University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: 2

Date: October 15, 2024

Group Number: 41

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Terence Yin	47554431	n6p7i@ugrad.cs.ubc.ca	terence.yin76@gmail.com
Fegico Chen	14033468	b9c8r@ugrad.cs.ubc.ca	jakeyeozh@gmail.com
Jake Yeo	86759529	k3c3v@ugrad.cs.ubc.ca	fegicochen@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.)

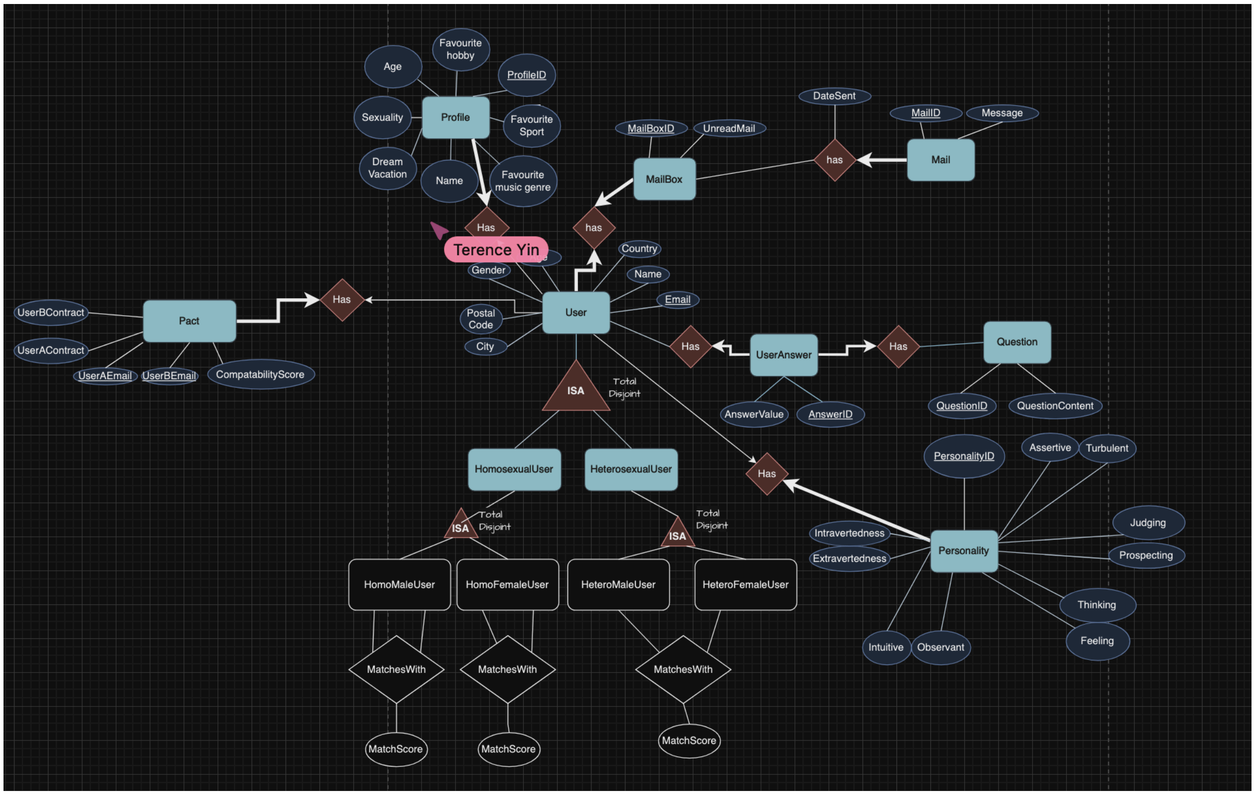
In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

Milestone 2

2) This application is a social matchmaking platform that uses personality assessments to connect individuals seeking compatible partners. Users complete standardized surveys to derive their personality traits, which are then analyzed to generate compatibility scores and facilitate meaningful matches. The app also features a messaging system and a unique marriage pact option, where users commit to marrying if both remain single by the age of 40. Built using a tech stack of Node.js, React, and MySQL, the application aims to create lasting connections based on shared values and compatibility.

