University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: 2
Date: Feb 24, 2023
Group Number: 1 (project mentor: Suki)

Name	Student Number	CS Alias (userid)	Preferred Email Address
Noreen Chan	89278618	s4v2b	chan.noreen609@gmail.com
Andy Liang	47847876	b4e3b	andyliang1000@gmail.com
Jerry Shao	84982321	c1i3b	shaojerry@icloud.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

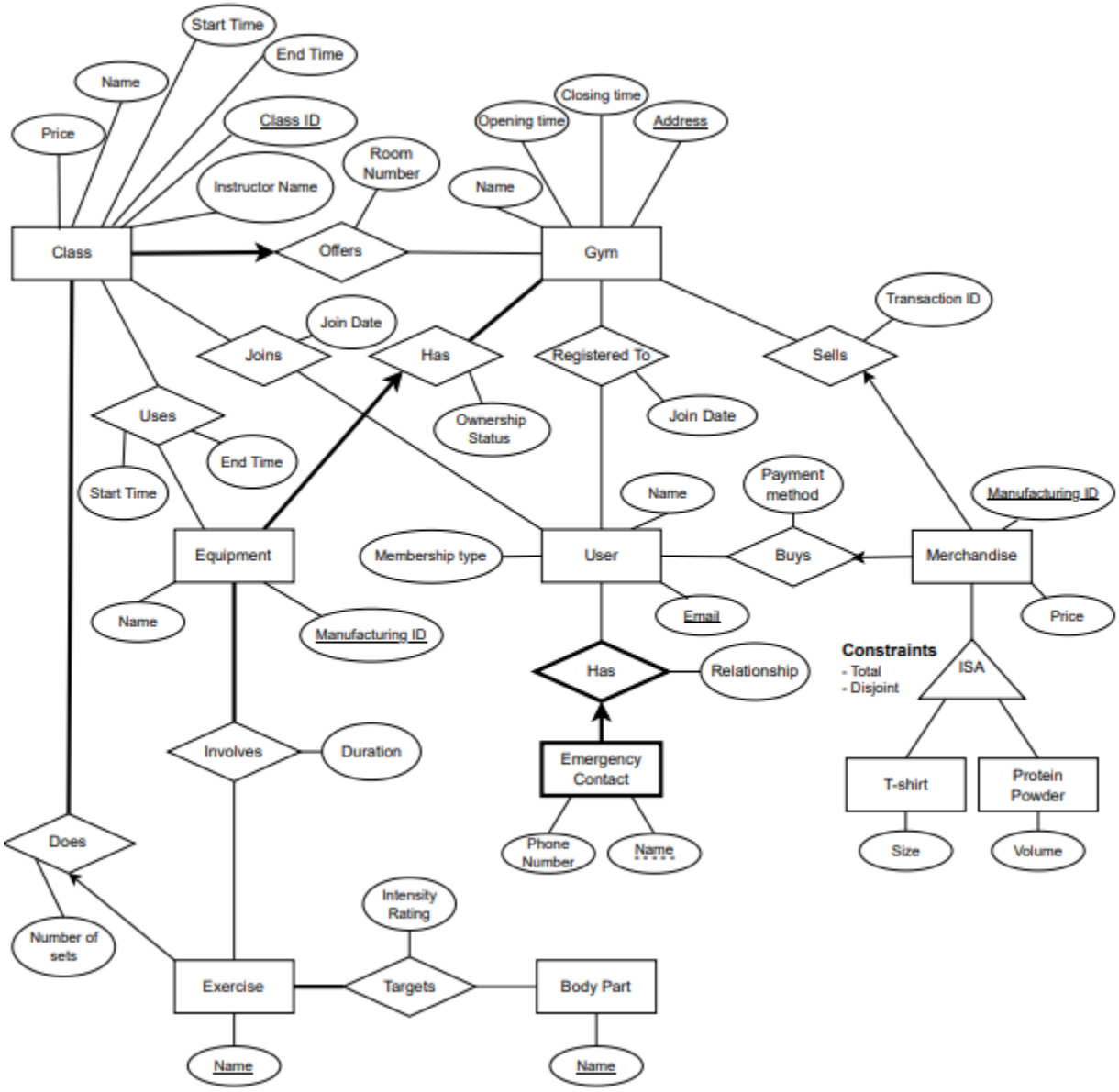
2. Project Summary

This application is a management tool for gym managers to keep track of gym logistics and users. Gym managers can query information like a list of users, equipment information such as exercises that use it, and the body parts that those exercises target.

