values ('11111112', 60.00, 2);
insert into Membership values ('11111113', 44.33, 3);
insert into Membership values ('11111114', 36.33, 6);
insert into Membership values ('11111115', 26.67, 12);

insert into PersonalTrainingPackage values ('11112221', 50.00, 5);
insert into PersonalTrainingPackage values ('11112222', 46.25, 16);
insert into PersonalTrainingPackage values ('11112223', 42.19, 32);
insert into PersonalTrainingPackage values ('11112224', 36.67, 72);
insert into PersonalTrainingPackage values ('11112225', 29.00, 108);
insert into PersonalTrainingPackage values ('11113331', 55.00, 5);
insert into PersonalTrainingPackage values ('11113332', 52.30, 16);
insert into PersonalTrainingPackage values ('11113333', 47.29, 32);
insert into PersonalTrainingPackage values ('11113334', 39.34, 72);
insert into PersonalTrainingPackage values ('11113335', 29.82, 108);

insert into Customer values ('10001', 'Amelia', 'Nelly', 'Ameliannelly@gmail.com',
'4039543991', '2021-12-13', 2);
insert into Customer values ('10002', 'Ava', 'Freda', 'Avaffreda@gmail.com',
'2503652794', '2022-03-11', 23);
insert into Customer values ('10003', 'Osca', 'Jim', 'Oscajjim@gmail.com',
'2895715072', '2022-08-24', 15);
insert into Customer values ('10004', 'Ivy', 'Du', 'Ivyddu@gmail.com', '8199567549',
'2021-12-13', 2);
insert into Customer values ('10005', 'Arthur', 'Aubry', 'Arthuraaubry@gmail.com',
'2508329440', '2022-01-02', 10);

insert into Order_Sell_Renew values ('1111111111', 2014-03-15, 320.04, '10004',
'203');
insert into Order_Sell_Renew values ('1111111112', 2019-11-13, 2832.48, '10003',
'203');
insert into Order_Sell_Renew values ('1111111113', 2020-07-08, 1350.08, '10004',
'202');
insert into Order_Sell_Renew values ('1111111114', 2021-05-26, 132.99, '10001',
'201');
insert into Order_Sell_Renew values ('1111111115', 2021-05-26, 3220.56, '10001',
'201');

insert into Includes values ('11112225', '1111111111');
insert into Includes values ('11111113', '1111111114');
insert into Includes values ('11113334', '1111111112');

```



```

insert into Includes values ('11112223', '111111113');
insert into Includes values ('11113335', '111111115');

insert into AssessmentResult_Assess values ('10001', '2021-01-13', 67.20, 168.00,
23.00, 11.50, 34.30);
insert into AssessmentResult_Assess values ('10002', '2021-05-21', 70.90, 173.00,
23.70, 11.90, 33.50);
insert into AssessmentResult_Assess values ('10003', '2021-07-04', 84.90, 179.00,
26.50, 22.30, 37.80);
insert into AssessmentResult_Assess values ('10004', '2020-09-08', 55.87, 162,17,
21.28, 22.50, 23.07);
insert into AssessmentResult_Assess values ('10005', '2021-09-22', 0, 0, 0, 0, 0);

insert into Equipment_Inventory values ('1', 'YogaMat', 'YES');
insert into Equipment_Inventory values ('1', 'Dumbbel', 'NO');
insert into Equipment_Inventory values ('2', 'Dumbbel', 'NO');
insert into Equipment_Inventory values ('1', 'JumpRope', 'NO');
insert into Equipment_Inventory values ('2', 'JumpRope', 'NO');
insert into Equipment_Inventory values ('3', 'JumpRope', 'NO');
insert into Equipment_Inventory values ('1', 'Kettlebell', 'NO');
insert into Equipment_Inventory values ('2', 'Kettlebell', 'NO');
insert into Equipment_Inventory values ('1', 'SitUpsAssistant', 'YES');
insert into Equipment_Inventory values ('1', 'RollerWheel', 'YES');
insert into Equipment_Inventory values ('2', 'RollerWheel', 'YES');