3) ER Diagram

- Removed UserID and made Email primary key
- Removed mailbox entity
- Added participation constraint and many to one relationship to useranswer and question
- Added constraints to all three ISA relationships



4) **Schema**

User
(**Email: VARCHAR PRIMARY KEY,**
Name: VARCHAR(255),
Age: INTEGER,
Gender: VARCHAR(255),
Country: VARCHAR(255),
PostalCode: CHAR(6)
PersonalityID: CHAR(8),
ProfileID: CHAR(8),
MailBoxID: CHAR(8),
City: VARCHAR(255),
FOREIGN KEY (ProfileID) REFERENCES Profile(ProfileID) ON DELETE CASCADE,
FOREIGN KEY (PersonalityID) REFERENCES Personality(PersonalityID) ON DELETE CASCADE,
FOREIGN KEY (MailBoxID) REFERENCES Personality(PersonalityID) ON DELETE CASCADE
)

Pact(
User_A_Email: VARCHAR(255),
User_B_Email: VARCHAR(255),
Compatibility_Score: NUMERIC,
User_A_Contract: VARCHAR(255),
User_B_Contract: VARCHAR(255),
FOREIGN KEY (User_A_Email) REFERENCES User(Email),
FOREIGN KEY (User_B_Email) REFERENCES User(Email),
FOREIGN KEY (User_A_Contract) REFERENCES UserAContract(Contract),
FOREIGN KEY (User_B_Contract) REFERENCES UserBContract(Contract),
PRIMARY KEY (User_A_Email, User_B_Email)
)

Mail (
MailID CHAR(20): PRIMARY KEY,
Email: VARCHAR(255),
Message: VARCHAR(255),
FOREIGN KEY (Email) REFERENCES User(Email)
)

Mailbox (
MailboxID: VARCHAR **PRIMARY KEY,**
UnreadMail: INT
)

UserAnswer (
AnswerID: VARCHAR(255) **PRIMARY KEY,**
AnswerValue: DECIMAL,
FOREIGN KEY (Email) REFERENCES User(Email)
FOREIGN KEY (QuestionID) **REFERENCES Question(QuestionID)**
)

Question
(QuestionID: CHAR(8) **PRIMARY KEY**,
QuestionContent: CHAR(2000)
)

Profile (
ProfileID: CHAR(20) **PRIMARY KEY,**
Name: CHAR(20),
Age: INTEGER,
Sexuality: CHAR(10),
DreamVacation: CHAR(50),
FavouriteHobby: CHAR(30),
FavouriteSport: CHAR(30),
FavouriteMusicGenre: CHAR(30)
)

Personality (
PersonalityID: CHAR(8) **PRIMARY KEY,**
Introvertedness: INTEGER,
Extrovertedness: INTEGER,
Intuitive: INTEGER,
Observant: INTEGER,
Thinking: INTEGER,
Feeling: INTEGER,
Prospecting: INTEGER,
Judging: INTEGER,
Turbulent: INTEGER,
Assertive: INTEGER
)

5. Functional Dependencies (FDs)

User
PostalCode -> City, Country
Email -> Name, City, Age, Gender, PostalCode, Gender, Country

Profile
ProfileID **->** FavouriteSport**,** FavouriteMusicGenre, Name, DreamVacation, Sexuality, Age, FavouriteHobby

UserAnswer
AnswerID -> AnswerValue
(Email, QuestionID) -> AnswerValue

Personality
PersonalityID -> Extrovertedness, Introvertedness, Intuitive, Observant,Judging, Prospecting,Thinking, Feeling, Assertive, Turbulent
Extrovertedness -> Introvertedness
Introvertedness -> Extrovertedness
Intuitive -> Observant
Observant -> Intuitive
Judging -> Prospecting
Prospecting -> Judging
Thinking -> Feeling
Feeling -> Thinking
Assertive -> Turbulent
Turbulent -> Assertive

Question
QuestionID -> QuestionContent
QuestionContent -> QuestionID

MailBox
MailBoxID->UnreadMail

Mail
MailID->Message

Pact
UserBEmail, UserAEmail -> CompatabilityScore, UserBContract, UserAContract
UserAEmail -> UserAContract
UserBEmail -> UserBContract

6 Normalization)

Decomposing Personality:
PersonalityID -> Extrovertedness
PersonalityID -> Introvertedness
PersonalityID -> Intuitive

PersonalityID -> Observant
PersonalityID -> Judging
PersonalityID -> Prospecting
PersonalityID-> Thinking
PersonalityID -> Feeling
PersonalityID -> Assertive
PersonalityID -> Turbulent
Extrovertedness -> Introvertedness
Introvertedness -> Extrovertedness
Intuitive -> Observant
Observant -> Intuitive
Judging -> Prospecting
Prospecting -> Judging
Thinking -> Feeling
Feeling -> Thinking
Assertive -> Turbulent
Turbulent -> Assertive

1. Finding minimal cover:

   Put FD's in standard form. All FD's are already in standard form.

