

CPSC 304 Project Cover Page

Milestone #: 2

Date: March 4, 2021

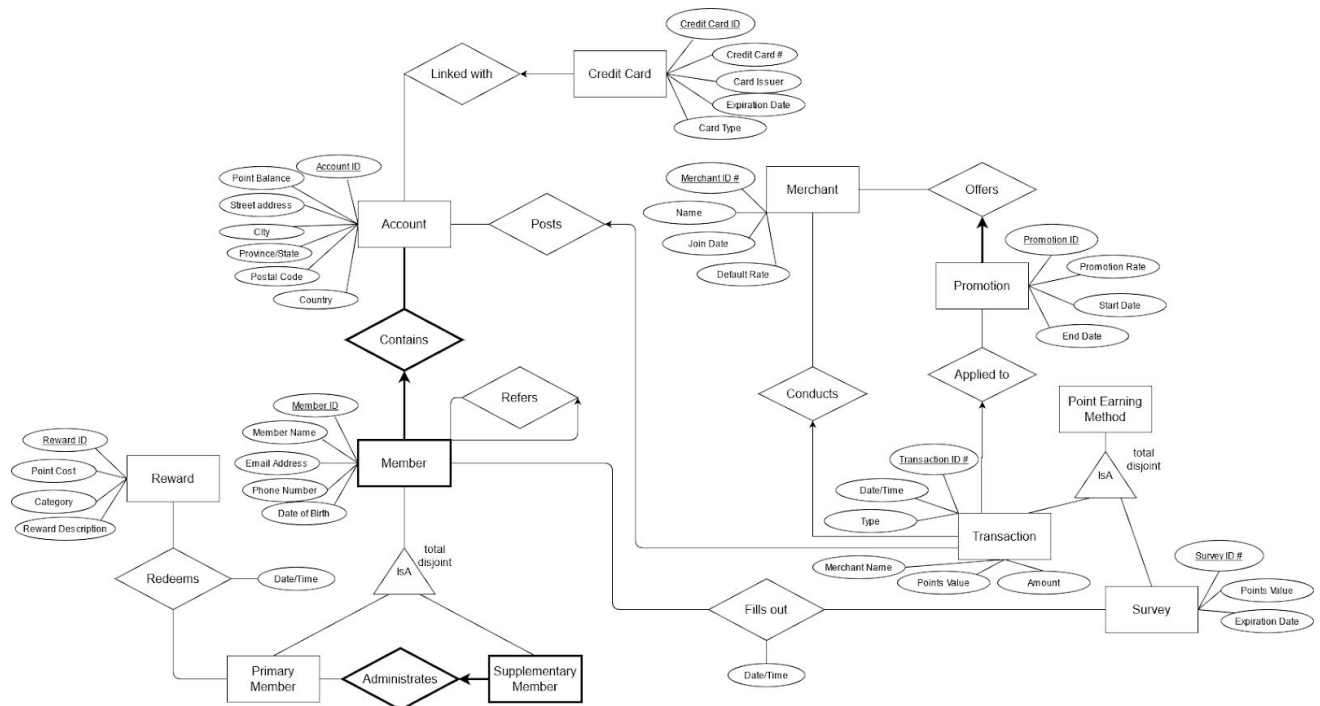
Group Number: 45

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Nazish Tazeem	45548682	h5a2b	nazish@student.ubc.ca
Austin Lee	82785106	h2s8	wjaustinlee@gmail.com
Jeffrey Kwok	32713125	g6m8	kwokjeff@outlook.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



Notes & changes from Milestone 1:

- General
 - Tables are renamed to UpperCamelCase
 - Tables that are in a M:1 relation (but not weak entities) are renamed to Table_Relation naming convention
 - Attributes are renamed to lowerCamelCase
 - Some attributes have been relabelled vs. in the ER diagram for clarity
- CreditCard
 - Credit Card # is no longer the primary key to create a non-trivial FD
 - added creditCardID as the new primary key
- Account
 - Shipping address has been replaced with streetAddress, city, provinceState, postalCode, and country to create a non-trivial FD
- Transaction
 - merchantName added as attribute to account for cases where we insert transactions with merchants who are not in our database