3. ER Diagram

Changes

- “Uses” in the Exercise-Uses-Equipment relation has been renamed to “Involves”, to avoid having 2 “Uses” tables.
- User “Visits” Gym has been changed to User “registeredTo” Gym as it makes more sense to track this, as opposed to tracking every user’s Visit.
    - Consequently, “Check-in-time” has been changed to “Join date”



4. Schema

Combine:

- Class+Offers
- Exercise+Does
- Equipment+Has
- Emergency Contact+Has
- Merchandise+Sells

Note: Primary Keys are underlined, **Foreign Keys** are bolded.

```
Offers_Class(
      Price:                integer,
      Name:                 string,
      Start_time:           TIME,
      End_time:             TIME,
      Class_ID:             string,
      Instructor_name:      string,
      Room_number:          integer,
      Gym_Address:          string
)

Does_Exercise(
      Number_of_sets: integer,
      Exercise_Name:  string,
      Class_ID:       string
)

Has_Equipment(
      Equipment_name:           string,
      Manufacturing_ID:         string,
      Ownership_status:         string,
      Gym_Address:              string
)

Has_EmergencyContact(
      Email:     string,
      Phone_number:  integer,
      Name:          string,
      Relationship:  string
)
```

```
Sells_Merchandise(
      Gym_Address:       string,
      Transaction_ID:    string,
      Manufacturing_ID:  string,
      Price:             number
)

User(
      Membership_type:  string,
      Email:            string,
      Name:             string
)

Gym(
      Name:          string,
      Opening_time:  TIME,
      Closing_time:  TIME,
      Address:       string,
)

Involves(
      Duration:                    string,
      Exercise_Name:               string,
      Equipment_Manufacturing_ID:  string
)

BodyPart(
      Name: string
)

Targets(
      Exercise_Name:       string,
      BodyPart_name:       string,
      Intensity_rating:    integer
)

Uses(
      Class_ID:              string,
      Manufacturing_ID:      string,
      Start_time:            TIME,
      End_time:              TIME
)
```

```
Joins(
      Join_date:       TIME,
      User_Email:      string,
      Class_ID:        string
)

Buys(
      Payment_method:          string,
      Merch_Manufacturing_ID: string,
      User_Email:              string
)

T-shirt(
      Manufacturing_ID: string,
      Size:             string
)

ProteinPowder(
      Manufacturing_ID: string,
      Volume:           number
)

RegisteredTo(
      Join_date:        TIME,
      User_Email:           string,
      Gym_Address:          string
)
```

5. Functional Dependencies

Offers_class:

- Class_ID → Price, Name, Start_time, End_time, Instructor_name, Room_number, Gym_address
- Start_time, End_time, Instructor_name → Room_number

Does_Exercise:

- Exercise_name → Number_of_sets, Class_ID

Has_Equipment:

- Manufacturing_ID → Equipment_name, Ownership_status, Gym_address

Has_EmergencyContact:

- Email, Name → Relationship, Phone_number

Sells_Merchandise:

- Manufacturing_ID → Transaction_ID, Gym_address, Price

User:

- Email → Name, Membership_type

Gym:

- Address → Name, Opening_time, Closing_time

Involves:

- Exercise_name, Equipment_manufacturing_ID → Duration

BodyPart: (TRIVIAL)

Targets:

- Exercise_name, BodyPart_name → Intensity_rating

Uses:

- Class_ID, Manufacturing_ID → Start_time, End_time

Joins:

Class_ID, User_email → Join_date

Buys (DONE):

- Merchandise, User_email → Payment_method

RegisteredTo (DONE):

- User_email, Gym_address → Join_date

6. Normalization

Definition 1:
BCNF: All determinants that determine a subset of the attributes must determine all attributes.

6a) Offers_Class

Scratch work:

`Offers_Class(Price: integer, Name: string, Start_time: TIME, End_time: TIME, Class_ID: string, Instructor_name: string, Room_number: integer, **Gym_Address**: string)`

FDs:

- Class_ID → Price, Name, Start_time, End_time, Instructor_name, Room_number, Gym_address
- Start_time, End_time, Instructor_name → Room_number

Since { Start_time, End_time, Instructor_name } determines only a subset of the attributes, this is not in BCNF.

Decompose on "Start_time, End_time, Instructor_name → Room_number":

- `Offers_Class_1(Price: integer, Name: string, **Start_time**: TIME, **End_time**: TIME, Class_ID: string, **Instructor_name**: string, **Gym_Address**: string)`
- `Offers_Class_2(Start_time: TIME, End_time: TIME, Instructor_name: string, Room_number: integer)`

Now tables are in BCNF as per Definition 1.

List of Tables:

- `Offers_Class_1(Price: integer, Name: string, **Start_time**: TIME, **End_time**: TIME, Class_ID: string, **Instructor_name**: string, **Gym_Address**: string)`
- `Offers_Class_2(Start_time: TIME, End_time: TIME, Instructor_name: string, Room_number: integer)`

Primary Keys:

- `Offers_Class_1: Class_ID`
- `Offers_Class_2: Start_time, End_time, Instructor_name`

Candidate Keys: N/A (outside of primary keys)

Foreign Keys:

- `Offers_Class_1:`
  - **`Gym_Address`** `(references Gym)`
  - **`Start_time, End_time, Instructor_name`** `(ref. Offers_Class_2)`

6b) Does_Exercise

This is already in BCNF since all attributes that determine other attributes determine all of them.

List of Tables:

- Does_Exercise(Number_of_sets: integer, Exercise_Name: string, **Class_ID:** string)

Primary Keys:

- Exercise_Name

Candidate Keys: N/A outside of primary keys

Foreign Keys:

- **Class_ID** (references Offers_Class_1)

6c) Has_Equipment

This is already in BCNF since all attributes that determine other attributes determine all of them.

List of Tables:

- Has_Equipment(Equipment_name: string, Manufacturing_ID: string, Ownership_status: string, **Gym_Address:** string)

Primary Keys:

- Manufacturing_ID

Candidate Keys: N/A outside of primary keys

Foreign Keys:

- **Gym_Address** (references Gym)

6d) Has_EmergencyContact

This is already in BCNF since all attributes that determine other attributes determine all of them.

List of Tables:

- Has_EmergencyContact(**Email**: string, Phone_number: integer, Name: string,Relationship: string)

Primary Keys:

- **Email**, Name

Candidate Keys: N/A outside of primary keys

Foreign Keys:

- **Email** (references User)

6e) Sells_Merchandise

This is already in BCNF since all attributes that determine other attributes determine all of them.

List of Tables:

- Sells_Merchandise(**Gym_Address**: string, Transaction_ID: string, Manufacturing_ID: string, Price: number)

Primary Keys:

- Manufacturing_ID

Candidate Keys: N/A outside of primary keys

Foreign Keys:

- **Gym_Address** (references Gym)

6f) User

This is already in BCNF since all attributes that determine other attributes determine all of them.

List of Tables:

- Sells_Merchandise(**Gym_Address**: string, Transaction_ID: string, Manufacturing_ID: string, Price: number)

Primary Keys:

- Manufacturing_ID

Candidate Keys: N/A outside of primary keys

Foreign Keys:

- **Gym_Address** (references Gym)

6g) Gym

This is already in BCNF since all attributes that determine other attributes determine all of them.

List of Tables:
Gym(Name: string, Opening_time: TIME, Closing_time: TIME, Address: string)

Primary Keys:

- Manufacturing_ID

Candidate Keys: N/A outside of primary keys

Foreign Keys: N/A

6h) Involves

This is already in BCNF since all attributes that determine other attributes determine all of them.

List of Tables:

- `Involves(Duration: string, **Exercise_Name**: string, **Equipment_Manufacturing_ID**: string)`

Primary Keys:

- **Equipment_Manufacturing_ID**, **Exercise_Name**

Candidate Keys: N/A outside of primary keys

Foreign Keys:

- **Exercise_Name** (references Does_Exercise)
- **Equipment_Manufacturing_ID** (references Has_Equipment)

6i) BodyPart

This is already in BCNF since all attributes that determine other attributes determine all of them.

List of Tables:

- `BodyPart(Name: string)`

Primary Keys:

- Name

Candidate Keys: N/A outside of primary keys

Foreign Keys: N/A

6j) Targets

This is already in BCNF since all attributes that determine other attributes determine all of them.

List of Tables:

- `Targets(**Exercise_Name**: string, **BodyPart_name**: string, Intensity_rating: integer)`

Primary Keys:

- **Exercise_Name**, **BodyPart_name**

Candidate Keys: N/A outside of primary keys

Foreign Keys:

- **Exercise_Name** (references Does_Exercise)
- **BodyPart_name** (references BodyPart)

6k) Uses

This is already in BCNF since all attributes that determine other attributes determine all of them.

List of Tables:

- Uses(**Class_ID**: string, **Manufacturing_ID**: string, Start_time: TIME, End_time: TIME)

Primary Keys:

- **Manufacturing_ID**, **Class_ID**

Candidate Keys: N/A outside of primary keys

Foreign Keys:

- **Manufacturing_ID** (references Has_Equipment)
- **Class_ID** (references Offers_Class_1)

6l) Joins

This is already in BCNF since all attributes that determine other attributes determine all of them.

List of Tables:

- Joins(Join_date: TIME, **User_Email**: string, **Class_ID**: string)

Primary Keys:

- **User_Email**, **Class_ID**

Candidate Keys: N/A outside of primary keys
Foreign Keys:

- **User_Email** (references User)
- **Class_ID** (references Offers_Class_1)

6m) Buys

This is already in BCNF since all attributes that determine other attributes determine all of them.

List of Tables:

- Buys(Payment_method: string, **Merch_Manufacturing_ID**: string, **User_Email**: string)

Primary Keys:

- **User_Email**, **Class_ID**

Candidate Keys: N/A outside of primary keys
Foreign Keys:

- **User_Email** (references User)
- **Merch_Manufacturing_ID** (references Sells_Merchandise)

6n) T-shirt

This is already in BCNF since all attributes that determine other attributes determine all of them.

List of Tables:

- `T-shirt(`**`Manufacturing_ID`**`: string, Size: string)`

Primary Keys:

- **`Manufacturing_ID`**

Candidate Keys: N/A outside of primary keys

Foreign Keys:

- **`Manufacturing_ID`** `(references Sells_Merchandise)`

6o) ProteinPowder

This is already in BCNF since all attributes that determine other attributes determine all of them.

List of Tables:

- `ProteinPowder(Manufacturing_ID: string, Volume: number)`

Primary Keys:

- `Manufacturing_ID`

Candidate Keys: N/A outside of primary keys

Foreign Keys:

- `Manufacturing_ID (references Sells_Merchandise)`

6p) RegisteredTo

This is already in BCNF since all attributes that determine other attributes determine all of them.

List of Tables:

- `RegisteredTo(Join_date: TIME,` **`User_Email`**`: string,` **`Gym_Address`**`: string)`

Primary Keys:

- **`User_Email`**, **`Gym_Address`**

Candidate Keys: N/A outside of primary keys
Foreign Keys:

- **`User_Email`** `(references User)`
- **`Gym_Address`** `(references Gym)`

7. Create Table Statements

7a) Offers_Class1 & Offers_Class_2

```
CREATE TABLE Offers_Class_2 (
  Start_time TIME(0),
  End_time TIME(0),
  Instructor_name varchar(127),
  Room_number int,
  PRIMARY KEY (Start_time, End_time, Instructor_name)
);

CREATE TABLE Offers_Class_1 (
  Class_ID varchar(32),
  Price int,
  Name varchar(127),
  Start_time TIME(0) NOT NULL,
  End_time TIME(0) NOT NULL,
  Instructor_name varchar(127) NOT NULL,
  Gym_Address varchar(127) NOT NULL,
  PRIMARY KEY (Class_ID),
  FOREIGN KEY (Gym_Address) references Gym(address),
  FOREIGN KEY (Start_time, End_time, Instructor_name)
    references Offers_Class_2(Start_time, End_time, Instructor_name)
);
```

7b) Does_Exercise

```
CREATE TABLE Does_Exercise (
  Number_of_sets int,
  Exercise_Name varchar(127),
  Class_ID varchar(127),
  PRIMARY KEY (Exercise_Name),
  FOREIGN KEY (Class_ID) references Offers_Class_1(Class_ID)
);
```

7c) Has_Equipment

```
CREATE TABLE Has_Equipment (
  Equipment_name varchar(127),
  Manufacturing_ID varchar(127),
  Ownership_status varchar(127),
  Gym_Address varchar(127) NOT NULL,
  PRIMARY KEY (Manufacturing_ID),
  FOREIGN KEY (Gym_Address) references Gym(Address)
);
```

7d) Has_EmergencyContact

```
CREATE TABLE Has_EmergencyContact(
  Email varchar(127),
  Phone_number bigint,
  Name varchar(127),
  Relationship varchar(127),
  PRIMARY KEY (Name, Email),
  FOREIGN KEY (Email) references User(Email)
);
```

7e) Sells_Merchandise

```
CREATE TABLE Sells_Merchandise(
  Gym_Address varchar(127),
  Transaction_ID varchar(127),
  Manufacturing_ID varchar(127),
  Price int,
  PRIMARY KEY (Manufacturing_ID),
  FOREIGN KEY (Gym_Address) references Gym(Address)
);
```

7f) User

```
CREATE TABLE User (
  Name varchar(127),
  Email varchar(127),
  MembershipType varchar(31),
  PRIMARY KEY (Email)
);
```

7g) Gym

```
CREATE TABLE Gym (
  Name varchar(127),
  Opening_time time(0),
  Closing_time time(0),
  Address varchar(127),
  PRIMARY KEY (Address)
);
```

7h) Involves

```
CREATE TABLE Involves(
  Duration varchar(127),
  Exercise_Name varchar(127),
  Equipment_Manufacturing_ID varchar(127),
  PRIMARY KEY (Exercise_Name, Equipment_Manufacturing_ID),
  FOREIGN KEY (Exercise_Name) references Does_Exercise(Exercise_Name),
  FOREIGN KEY (Equipment_Manufacturing_ID)
    references Has_Equipment(Manufacturing_ID)
);
```

7i) BodyPart

```
CREATE TABLE BodyPart (
  Name varchar(127),
  PRIMARY KEY (Name)
 );
```

7j) Targets

```
CREATE TABLE Targets(
  Exercise_Name varchar(127),
  BodyPart_name varchar(127),
  Intensity_rating int,
  PRIMARY KEY (Exercise_Name, BodyPart_name),
  FOREIGN KEY (Exercise_Name) references Does_Exercise(Exercise_Name),
  FOREIGN KEY (BodyPart_name) references BodyPart(Name)
);
```

7k) Uses

```
CREATE TABLE Uses(
  Class_ID varchar(127),
  Manufacturing_ID varchar(127),
  Start_time TIME(0),
  End_time TIME(0),
  PRIMARY KEY (Manufacturing_ID, Class_ID),
  FOREIGN KEY (Manufacturing_ID)
    references Has_Equipment(Manufacturing_ID),
  FOREIGN KEY (Class_ID) references Offers_Class_1(Class_ID)
);
```

7l) Joins

```
CREATE TABLE Joins(
  Join_date DATE,
  User_Email varchar(127),
  Class_ID varchar(127),
  PRIMARY KEY (User_Email, Class_ID),
  FOREIGN KEY (User_Email) references User(Email),
  FOREIGN KEY (Class_ID) references Offers_Class_1(Class_ID)
);
```

7m) Buys

```
CREATE TABLE Buys (
  Payment_method varchar(127),
  Merch_Manufacturing_ID varchar(127),
  User_Email varchar(127),
  PRIMARY KEY (Merch_Manufacturing_ID),
  FOREIGN KEY (Merch_Manufacturing_ID)
    REFERENCES Sells_Merchandise(Manufacturing_ID),
  FOREIGN KEY (User_Email) REFERENCES User(Email)
);
```

7n) TShirt

```
CREATE TABLE TShirt (
  Manufacturing_ID varchar(127),
  Size varchar(127),
  PRIMARY KEY (Manufacturing_ID)
```

```
);
```

7o) ProteinPowder

```
CREATE TABLE ProteinPowder (
  Manufacturing_ID varchar(127),
  Volume int,
  PRIMARY KEY (Manufacturing_ID)
);
```

7p) RegisteredTo

```
CREATE TABLE RegisteredTo(
  Join_date TIME(0),
  User_Email varchar(127),
  Gym_Address varchar(127),
  PRIMARY KEY (User_Email, Gym_Address),
  FOREIGN KEY (User_Email) references User(Email),
  FOREIGN KEY (Gym_Address) references Gym(Address)
);
```

8. Insert Statements

8a) Offers_Class1 & Offers_Class_2

```
INSERT INTO Offers_Class_2 VALUES
  ('5:15', '6:15', 'Andy Liang', 306),
  ('14:00', '15:00', 'Jerry Shao', 021),
  ('5:15', '6:15', 'Jerry Shao', 306),
  ('5:15', '6:30', 'Andy Liang', NULL),
  ('17:00', '17:01', 'Noreen Chan', 013);

INSERT INTO Offers_Class_1 VALUES
  ('1', 35, '1-on-1 Training', '5:15', '6:15', 'Andy Liang', '1234
Anystreet Rd, Richmond, BC'),
  ('2', 25, 'Cardio', '14:00', '15:00', 'Jerry Shao', '3746 Maine St,
Vancouver, BC'),
  ('3', 85, 'Pilates', '5:15', '6:15', 'Jerry Shao', '8686 Burns Rd,
Burnaby, BC'),
  ('4', 95, 'Yoga', '5:15', '6:30', 'Andy Liang', '1234 Anystreet Rd,
Richmond, BC'),
  ('5', 0, 'Cycling', '17:00', '17:01', 'Noreen Chan', '32567 Steveston
Hwy, Richmond, BC');
```

8b) Does_Exercise

```
INSERT INTO Does_Exercise VALUES
  (5, 'Bench Press', '1'),
  (35, 'Push-ups', '2'),
  (35, 'Lunges', '3'),
  (10, 'Squats', '4'),
  (2, 'Cycling', '1');
```

8c) Has_Equipment

```
INSERT INTO Has_Equipment VALUES
  ('Bench', '342', 'Rent', '1234 Anystreet Rd, Richmond, BC'),
  ('Bench', '343', 'Own', '1234 Anystreet Rd, Richmond, BC'),
  ('Bench', '000', NULL, '8686 Burns Rd, Burnaby, BC'),
  (NULL, '824', 'Own', '1234 Anystreet Rd, Richmond, BC'),
  ('Elliptical', '4327', 'Rent', '32567 Steveston Hwy, Richmond, BC');
```

8d) Has_EmergencyContact

```
INSERT INTO Has_EmergencyContact VALUES
  ('rng@gmail.com', 7784949396, 'Jessica Wong', 'Colleague'),
  ('rng@gmail.com', 7780183849, 'Mrs. Ng', 'Wife'),
  ('jeffclune@gmail.com', 6045729296, 'Andreas Lehrmann', 'Colleague'),
  ('norm@ubc.ca', 7784949396, 'Jessica Wong', 'Friend'),
  ('jjim@ubc.ca', 6045828272, 'Jessica Wong', NULL);
```

8e) Sells_Merchandise

```
INSERT INTO Sells_Merchandise VALUES
  ('1234 Anystreet Rd, Richmond, BC', 'T83824', 'M1', 13),
  ('32567 Steveston Hwy, Richmond, BC', 'T83824', 'M2', 13),
  ('1234 Trout Place, Coquitlam, BC', 'T3', 'M3', NULL),
  ('3746 Maine St, Vancouver, BC', 'T5', 'M4', 13),
  ('1234 Anystreet Rd, Richmond, BC', 'T6', 'M5', 13);
```