   Minimize LHS of FD's. All FD's are minimized with the RHS having one attribute.

   Delete redundant FD's:

   - PersonalityID+ = {Extrovertedness, Introvertedness, Intuitive, Observant, Judging, Prospecting, Thinking, Feeling, Assertive, Turbulent}
   - PersonalityID+ without considering (PersonalityID -> Extrovertedness) = {Extrovertedness, Introvertedness, Intuitive, Observant, Judging, Prospecting, Thinking, Feeling, Assertive, Turbulent}
     - Same closures, therefore (PersonalityID -> Extrovertedness) is redundant
   - PersonalityID+ without considering (PersonalityID -> Introvertedness) does not contain Introvertedness or Extrovertedness. Therefore (PersonalityID -> Introvertedness) is not redundant.
     - **This logic follows with all FD's with form PersonalityID -> b**
       - **Therefore, every second PersonalityID -> b FD can be removed**
   - Extrovertedness+ = { Extrovertedness, Introvertedness}
   - Extrovertedness+ without considering (Extrovertedness → Introvertedness) = {Extrovertedness}
     - The closures are different, therefore (Extrovertedness → Introvertedness) is not redundant.

- **The remaining FD’s not in the form PersonalityID -> b all follow the above pattern/logic for deleting redundant FD’s. Therefore no FD’s of this form are redundant.**

Minimal Cover:

PersonalityID -> Introvertedness

PersonalityID -> Observant

PersonalityID -> Prospecting

PersonalityID -> Feeling

PersonalityID -> Turbulent

Extrovertedness -> Introvertedness

Introvertedness -> Extrovertedness

Intuitive -> Observant

Observant -> Intuitive

Judging -> Prospecting

Prospecting -> Judging

Thinking -> Feeling

Feeling -> Thinking

Assertive -> Turbulent

Turbulent -> Assertive

Using Synthesis:

R1(PersonalityID, Introvertedness)

R2(PersonalityID, Observant)

R3(PersonalityID, Prospecting)

R4(PersonalityID, Feeling)

R5(PersonalityID, Turbulent)

R6(Extrovertedness, Introvertedness)

R7(Introvertedness, Extrovertedness)

R8(Intuitive, Observant)

R9(Observant, Intuitive)

R10(Judging, Prospecting)

R11(Prospecting, Judging)

R12(Thinking, Feeling)

R13(Feeling, Thinking)

R14(Assertive, Turbulent)

R15(Turbulent, Assertive)

After decomposition there is still some redundancy. For example, R7 is redundant since its attributes are contained within R6. Removed all redundant relationships.

Final Relationships after Synthesis:

R1(PersonalityID, Introvertedness)

R2(PersonalityID, Observant)

R3(PersonalityID, Prospecting)

R4(PersonalityID, Feeling)

R5(PersonalityID, Turbulent)

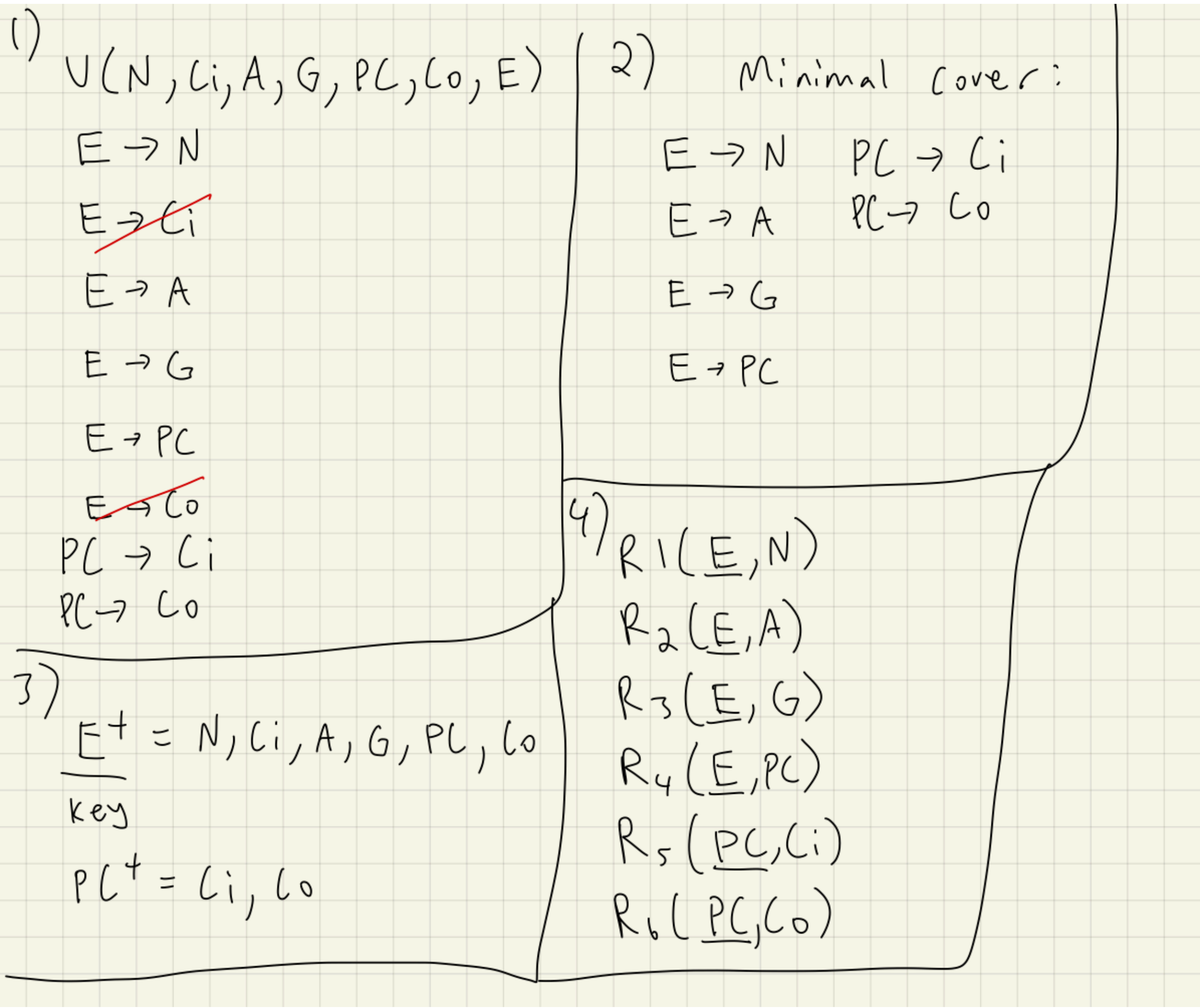
R6(Extrovertedness, Introvertedness)

R7(Intuitive, Observant)

R8(Judging, Prospecting)

R9(Thinking, Feeling)

R10(Assertive, Turbulent)



1)

$U(N, Ci, A, G, PC, Co, E)$

$E \to N$

~~$E \to Ci$~~

$E \to A$

$E \to G$

$E \to PC$

~~$E \to Co$~~

$PC \to Ci$

$PC \to Co$

2) Minimal cover:

$E \to N$ $PC \to Ci$

$E \to A$ $PC \to Co$

$E \to G$

$E \to PC$

3)

$\underline{E^+} = N, Ci, A, G, PC, Co$

key

$PC^+ = Ci, Co$

4)

$R1(\underline{E}, N)$

$R_2(\underline{E}, A)$

$R_3(\underline{E}, G)$

$R_4(\underline{E}, PC)$

$R_5(\underline{PC}, Ci)$

$R_6(\underline{PC}, Co)$

<u>Decomposed User Table</u>:

User (
Email: VARCHAR **PRIMARY KEY**,
Name: VARCHAR,
PersonalityID: CHAR(8),
ProfileID: CHAR(8),
MailBoxID: CHAR(8),
PostalCode: VARCHAR,
FOREIGN KEY (ProfileID) REFERENCES Profile(ProfileID) ON DELETE CASCADE,
FOREIGN KEY (PersonalityID) REFERENCES Personality(PersonalityID) ON DELETE CASCADE,

FOREIGN KEY (MailBoxID) REFERENCES Personality(PersonalityID) ON DELETE CASCADE
FOREIGN KEY (PostalCode) REFERENCES PostalCode(PostalCode) ON DELETE CASCADE
)

UserEmailAge(
Email: VARCHAR **PRIMARY KEY**,
Age: VARCHAR,
FOREIGN KEY (Email) REFERENCES User(Email) ON DELETE CASCADE
)