3) Schema

***NOTE: underline means primary key, **bold** means foreign key, any additional notes and constraints are added at the end of the each schema

Account(
 accountID:string,
 pointBalance:integer,
 streetAddress:string,
 city:string,
 postalCode:string,
 provinceState:string,
 country:string)

Additional constraints:

- pointBalance is not null

CreditCard_Linked(
 creditCardID:string,
 creditCardNum:string,
 accountID:integer,
 cardIssuer:string,
 cardType:string,
 expirationDate:date)

FKs:

- accountID references Account

Additional constraints:

- on update and on delete accountID, cascade
- creditCardNum is not null and unique

Member(
 memberID:string,
 accountID:string,
 memberName:string,
 emailAddress:string,
 phoneNumber:string,
 birthDate:date,
 referrerID:integer)

FKs:

- accountID references Account
- referrerID references Member

Candidate keys: emailAddress

Additional constraints:

- emailAddress is not null and unique (as a CK)

- phoneNumber is not null
- on update accountID, cascade

NOTE: PrimaryMember subclass and Member superclass share the same table since PrimaryMember has no unique attributes.

SupplementaryMember(memberID:string,
 accountID:string,
 primaryMemberID:string)

FKs:

- memberID, accountID, primaryMemberID references Member

Merchant(merchantID: string,
 merchantName: string,
 joinDate: date,
 defaultRate: decimal)

Additional constraints:

- merchantName is not null
- defaultRate is not null and non-negative

NOTE: decimal is a numerical data type that stores decimal values

Promotion_Offers(promotionID:string,
 merchantID:string,
 promotionRate:decimal,
 startDate:date,
 endDate:date)

Additional constraints:

- promotionRate is not null and non-negative
- if startDate and endDate are both not null, then endDate must not be before startDate

FKs:

- merchantID references Merchant

Reward(rewardID:string,
 pointCost:integer,
 rewardCategory:string,

rewardDescription:string)

Additional constraints:

- pointCost is not null
- rewardDescription is not null

Redeems(
 rewardID:string,
 accountID:string,
 memberID:string,
 dateTime:datetime)

FKs:

- rewardID references Reward
- accountID, memberID references PrimaryMember

Additional constraints:

- dateTime is not null

NOTE: this table is for the PrimaryMember-Reward M:N relation

NOTE2: datetime is data type that stores both date and time

Survey(
 surveyID:string,
 pointsValue:integer,
 expirationDate:date)

Additional constraints:

- pointsValue is not null

FillsOut(
 accountID:string,
 memberID:string,
 surveyID:string,
 dateTime:datetime)

FKs:

- surveyID references Survey
- accountID, memberID references PrimaryMember

Additional constraints:

- dateTime is not null

NOTE: this table is for the Member-Survey M:N relation

Transaction(
 transactionID:string,
 promotionID:string,
 merchantID:string

```
merchantName:string,  
accountID:string,  
dateTime:datetime,  
type:string,  
pointsValue:integer,  
transactionAmount:decimal)
```

FKs:

- promotionID references Promotion_Offers
- merchantID references Merchant
- accountID references Account

Additional constraints:

- accountID is not null
- dateTime is not null
- merchantName is not null
- type is not null
- pointsValue is not null
- transactionAmount is not null

NOTE: Transaction does not follow the usual Table_Relation naming convention because it is in M:1 relation with several other entities

4) Functional dependencies (FDs)

NOTE: trivial FDs not included

Account

accountID -> pointBalance, streetAddress, city,
postalCode, provinceState, country)
country, postalCode -> provinceState

CreditCard_Linked

creditCardID -> creditCardNum, accountID, cardIssuer,
cardType, expirationDate
creditCardNum -> cardType, cardIssuer

Member

accountID, memberID -> memberName, emailAddress,
phoneNumber, birthDate, referrerID
emailAddress -> accountID, memberID, memberName,
phoneNumber, birthDate

SupplementaryMember

no non-trivial FDs in this table

Merchant

merchantID -> name, joinDate, defaultRate

Promotion_Offers

promotionID -> promotionRate, startDate, endDate

Reward

rewardID -> pointCost, rewardCategory,
rewardDescription

Redeems

rewardID, accountID, memberID -> dateTime

Survey

surveyID -> pointsValue, expirationDate

FillsOut

surveyID, accountID, memberID -> dateTime

Transaction

```
transactionID -> promotionID, merchantID, accountID,  
dateTime, type, pointsValue, transactionAmount,  
merchantName
```


5) Normalization

NOTE: Any tables already in 3NF/BCNF form from (4) are not listed here for brevity

a)

Account

```
    accountID -> pointBalance, streetAddress, city,  
    postalCode, provinceState, country)  
    country, postalCode -> provinceState
```

FD rule (country, postalCode -> provinceState) violates BCNF since {country, postalCode} is not a minimal key so decompose.

BCNF decomposition:

```
Account1(    accountID:integer,  
            pointBalance:integer,  
            streetAddress:string,  
            city:string,  
            postalCode:string,  
            country:string)
```

```
Account2(    country:string,  
            postalCode:string,  
            provinceState:string)
```

Account1 and Account2 meet all criteria for BCNF; no further decomposition

b)

CreditCard_Linked

creditCardID -> creditCardNum, accountID, cardIssuer,
cardType, expirationDate

creditCardNum -> cardType, cardIssuer

FD rule (creditCardNum -> cardType, cardIssuer) violates BCNF
since {creditCardNum} is not a minimal key so decompose.

BCNF decomposition:

CreditCard_Linked1(creditCardID:integer,
creditCardNum:string,
accountID:integer,
expirationDate:date)

CreditCard_Linked2(creditCardNum:string,
cardIssuer:string,
cardType:string)

CreditCard_Linked1 and CreditCard_Linked2 meet all criteria for
BCNF; no further decomposition