insert into Equipment_Deposit values ('YogaMat', 10.00);
insert into Equipment_Deposit values ('Dumbbel', 10.00);
insert into Equipment_Deposit values ('JumpRope', 5.00);
insert into Equipment_Deposit values ('Kettlebell', 20.00);
insert into Equipment_Deposit values ('SitUpsAssistant', 5.00);
insert into Equipment_Deposit values ('RollerWheel', 5.55);

insert into Train values (15, '2021-01-13', '10001', '105');
insert into Train values (15, '2021-01-13', '10003', '105');
insert into Train values (19, '2021-01-13', '10003', '101');
insert into Train values (15, '2021-01-20', '10004', '104');
insert into Train values (19, '2021-01-13', '10005', '102');

insert into Certify values ('NASM', '101', '2017-01-10');
insert into Certify values ('ACE', '102', '2017-01-10');
insert into Certify values ('NASM', '101', '2020-01-03');
insert into Certify values ('NASM', '104', '2017-11-20');

```

```
insert into Certify values ('NASM', '105', '2017-01-10');
insert into Certify values ('FitnessWorld', '101', '2020-01-03');
insert into Certify values ('AcademyofPersonalTrainingEducation', '104',
'2017-11-20');
insert into Certify values ('glpti', '105', '2017-01-10');

insert into PersonalTrainerCertification values ('NASM');
insert into PersonalTrainerCertification values ('ACE');
insert into PersonalTrainerCertification values ('FitnessWorld');
insert into PersonalTrainerCertification values
('AcademyofPersonalTrainingEducation');
insert into PersonalTrainerCertification values ('glpti');

insert into GroupLesson_ActivityDuration values ('Yoga', 90);
insert into GroupLesson_ActivityDuration values ('Cycle', 45);
insert into GroupLesson_ActivityDuration values ('Lift', 45);
insert into GroupLesson_ActivityDuration values ('BoxingBootcamp', 45);
insert into GroupLesson_ActivityDuration values ('Core', 45);

insert into GroupLesson_Details values ('17:00:00', '2020-01-10', 'Yoga', 20, '101');
insert into GroupLesson_Details values ('17:00:00', '2020-01-15', 'Yoga', 15, '101');
insert into GroupLesson_Details values ('18:30:00', '2020-01-10', 'Cycle', 3, '104');
insert into GroupLesson_Details values ('17:00:00', '2020-01-15', 'Lift', 15, '102');
insert into GroupLesson_Details values ('17:00:00', '2020-01-17', 'BoxingBootcamp',
20, '103');
insert into GroupLesson_Details values ('17:00:00', '2020-01-18', 'Core', 20, '105');
```

8. Queries

Queries we plan to use to fulfill the requirements in milestone 5

Insert Operation:

- Insert an assessment result of a customer to the AssessmentResult_Assess table.
- Insert an activity type and its duration to the GroupLesson_ActivityDuration table.

Delete Operation:

- This is a cascade-on-delete situation. Delete a product from the Product table. Because a product is either a Membership or a PersonalTrainingPackage, when a product is deleted, the referenced Membership/PersonalTrainingPackage will also be deleted.

Update Operation:

- Update the lead Trainer in one group lesson in GroupLesson_Details table.
- Update the deposit of one type of equipment in Equipment_Deposit table.

Selection:

- From the Customer table, select all the email of customers whose membership is expiring within certain date, or the private lesson remaining is less than a certain number, so that an notification will be send to their email.
- From the Salespersons table, select the staffID of salespersons whose total sales amount is higher than a certain number.

Topics we have not covered in class yet:

Projection:

Join:

Aggregation with Group By:

Aggregation with Having:

Nested Aggregation with Group By:

Division: