University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: 2

Date: 02/27/23

Group Number: 34

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Amy Jo	88162888	k4y2b	amyjojjm2001@gmail.com
Sophia Lee	57671240	d7u5t	sophia.ahyl@gmail.com
Bloom Soh Pei Xuan	79461265	f8z3q	bloomspx@gmail.com

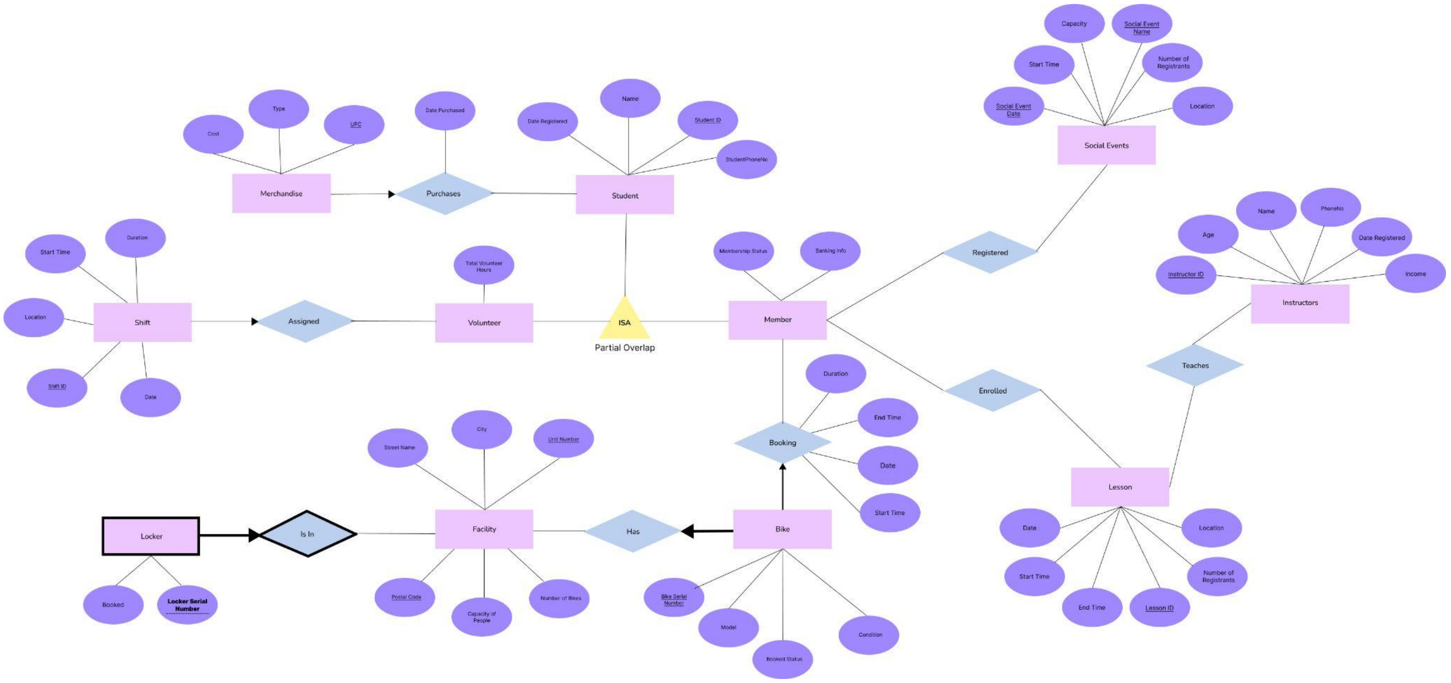
By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address,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

An on campus bike sharing club/program for university students. Students can pay an annual membership fee to become a member which will allow them to book the club's bicycles and facilities. Students can also volunteer at the club while members can attend member-only social events or lessons.

3. ER Diagram



New Changes Implemented:

- Bolded IsIn relationship and added bolded arrow to better illustrate Locker as a weak entity
- Added a new Merchandise entity and a new many-to-one relationship Purchases between Merchandise and Student so that we have 7 relationships (had 6 before)
- Changed the Member-Booking-Bike relationship from many-to-many to one-to-one
- Added PhoneNo attribute to Student entity
- Added PhoneNo attribute to Instructor entity
- Added Duration attribute to Booking
- Replaced Address attribute with street name, postal code, unit number, city in Facility entity
- Added Date attribute to Shift
- Changed the Shift-Assigned-Volunteer relationship from many-to-many to many-to-one

4. The schema derived from your ER diagram (above).

LockerIsIn(
Booked: BOOL,
LockerNumber: INT,
PostalCode: CHAR(6),
UnitNumber: INT)