6) SQL DDL

```
CREATE TABLE Account1(
    accountID          CHAR(10)  PRIMARY KEY,
    pointBalance       INTEGER    NOT NULL,
    streetAddress      CHAR(100),
    city               CHAR(50),
    postalCode         CHAR(10),
    country            CHAR(50)
);

CREATE TABLE Account2(
    postalCode         CHAR(10),
    country            CHAR(50),
    provinceState     CHAR(50),
    PRIMARY KEY(postalCode, country)
);

CREATE TABLE CreditCard1(
    creditCardID       CHAR(10)  PRIMARY KEY
    creditCardNum      CHAR(16)  NOT NULL UNIQUE,
    accountID          CHAR(10)  NOT NULL,
    expirationDate     DATE,
    FOREIGN KEY(accountID) REFERENCES Account(accountID)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);

CREATE TABLE CreditCard2(
    creditCardNum      CHAR(16)  PRIMARY KEY,
    cardType           CHAR(30),
    cardIssuer         CHAR(30)
);

CREATE TABLE Member(
    memberID          CHAR(10),
    accountID         CHAR(10),
    memberName        CHAR(50),
    emailAddress      CHAR(50)  UNIQUE NOT NULL,
    phoneNumber       Char(20)  NOT NULL,
    birthDate         DATE,
```

```
        referrerID      CHAR(10),
        PRIMARY KEY(memberID, accountID),
        FOREIGN KEY(accountID) REFERENCES Account(accountID)
            ON UPDATE CASCADE
        FOREIGN KEY(referrerID) REFERENCES Member(memberID)
    );

CREATE TABLE SupplementaryMember(
    memberID            CHAR(10),
    accountID           CHAR(10),
    primaryMemberID     CHAR(10),
    PRIMARY KEY (memberID, accountID, primaryMemberID),
    FOREIGN KEY (memberID, accountID, primaryMemberID)
        REFERENCES Member(memberID, accountID, memberID)
);

CREATE TABLE Merchant(
    merchantID          CHAR(10)  PRIMARY KEY,
    merchantName        CHAR(30)  NOT NULL,
    joinDate            DATE,
    defaultRate         DECIMAL(6,2)  NOT NULL
);

CREATE TABLE Promotion_Offers(
    promotionID         CHAR(10)  PRIMARY KEY,
    merchantID          CHAR(10),
    promotionRate       DECIMAL(6,2)  NOT NULL,
    startDate           DATE,
    endDate             DATE,
    FOREIGN KEY(MerchantID) REFERENCES Merchant(merchantID)
);

CREATE TABLE Reward(
    rewardID            CHAR(10)  PRIMARY KEY,
    pointCost           INTEGER  NOT NULL,
    rewardCategory       CHAR(30),
    rewardDescription    CHAR(100) NOT NULL
);

CREATE TABLE Redeems(
```

```
        rewardID          CHAR(10),
        accountID         CHAR(10),
        memberID          CHAR(10),
        dateTime          DATETIME NOT NULL,
        PRIMARY KEY(rewardID, accountID, memberID),
        FOREIGN KEY(rewardID) REFERENCES Reward(rewardID),
        FOREIGN KEY (accountID, memberID) REFERENCES
            Member(accountID, memberID)
    );

CREATE TABLE Survey(
    surveyID              CHAR(10)  PRIMARY KEY,
    pointsValue           INTEGER   NOT NULL,
    expirationDate        DATE
);

CREATE TABLE FillsOut(
    accountID             CHAR(10),
    memberID              CHAR(10),
    surveyID              CHAR(10),
    dateTime              DATETIME NOT NULL,
    PRIMARY KEY(accountID, memberID, surveyID),
    FOREIGN KEY(memberID, accountID) REFERENCES Member(memberID,
accountID),
    FOREIGN KEY(surveyID) REFERENCES Survey(surveyID)
);

CREATE TABLE Transaction(
    transactionID          CHAR(10), PRIMARY KEY,
    promotionID            CHAR(10),
    merchantID             CHAR(10),
    merchantName           CHAR(50) NOT NULL,
    accountID              CHAR(10) NOT NULL,
    dateTime               DATETIME NOT NULL,
    type                   CHAR(20) NOT NULL,
    pointsValue            INTEGER  NOT NULL,
    transactionAmount       DECIMAL(6,2) NOT NULL,
    FOREIGN KEY(promotionID) REFERENCES Promotion(promotionID)
    FOREIGN KEY(merchantID) REFERENCES Merchant(merchantID)
    FOREIGN KEY(accountID) REFERENCES Account(accountID)
```

);

7) Table population

```
INSERT INTO Account1(accountID, pointBalance, streetAddress,
city, postalCode, country)
VALUES    ('A1001', 0, '3308 Ash St', 'Vancouver', 'V5Z 3E3',
          'Canada'),
          ('A1002', 0, '374 Brisdale Dr, Brampton', 'L7A 3M5',
          'Canada'),
          ('A1003', 0, '500 Kingston Rd', 'Toronto', 'M4L 1V3',
          'Canada'),
          ('A1004', 100, '7503 Rue St Denis', 'Montreal', 'H2R
          2E7', 'Canada'),
          ('A1005', 150, '3124 Doctors Drive', 'Los Angeles',
          '90017', 'USA');
```

```
INSERT INTO Account2(postalCode, country, provinceState)
VALUES    ('V5Z 3E3', 'Canada', 'British Columbia'),
          ('L7A 3M5', 'Canada', 'Ontario'),
          ('M4L 1V3', 'Canada', 'Ontario'),
          ('H2R 2E7', 'Canada', 'Quebec'),
          ('90017', 'USA', 'California');
```

```
INSERT INTO CreditCard1 (creditCardID, creditCardNum, accountID,
expirationDate)
VALUES    ('C1001', '4147382978379182', 'A1001', 2025-01-01),
          ('C1002', '5214231107639819', 'A1002', 2023-03-01),
          ('C1003', '5214232637822867', 'A1003', 2022-03-01),
          ('C1004', '3413741564427891', 'A1004', 2022-02-01),
          ('C1005', '4246315236423180', 'A1005', 2021-01-01);
```

```
INSERT INTO CreditCard2 (creditCardNum, cardType, cardIssuer)
VALUES    ('4147382978379182', 'visa', 'Royal Bank of Canada'),
          ('5214231107639819', 'visa', 'TD Canada Trust'),
          ('5214232637822867', 'mastercard', 'TD Canada Trust'),
          ('3413741564427891', 'mastercard', 'Scotia Bank Canada');
```

```
( '4246315236423180', 'visa', 'Bank of America' );
```

```
INSERT INTO Member(memberID, accountID, memberName,
emailAddress, phoneNumber, birthDate, referrerID)
VALUES  ( 'M1001', 'A1001', 'Florence R.Cummings',
        'florence@gmail.com', '647-897-8250', 1982-03-14, null),
        ( 'M1002', 'A1002', 'Stephanie R. McCarthy',
        'stephanie@gmail.com', '514-887-2380', 1961-09-04,
        null),
        ( 'M1003', 'A1003', 'Charles M. Freeman',
        'charles@gmail.com', '604-435-5767', 1977-10-07,
        'M1002'),
        ( 'M1004', 'A1004', 'Tracy G. Davis', 'tracy@gmail.com',
        '705-440-7929', 1989-04-15, 'M1003'),
        ( 'M1005', 'A1005', 'Leonard S. Cass',
        'leonard@gmail.com', '281-791-2248', 2000-03-13,
        'M1001'),
        ( 'M1006', 'A1003', 'Laura W Simmons',
        'coralie.torp@gmail.com', '701-326-3675', 1972-08-20,
        'null'),
        ( 'M1007', 'A1001', 'Justin Smith',
        'justinsmith@gmail.com', '281-464-2248', 1992-06-29,
        'null'),
        ( 'M1008', 'A1001', 'John Smith', 'johnsmith@gmail.com',
        '202-791-2248', 2000-07-27, 'null'),
        ( 'M1009', 'A1002', 'Alison Liu', 'alison.liu@gmail.com',
        '281-791-2248', 1998-03-13, 'M1007'),
        ( 'M1010', 'A1004', 'David Barrett',
        'david.barrett@gmail.com', '908-992-2248', 1995-06-13,
        'M1001');
```

```
INSERT INTO SupplementaryMember(memberID, accountID,
primaryMemberID);
VALUES ( 'M1006', 'A1003', 'M1003'),
        ( 'M1007', 'A1001', 'M1001'),
        ( 'M1008', 'A1001', 'M1001'),
        ( 'M1009', 'A1002', 'M1002'),
        ( 'M1010', 'A1004', 'M1004');
```

```
INSERT TO Merchant(merchantID, merchantName, joinDate,
defaultRate)
```

```