8f) User

```
INSERT INTO User (Name, Email, MembershipType) VALUES
  ('Raymond Ng', 'rng@gmail.com', 'Basic'),
  ('Jessica Wong', 'jwong@ubc.ca', 'Basic'),
  ('Jeff Clune', 'jeffclune@gmail.com', 'Pro'),
  (NULL, 'jjim@ubc.ca', 'Basic'),
  ('Norm Hutchinson', 'norm@ubc.ca', 'Pro');
```

8g) Gym

```
INSERT INTO Gym VALUES
  ('Anytime Fitness', NULL, NULL, '1234 Anystreet Rd, Richmond, BC'),
  ('Golds', '3:30', '13:30', '3746 Maine St, Vancouver, BC'),
  ('24-Hour Fitness', '9:30', '23:00', '8686 Burns Rd, Burnaby, BC'),
  ('Trout Lake', '12:00', '23:00', '1234 Trout Place, Coquitlam, BC'),
  (NULL, '00:30', '9:30', '32567 Steveston Hwy, Richmond, BC');
```

8h) Involves

```
INSERT INTO Involves VALUES
  ('1 hour', 'Cycling', '4327'),
  ('10 mins', 'Bench Press', '343'),
  (NULL, 'Bench Press', '342'),
  ('15 mins', 'Squats', '824'),
  ('10 mins', 'Lunges', '824');
```

8i) BodyPart

```
INSERT INTO BodyPart VALUES
  ('Chest'),
  ('Arms'),
  ('Legs'),
  ('Back'),
  ('Shoulders');
```

8j) Targets

```
INSERT INTO Targets VALUES
  ('Push-ups', 'Arms', 2),
  ('Push-ups', 'Chest', 2),
  ('Squats', 'Legs', 2),
  ('Bench Press', 'Chest', 2),
  ('Cycling', 'Legs', 2);
```

8k) Uses

```
INSERT INTO Uses VALUES
  ('1', '824', '14:00', '15:00'),
  ('1', '342', '14:30', '15:00'),
  ('3', '000', '11:00', '11:15'),
  ('4', '000', '11:15', '12:00'),
  ('5', '4327', '17:00', '17:01');
```

8l) Joins

```
INSERT INTO Joins VALUES
  ('2023-03-01', 'norm@ubc.ca', '1'),
  ('2022-03-11', 'rng@gmail.com', '4'),
  ('2021-05-09', 'norm@ubc.ca', '3'),
  ('2023-04-01', 'jjim@ubc.ca', '3'),
  ('2023-03-01', 'jeffclune@gmail.com', '2');
```

8m) Buys

```
INSERT INTO Buys VALUES
  ('Cash', 'M1', 'rng@gmail.com'),
  ('Credit Card', 'M2', 'jwong@ubc.ca'),
  ('Cash', 'M3', 'jeffclune@gmail.com'),
  ('Debit Card', 'M4', 'jjim@ubc.ca'),
```

```
  ('Cash', 'M5', 'norm@ubc.ca');
```

8n) TShirt

```
INSERT INTO TShirt VALUES
('1', 'XS'),
('2', 'S'),
('3', 'M'),
('4', 'L'),
('5', 'XL')
```

8o) ProteinPowder

```
INSERT INTO ProteinPowder (Manufacturing_ID, Volume) VALUES
('1', 500),
('2', 300),
('3', 450),
('4', 450),
('5', 450);
```

8p) RegisteredTo

```
INSERT INTO RegisteredTo VALUES
  ('13:00', 'norm@ubc.ca', '1234 Anystreet Rd, Richmond, BC'),
  ('16:00', 'norm@ubc.ca', '8686 Burns Rd, Burnaby, BC'),
  ('10:00', 'rng@gmail.com', '1234 Anystreet Rd, Richmond, BC'),
  ('13:00', 'jjim@ubc.ca', '32567 Steveston Hwy, Richmond, BC'),
  ('21:00', 'jwong@ubc.ca', '1234 Anystreet Rd, Richmond, BC');
```