Volunteer(
TotalVolunteerHours: INT,
DateRegistered: DATE,
Name: CHAR(20),
StudentID: CHAR(8))

Member(
MembershipStatus: CHAR(12),
BankingInfo: CHAR(19),
DateRegistered: DATE,
Name: CHAR(20),
StudentID: CHAR(8))

Student(
DateRegistered: DATE,
StudentName: CHAR(20),
StudentID: CHAR(8),
StudentPhoneNo: INT)

BikeHas(
BikeSerialNumber: INT,
Model: CHAR(20),
BookedStatus: BOOL,
Condition:CHAR(8),
PostalCode:CHAR(6),
UnitNumber: INT)

Booking(
StudentID: CHAR(8),
BikeSerialNumber: INT,
EndTime: INT,
StartTime: INT,
Duration: TIME,
Date: DATE)

SocialEvents(
SocialEventName: CHAR(20),

```
     SocialEventDate: DATE,
     StartTime: INT,
     Capacity: INT,
     NumberOfRegistrants: INT,
     Location: CHAR(20))

Instructors(
     ID: CHAR(8),
     Age: INT,
     Name: CHAR(20),
     DateRegistered: DATE,
     Income: INT)

Facility(
     PostalCode: CHAR(6),
     UnitNumber: CHAR(8),
     StreetName: CHAR(20),
     City: CHAR(20),
     CapacityOfPeople: INT,
     NumberOfBikes: INT)

Shift(
     ShiftID: CHAR(8),
     Date: DATE,
     Duration: INT,
     StartTime: INT,
     Location: CHAR(20))

Lesson(
     Date: DATE,
     StartTime: INT,
     EndTime: INT,
     LessonID: CHAR(8),
     NumberOfRegistrants: INT,
     Location: CHAR(20))

MerchandisePurchased(
     Cost: FLOAT,
     Type:CHAR(20),
     UPC: CHAR(12),
     DatePurchased: DATE,
     StudentId: CHAR(8))

Registered(
      StudentID: CHAR(8),
      SocialEventDate: DATE,
      SocialEventName: CHAR(20))

Enrolled(
      StudentID: CHAR(8),
      LessonID: CHAR(8))

Teaches(
      InstructorID: CHAR(8),
      LessonID: CHAR(8))
```

5. Functional Dependencies (FDs)

LockerIsIn

LockerNumber UnitNumber PostalCode → Booked
LockerNumber UnitNumber PostalCode → City StreetName CapacityOfPeople NumberOfBikes
UnitNumber PostalCode → City StreetName CapacityOfPeople NumberOfBikes

BikeHas

BikeSerialNumber → UnitNumber PostalCode
BikeSerialNumber → Model BookedStatus Condition
UnitNumber PostalCode → City StreetName CapacityOfPeople NumberOfBikes

Booking

StudentId BikeSerialNumber → EndTime Date StartTime Duration
StartTime EndTime → Duration
EndTime Duration → StartTime
StartTime Duration → EndTime

MerchandisePurchased

UPC → Type Cost
UPC → StudentID
UPC → DatePurchased
StudentID → StudentName DateRegistered StudentPhoneNo

Teaches

InstructorID → Age Name DateRegistered Income InstructorPhoneNo
InstructorPhoneNo → InstructorID
LessonID → Date StartTime EndTime Location NumberOfRegistrants

Enrolled

StudentID → StudentName DateRegistered StudentPhoneNo MembershipStatus BankingInfo
StudentPhoneNo → StudentID
StudentPhoneNo → StudentName
LessonID → Date StartTime EndTime Location NumberOfRegistrants

Registered
SocialEventName Date → StartTime Capacity NumberOfRegistrants Location
StudentID → Name DateRegistered StudentPhoneNo MembershipStatus BankingInfo

Assigned
ShiftID → Duration StartTime Location Date
StudentID → Name DateRegistered PhoneNo TotalVolunteerHours

6. Normalization

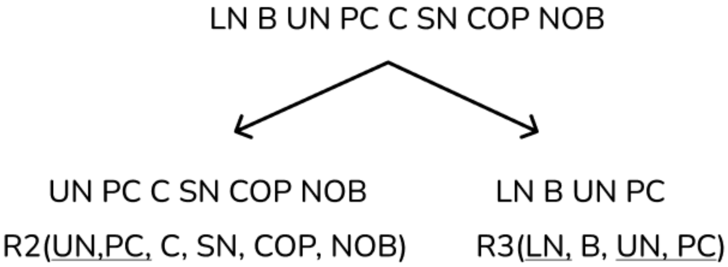
R1: LockerIsIn
LockerNumber UnitNumber PostalCode → Booked
LockerNumber UnitNumber PostalCode → City StreetName CapacityOfPeople NumberOfBikes
UnitNumber PostalCode → City StreetName CapacityOfPeople NumberOfBikes

Primary Key: LockerNumber UnitNumber PostalCode

$\{\text{UnitNumber PostalCode}\}^{+}$ = {UnitNumber PostalCode City StreetName CapacityOfPeople NumberOfBikes}
∴ **LockerIsIn** is NOT in BCNF form since for UnitNumber PostalCode → City StreetName CapacityOfPeople NumberOfBikes, UnitNumber PostalCode is not the superkey

We can simplify the attributes as follows and perform decomposition:

```
UN PC → C SN COP NOB
LN UN PC → B C SN COP NOB
```



```
R2(
     UnitNumber: INT,
     PostalCode: CHAR(6),
     City: CHAR(20),
     StreetName: CHAR(20),
     CapacityOfPeople: INT,
     NumberOfBikes: INT)
```

Facility

```
R3(
      LockerNumber: INT,
      Booked: BOOLEAN,
      UnitNumber: INT,
      PostalCode: CHAR(6))
```

LockerIsIn

R4: BikeHas

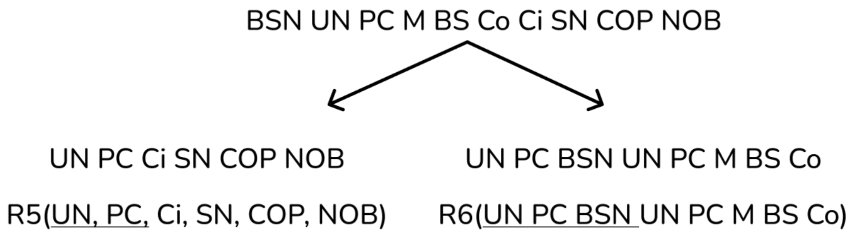
BikeSerialNumber → UnitNumber PostalCode
BikeSerialNumber → Model BookedStatus Condition
UnitNumber PostalCode → City StreetName CapacityOfPeople NumberOfBikes

Primary Key: BikeSerialNumber

{BikeSerialNumber UnitNumber PostalCode }$^{+}$ = {BikeSerialNumber UnitNumber PostalCode Model BookedStatus Condition City StreetName CapacityOfPeople NumberOfBikes}
∴ **BikeHas** is NOT in BCNF form because for BikeSerialNumber → UnitNumber PostalCode, BikeSerialNumber is not a superkey

We can simplify the attributes as follows and perform decomposition:

```
UN PC  → Ci SN COP NOB
BSN  → UN PC M BS Co
```


BSN UN PC M BS Co Ci SN COP NOB
UN PC Ci SN COP NOB
UN PC BSN UN PC M BS Co
R5(UN, PC, Ci, SN, COP, NOB)
R6(UN PC BSN UN PC M BS Co)

```
R5(
      PostalCode: CHAR(6),
      UnitNumber: CHAR(8),
      City: CHAR(20),
      StreetName: CHAR(20),
      CapacityOfPeople: INT,
      NumberOfBikes: INT)
```

Facility

```
R6(
     PostalCode: CHAR(6),
     UnitNumber: CHAR(8),
     BikeSerialNumber: INT,
     Model: CHAR(20),
     BookedStatus: BOOL,
     Condition:CHAR(8)
     )
```

BikeHas

While the BCNF decomposition requires us to split BikeHas into 2 Relations - Facility and BikeHas, given that R5 is an exact copy of R2 (both are Facility relations), we will take R5 as a duplicate relation and will not be creating 2 duplicate Facility tables subsequently.

R7: Booking

StudentId BikeSerialNumber → EndTime Date StartTime Duration
StartTime EndTime → Duration
EndTime Duration → StartTime
StartTime Duration → EndTime

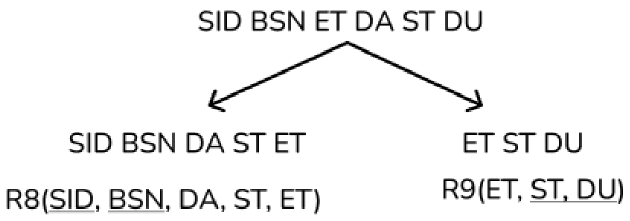
Primary Key: StudentId, BikeSerialNumber

$\{\text{StudentId BikeSerialNumber}\}^{+}$ → {StudentId BikeSerialNumber EndTime Date StartTime Duration}

∴ **Booking** is NOT in BCNF form because for StartTime EndTime → Duration, StartTime and Duration is not a superkey

We can simplify the attributes as follows and perform decomposition:

```
SID BSN → ET DA ST DU
ET DU →  ST
ST DU →  ET
```



```
R8(
      StudentID: CHAR(8),
      BikeSerialNumber: INT,
      EndTime: INT,
      StartTime: INT,
      Date: DATE)
```