VALUES ('MC1001', 'Lululemon', 2019-12-20, 0.2),
      ('MC1002', 'Starbucks', 2021-01-15, 0.1),
      ('MC1003', 'SportChek', 2020-05-01, 0.8),
      ('MC1004', 'Ikea', 2020-08-01, 1.5),
      ('MC1005', 'Home Depot', 2020-09-01, 0.5);
```

```
INSERT TO Promotion_Offers(promotionID, merchantID,
promotionRate, startDate, endDate)
```

```
VALUES ('P1001', 'MC1001', 2.0, 2019-12-01, 2021-03-05),
      ('P1002', null, 1.0, 2020-01-01, null),
      ('P1003', 'MC1003', 2.5, 2020-05-15, 2020-12-31),
      ('P1004', 'MC1004', 5.0, 2020-08-01, 2020-08-31),
      ('P1005', 'MC1005', 12.0, 2020-02-02, 2020-02-05);
```

```
INSERT TO Reward(rewardID, pointCost, rewardCategory,
rewardDescription);
```

```
VALUES ('R1001', 5000, 'Gift Card', '$50 Starbucks Card'),
      ('R1002', 500000, 'Merchandise', 'iPad 64GB'),
      ('R1003', 1000, 'Gift Card', '10 Starbucks Card'),
      ('R1004', 500, 'Donation', 'Food Bank $5 Donation'),
      ('R1005', 25000, 'Travel', 'Domestic Flight Ticket');
```

```
INSERT TO Redeems(rewardID, accountID, memberID, dateTime);
```

```
VALUES ('R1001', 'A1001', 'M1001', 2020-08-24 13:45:23),
      ('R1003', 'A1002', 'M1002', 2021-01-16 11:00:00),
      ('R1004', 'A1003', 'M1003', 2021-01-17 09:37:12),
      ('R1004', 'A1004', 'M1004', 2021-02-13 04:01:56),
      ('R1001', 'A1005', 'M1005', 2021-02-21 22:47:41);
```

```
INSERT TO Survey(surveyID, pointsValue, expirationDate);
```

```
VALUES ('S1001', 50, 2021-03-01),
      ('S1002', 25, 2020-09-01),
      ('S1003', 10, 2021-05-01),
      ('S1004', 10, null),
      ('S1005', 10, 2022-12-31);
```



```
INSERT TO FillsOut(accountID, memberID, surveyID, dateTime);
VALUES ('A1001', 'M1001', 'S1001', 2021-02-28 11:00:00),
       ('A1002', 'M1002', 'S1002', 2020-01-01 09:43:22),
       ('A1003', 'M1003', 'S1003', 2020-06-24 10:15:44),
       ('A1004', 'M1004', 'S1004', 2019-03-10 15:30:01),
       ('A1005', 'M1005', 'S1005', 2018-09-22 20:21:22),
       ('A1004', 'M1010', 'S1003', 2019-11-13 11:13:45);
```

```
INSERT TO Transaction(transactionID, promotionID, merchantID,
merchantName, accountID, dateTime, type, pointsValue,
transactionAmount);
VALUES ('T1001', 'P1001', 'MC1001', 'Lululemon', 'A1001',
2021-02-28 11:00:00, 'refund', -52, -52.00),
       ('T1002', null, null, 'A1 Computers', 'A1003', 2020-12-27
16:23:18, 'purchase', 0, 15.45),
       ('T1003', 'P1001', 'MC1003', 'SportChek', 'A1001',
2021-03-31 12:38:46, 'purchase', 65, 65.47),
       ('T1004', 'P1002', 'MC1004', 'Ikea', 'A1001', 2019-04-03
18:37:00, 'purchase', 320, 320.06),
       ('T1005', 'P1003', 'MC1002', 'Starbucks', 'A1005',
2021-01-30 12:15:00, 'purchase', 6, 5.50);
```

8) Queries

Insertion: Insert a new purchase transaction to the transaction table.

Delete: Delete an account that has no members and at least 1 credit card associated with it.

Update: Update an account ID number that has at least 1 member associated with it.

Select: Select all transactions of type purchase from the transaction table.

Projection: Project the merchant name, datetime, type, and amount from the transaction table.

Join: Find all purchase transactions made by accounts in Canada that have a promotion rate of >2% applied to them.

Division: Find the rewards that have been redeemed by all primary members in 2020.