BikeBooked

```
R9(
      EndTime: INT,
      StartTime: INT,
      Duration: TIME)
```

Booking

R10: MerchandisePurchased

StudentID → StudentName DateRegistered StudentPhoneNo
UPC → Type Cost
UPC → StudentID
UPC → DatePurchased

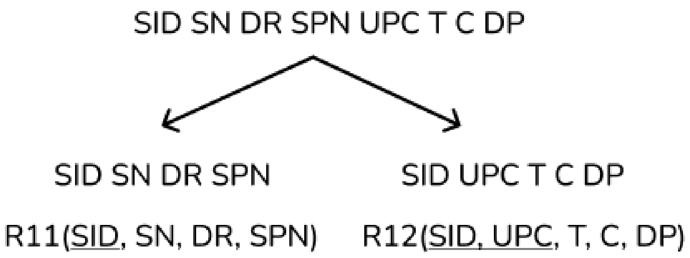
Primary Key: UPC

$\{StudentID\}^+$ = {StudentID StudentName DateRegistered StudentPhoneNo}

∴ StudentID is NOT in BCNF form since StudentID StudentName DateRegistered StudentPhoneNo is not a superkey

We can simplify the attributes as follows and perform decomposition:

```
SID → SN DR SPN
UPC → T C SID DP
```



```
R11(
      StudentID: CHAR(8)
      StudentName: CHAR(20),
      DateRegistered: DATE,
      StudentPhoneNo: INT)
```

Student

```
R12(
      StudentID: CHAR(8),
      UPC: CHAR(12),
      Type:CHAR(20),
      Cost: FLOAT,
      DatePurchased: DATE)
```

MerchandisePurchased

R13: Teaches

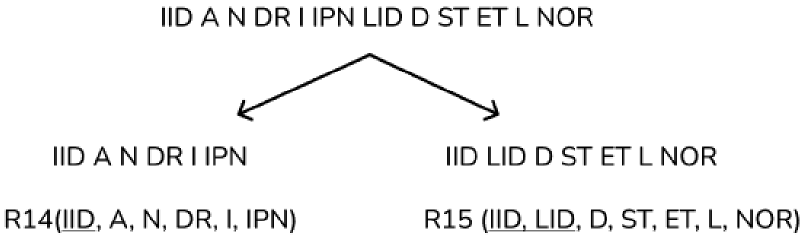
InstructorID → Age Name DateRegistered Income InstructorPhoneNo
InstructorPhoneNo → InstructorID
LessonID → Date StartTime EndTime Location NumberOfRegistrants LessonID

Primary Key: InstructorID, LessonID

{InstructorID}$^+$ = {InstructorID Age Name DateRegistered Income InstructorPhoneNo}
∴ **Teaches** is NOT in BCNF form because for InstructorPhoneNo → InstructorID, InstructorPhoneNo is not a superkey

We can simply the attributes as follows and perform decomposition:

```
IID → A N DR I IPN
IPN → IID
LID → D ST ET L NOR
```



IID A N DR I IPN LID D ST ET L NOR

IID A N DR I IPN

IID LID D ST ET L NOR

R14(IID, A, N, DR, I, IPN)

R15 (IID, LID, D, ST, ET, L, NOR)

```
R14(
      InstructorID: CHAR(8),
      Age: INT,
      Name: CHAR(20),
      DateRegistered: DATE,
      Income: INT,
      InstructorPhoneNo: INT)
```

Instructor

```
R15(
      InstructorID: CHAR(8),
      LessonID: CHAR(8)
      Date: DATE,
      StartTime: INT,
      EndTime: INT,
      Location: CHAR(20),
      NumberOfRegistrants: INT)
```

Teaches

R16: Registered

SocialEventName Date → StartTime Capacity NumberOfRegistrants Location
StudentID → StudentName DateRegistered StudentPhoneNo MembershipStatus BankingInfo

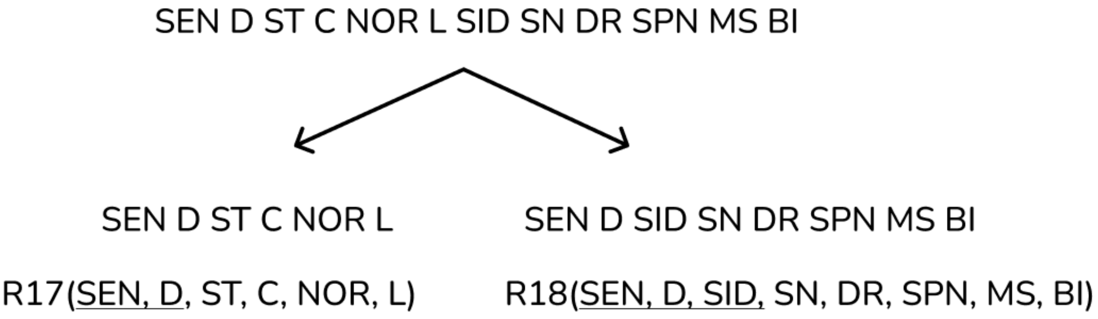
Primary Key: SocialEventName, Date, StudentID

$\{\text{SocialEventName Date}\}^{+}$ = { SocialEventName Date StartTime Capacity NumberOfRegistrants Location}

∴ **SocialEventName, Date, StudentID** is not in BCNF because for StudentID → Name DateRegistered StudentPhoneNo MembershipStatus BankingInfo, StudentID is not a superkey

We can simply the attributes as follows and perform decomposition:

```
SEN D → ST C NOR L
SID → SN DR SPN MS BI
```


SEN D ST C NOR L SID SN DR SPN MS BI
SEN D ST C NOR L
SEN D SID SN DR SPN MS BI
R17(SEN, D, ST, C, NOR, L)
R18(SEN, D, SID, SN, DR, SPN, MS, BI)

```
R17(
      SocialEventName: CHAR(20),
      Date: DATE,
      StartTime: INT,
      Capacity: INT,
      NumberOfRegistrants: INT,
      Location: CHAR(20))
```

SocialEvent

```
R18(
     SocialEventName: CHAR(20),
     Date: DATE,
     StudentID: CHAR(8)
     StudentName: CHAR(20),
     DateRegistered: DATE,
     StudentPhoneNo: INT,
     MembershipStatus: CHAR(12),
     BankingInfo: CHAR(19))
```

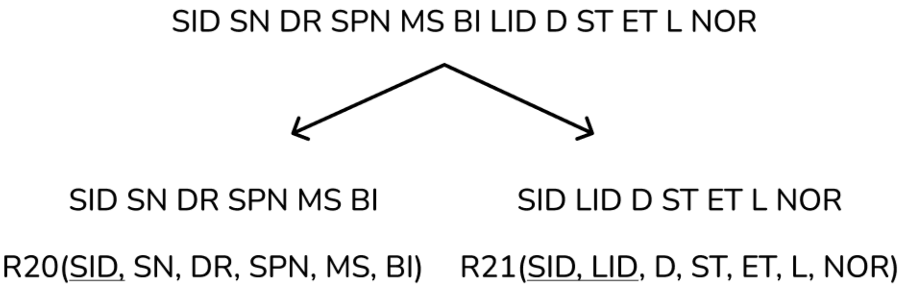
Registered

R19: Enrolled
StudentID → StudentName DateRegistered StudentPhoneNo MembershipStatus BankingInfo
StudentPhoneNo → StudentID
StudentPhoneNo → StudentName
LessonID → Date StartTime EndTime Location NumberOfRegistrants

Primary Key: StudentID, LessonID
$\{StudentID\}^{+}$ = {StudentID StudentName DateRegistered StudentPhoneNo MembershipStatus BankingInfo}
∴ **Enrolled** is NOT in BCNF form because StudentID → StudentName DateRegistered StudentPhoneNo MembershipStatus BankingInfo is not a superkey

We can simply the attributes as follows and perform decomposition:

```
SID → SN DR SPN MS BI
SPN → SID
SPH → SN
LID → D ST ET L NOR
```



R20(SID, SN, DR, SPN, MS, BI) R21(SID, LID, D, ST, ET, L, NOR)

```
R20(
     StudentID: CHAR(8)
     StudentName: CHAR(20),
     DateRegistered: DATE,
     StudentPhoneNo: INT,
     MembershipStatus: CHAR(12),
     BankingInfo: CHAR(19))
```

Member

```
R21(
     StudentID: CHAR(8),
     LessonID: CHAR(8),
     Date: DATE,
     StartTime: INT,
     EndTime: INT,
     Location: CHAR(20),
     NumberOfRegistrants: INT)
```

Enrolled

R22: Assigned

StudentID → StudentName DateRegistered PhoneNo TotalVolunteerHours
ShiftID → Duration StartTime Location Date

Primary Key: StudentID, ShiftID

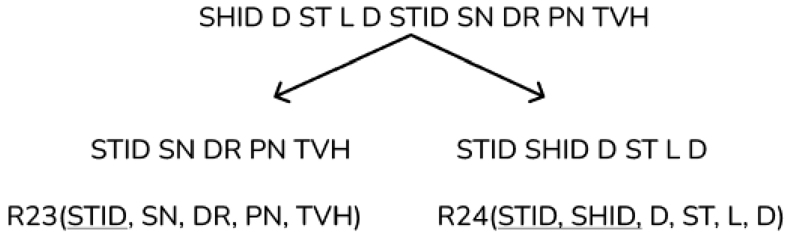
$\{StudentID\}^+$ = {StudentID StudentName DateRegistered PhoneNo TotalVolunteerHours }

∴ Assigned is NOT in BCNF form because for StudentID → StudentName DateRegistered PhoneNo TotalVolunteerHours, StudentID is not a superkey

We can simply the attributes as follows and perform decomposition:

```
STID → SN DR PN TVH
```

```
SHID →  D ST L D
```


SHID D ST L D STID SN DR PN TVH
STID SN DR PN TVH
STID SHID D ST L D
R23(STID, SN, DR, PN, TVH)
R24(STID, SHID, D, ST, L, D)

```
R23(
     StudentID: CHAR(8),
     StudentName: CHAR(20),
     DateRegistered: DATE,
     StudentPhoneNo: INT,
     TotalVolunteerHours: INT)
```

Volunteer

```
R24(
     StudentID: CHAR(8),
     ShiftID: CHAR(8),
     Duration: INT,
     StartTime: INT,
     Location: CHAR(20),
     Date: DATE)
```

Shift

7. The SQL DDL statements required to create all the tables from item #6. The statements should use the appropriate foreign keys, primary keys, UNIQUE constraints, etc.

```
CREATE TABLE LockersIn(
     Booked NOT NULL BOOLEAN,
     LockerNumber INT,
     PostalCode CHAR(6),
     UnitNumber INT,
     PRIMARY KEY (LockerNumber, PostalCode, UnitNumber),
     FOREIGN KEY (PostalCode, UnitNumber) REFERENCES Facility
          ON DELETE CASCADE
          ON UPDATE CASCADE
)

CREATE TABLE BikeHas(
     BikeSerialNumber INT,
     Model CHAR(20),
     BookedStatus NOT NULL BOOLEAN,
     Condition CHAR(8),
     PostalCode CHAR(6),
     UnitNumber INT,
     PRIMARY KEY (BikeSerialNumber),
     FOREIGN KEY (PostalCode, UnitNumber) REFERENCES Facility
)


CREATE TABLE Booking(
     StudentID CHAR(8),
     BikeSerialNumber INT,
     StartTime INT,
     EndTime INT,
     Date DATE,
     PRIMARY KEY (StudentID, BikeSerialNumber),
     FOREIGN KEY (StudentID) REFERENCES Member,
     FOREIGN KEY (BikeSerialNumber) REFERENCES BikeHas
          ON DELETE CASCADE
          ON UPDATE CASCADE
)


CREATE TABLE Duration(
     EndTime INT,
     StartTime INT,
     Duration TIME,
     PRIMARY KEY (StartTime, Duration)
)

CREATE TABLE MerchandisePurchased(
     Cost FLOAT,
     Type CHAR(20),
     UPC CHAR(12),
     DatePurchased DATE,
     StudentID CHAR(8),
     PRIMARY KEY (UPC),
     FOREIGN KEY (StudentID) REFERENCES Student
)

CREATE TABLE Instructor(
     InstructorID CHAR(8),
     Age INT,
     Name CHAR(20),
     DateRegistered DATE,
     Income INT,
     InstructorPhoneNo INT,
     PRIMARY KEY (InstructorID)
)

CREATE TABLE Teaches(
     InstructorID CHAR(8),
     LessonID CHAR(8))
     Date DATE,
     StartTime INT,
     EndTime INT,
     Location CHAR(20),
     NumberOfRegistrants INT,
     PRIMARY KEY (InstructorID, LessonID),
     FOREIGN KEY (InstructorID) REFERENCES Instructor,
          ON DELETE CASCADE
          ON UPDATE CASCADE
     FOREIGN KEY (LessonID) REFERENCES Lesson
          ON DELETE CASCADE
          ON UPDATE CASCADE
)

CREATE TABLE Member(
     StudentID CHAR(8)
     StudentName CHAR(20),
     DateRegistered DATE,
     StudentPhoneNo INT,
     MembershipStatus CHAR(12),
     BankingInfo CHAR(19),
     PRIMARY KEY (StudentID),
     )

CREATE TABLE Enrolled(
     StudentID CHAR(8),
     LessonID CHAR(8),
     Date DATE,
     StartTime INT,
     EndTime INT,
     Location CHAR(20),
     NumberOfRegistrants INT,
     PRIMARY KEY (StudentID, LessonID),
     FOREIGN KEY (LessonID) REFERENCES Lesson,
          ON DELETE CASCADE
          ON UPDATE CASCADE
     )

CREATE TABLE SocialEvents(
     SocialEventName CHAR(20),
     Date DATE,
     StartTime INT,
     Capacity INT,
     NumberOfRegistrants INT,
     Location CHAR(20),
     PRIMARY KEY (SocialEventName, Date)
     )

CREATE TABLE Registered(
     SocialEventName CHAR(20),
     Date DATE,
     StudentID CHAR(8),
     StudentName CHAR(20),
     DateRegistered DATE,
     StudentPhoneNo INT,
     MembershipStatus CHAR(12),
     BankingInfo CHAR(19),
     PRIMARY KEY (SocialEventName, Date),
     FOREIGN KEY (SocialEventName) REFERENCES SocialEvents,
     FOREIGN KEY (StudentID) REFERENCES Member,
     )

CREATE TABLE Volunteer(
     StudentID CHAR(8),
     StudentName CHAR(8),
     DateRegistered DATE,
     StudentPhoneNo INT,
     TotalVolunteerHours INT,
     PRIMARY KEY (StudentID),
     FOREIGN KEY (StudentID) REFERENCES Student,
     )

CREATE TABLE Shift(
     StudentID CHAR(8),
     ShiftID CHAR(8),
     Duration INT,
     StartTime INT,
     Location CHAR(20),
     Date DATE,
     PRIMARY KEY (ShiftID),
     FOREIGN KEY (StudentID) REFERENCES Volunteer,
     )

CREATE TABLE Student(
     DateRegistered DATE,
     StudentName CHAR(20),
     StudentID CHAR(8),
     StudentPhoneNo UNIQUE INT,
     PRIMARY KEY (StudentID),
     )

CREATE TABLE Lesson(
     LessonID CHAR(8),
     Date DATE,
     StartTime INT,
     EndTime INT,
     NumberOfRegistrants INT,
     Location CHAR(20),
     PRIMARY KEY (LessonID),
     )

CREATE TABLE Facility(
     UnitNumber INT,
     PostalCode CHAR(6),
     City CHAR(20),
     StreetName CHAR(20),
     CapacityOfPeople INT,
     NumberOfBikes INT,
     PRIMARY KEY (UnitNumber, PostalCode)
     )
```

8. INSERT statements to populate each table with at least 5 tuples.

```
INSERT INTO LockersIn
VALUES (True, 00000000, V2W4T5, 0),
(True, 00000001, V6Y7K4, 1),
(False, 00000002, V4K7N3, 2),
(True, 00000003, V8J3N5, 3),
(False, 00000004, V9N2L2, 4),
(False, 00000005, V7N3K6, 5)
```

```
INSERT INTO BikeHas
VALUES (11111111, Basso, Good, V2W4T5, 0),
(11111112, Electra, Superb, V9N2L2, 4),
(11111113, Giordano, Good, V6Y7K4, 1),
(11111114, Kuota, Bad, V4K7N3, 2),
(11111115, Nicolai, Repair, V2W4T5, 6)

INSERT INTO Booking
VALUES (00000000, 11111111, 1230, 130, 2023-03-01),
(00000000, 11111112, 1230, 130, 2023-04-01),
(00000001, 11111113, 1930, 2130, 2023-05-01),
(00000000, 11111114, 1630, 1730, 2022-06-01),
(00000003, 11111115, 1130, 130, 2023-05-11)

INSERT INTO Duration
VALUES (1230, 130, 60),
(1100, 1130, 30),
(1500, 1630, 90),
(1000, 1100, 60),
(1400, 1600, 120)

INSERT INTO MerchandisePurchased
VALUES (10.30, ‘Helmet’, 123456789012, 2023-02-28, 00000000),
(112.45, ‘Bike’, 123456789011, 2023-02-28, 00000000),
(10.35, ‘Helmet’, 123456789013, 2023-01-28, 00000001),
(400.40’, ‘Bike’, 123456789014, 2022-04-20, 00000002),
(20.45, ‘Helmet’, 123456789015, 2020-06-06, 00000002)

INSERT INTO Instructor
VALUES (55555555, 24, ‘Laila’, 2022-05-03, 45000, 7781111111),
(55555551, 25, ‘Sheila’, 2022-05-06, 55000, 7782222222),
(55555552, 21, ‘Joe’, 2022-05-12, 65000, 7783333333),
(55555553, 34, ‘Bob’, 2020-06-07, 30000, 7784444444),
(55555554, 32, ‘Carson’, 2021-08-12, 60000, 7785555555),

INSERT INTO Teaches
VALUES (55555555, 44444444, 2022-05-03, 1230, 1330, Dance St, 12),
(55555551, 44444441, 2023-04-22, 0140, 0340, Cron St, 30),
(55555552, 44444442, 2022-06-03, 0930, 1030, Bumble Rd, 11),
(55555553, 44444443, 2023-05-30, 1500, 1600, Chocolate St, 45),
(55555554, 44444444, 2023-10-13, 1215, 1315, Renfrew Ave, 32)

INSERT INTO Member
VALUES
```

```
(00000000, ‘Bella’, 2021-05-04, 6041234567, Gold, 1234567890123456789),
(00000001, ‘Johnny’, 2023-01-04, 6047654321, Gold, 1234567890123456781),
(00000002, ‘Maple’, 2021-05-04, 6041111111, Silver, 1234567890123456782),
(00000003, ‘Jennie’, 2020-06-013, 6042222222, Bronze,
1234567890123456783),
(00000004, ‘Han’, 2021-05-04, 6043333333, Gold, 1234567890123456784),

INSERT INTO Enrolled
VALUES (00000000, 44444444, 2022-05-03, 1230, 1330, Dance St, 12),
(00000001, 44444444, 2022-05-03, 1230, 1330, Dance St, 12),
(00000002, 44444444, 2022-05-03, 1230, 1330, Dance St, 12),
(00000003, 44444444, 2022-05-03, 1230, 1330, Dance St, 12),
(00000004, 44444444, 2022-05-03, 1230, 1330, Dance St, 12)

INSERT INTO SocialEvents
VALUES (‘HappyLand Day’, 2023-04-05, 1230, 15, 5, Hastings Ave),
(‘Bike Day’, 2023-04-15, 1330, 25, 4, Hastings Ave),
(‘Wonderworld’, 2023-04-25, 1230, 35, 20, Bern St),
(‘Practicing biking’, 2023-05-05, 1230, 20, 9, Hastings Ave),
(‘Watch and learn’, 2023-05-15, 1130, 30, 12, Bern St)

INSERT INTO Registered
VALUES (‘HappyLand Day’, 2023-04-05, 00000000, ‘Bella’, 6041234567, Gold,
1234567890123456789),
(‘HappyLand Day’, 2023-04-05, 00000001, ‘Johnny’, 6047654321, Gold,
1234567890123456781),
(‘HappyLand Day’, 2023-04-05, 00000002, ‘Maple’, 6041111111, Silver,
1234567890123456782),
(‘HappyLand Day’, 2023-04-05, 00000003, ‘Jennie’, 6042222222, Bronze,
1234567890123456783),
(‘HappyLand Day’, 2023-04-05, 00000004, ‘Han’, 6043333333, Gold,
1234567890123456784)

INSERT INTO Volunteer
VALUES (00000000, ‘Bella’, 2021-05-04, 6041234567, 120),
(00000002, ‘Maple’, 2021-05-04, 6041111111, 100),
(00000003, ‘Jennie’, 2020-06-13, 6042222222, 320),
(00000005, ‘Joelle’, 2019-12-01, 6044444444, 400),
(00000001, ‘Johnny’, 2023-01-04, 6047654321, 120)

INSERT INTO Shift
VALUES (00000000, 33333333, 60, 1230, Hastings Ave, 2023-03-04),
(00000001, 33333331, 60, 1130, Henria St, 2023-03-05),
(00000002, 33333332, 120, 1030, Honeysuckle Ave, 2023-03-06),
(00000003, 33333334, 360, 1600, Hastings Ave, 2023-03-07),
(00000004, 33333335, 60, 1500, Bern St, 2023-03-08)

INSERT INTO Student
VALUES (2021-05-04, ‘Bella’, 00000000, 6041234567),
(2023-01-04, ‘Johnny’, 00000001, 6047654321),
(2021-05-04, ‘Maple’, 00000002, 6041111111),
(2020-06-13, ‘Jennie’, 00000003, 6042222222),
(2015-11-21, ‘Han’, 00000004, 6043333333),
(2019-12-01, ‘Joelle’, 00000005, 6044444444)

INSERT INTO Lesson
VALUES (44444444, 2022-05-03, 1230, 1330, 12, Dance St),
(44444441, 2023-04-22, 0140, 0340, 30, Bella St),
(44444442, 2022-06-03, 0930, 1030, 11, Bumble Rd),
(44444443, 2023-05-30, 1500, 1600, 45, Chocolate St),
(44444445, 2023-10-13, 1215, 1315, 32, Renfrew Ave)

INSERT INTO Facility
VALUES (0, V2W4T5, Burnaby, Hastings Ave, 100, 12),
(1, V6Y7K4, Vancouver, Renford St, 80, 47),
(2, V4K7N3, Richmond, Bern St, 40, 7),
(3, V8J3N5, Coquitlam, Honeysuckle Avenue, 28),
(4, V9N2L2, Vancouver, Henria St, 5, 12),
(5, V7N3K6, Vancouver, Camelia Ave, 20, 8),
(6, V2W4T5, Vancouver, Bumblebee St, 8, 0)
```