UserEmailGender(
Email: VARCHAR **PRIMARY KEY**,
Gender: VARCHAR,
FOREIGN KEY (Email) REFERENCES User(Email) ON DELETE CASCADE
)

UserEmailPostalCode(
Email: VARCHAR **PRIMARY KEY**,
PostalCode: VARCHAR,
FOREIGN KEY (Email) REFERENCES User(Email) ON DELETE CASCADE
)

PostalCode (
PostalCode: VARCHAR **PRIMARY KEY**,
Country: VARCHAR,
)

PostalCodeCity (
PostalCode: VARCHAR **PRIMARY KEY**,
City: VARCHAR,
FOREIGN KEY (PostalCode) REFERENCES PostalCode(PostalCode) ON DELETE CASCADE
)

Decomposed Pact Table:

UBE, UAE → CS, UBC, UAC

UAE → UAC

UBE → UBC

↓ To standard form

UAE → UAC

UBE → UBC

UBE, UAE → CS

UBE, UAE → UBC

UBE, UAE → UAC

↓ Minimize LHS

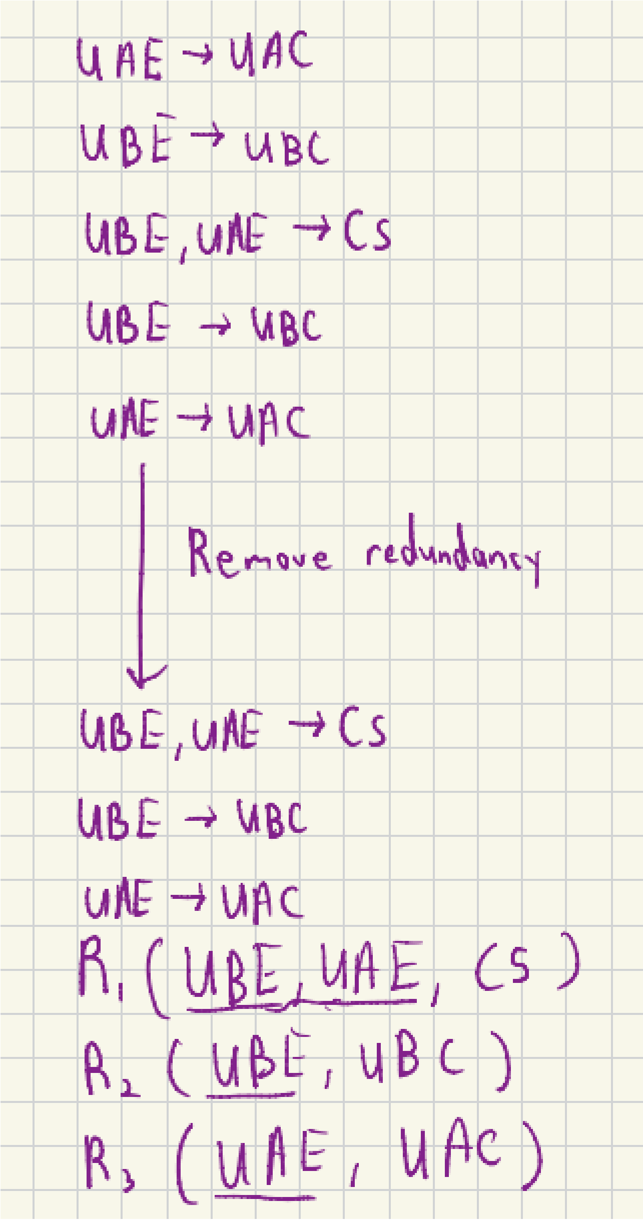
$UBE^{+} = \{UBE, UBC\}$

$UAE^{+} = \{UAE, UAC\}$

Remove UAE from UBE, UAE → UBC

Remove UAE from UBE, UAE → UBC

↓



UAE → UAC

UBE → UBC

UBE, UAE → CS

UBE → UBC

UAE → UAC

Remove redundancy

UBE, UAE → CS

UBE → UBC

UAE → UAC

$R_1(\underline{UBE, UAE}, CS)$

$R_2(\underline{UBE}, UBC)$

$R_3(\underline{UAE}, UAC)$

<u>Decomposed Personality Tables:</u>

PToIntro (
    PersonalityID: CHAR(8), **PRIMARY KEY,**
    Introvertedness: NUMBER
)

PToObservant (
    PersonalityID: CHAR(8), **PRIMARY KEY,**
    Observant: NUMBER
)

PToProspecting (
PersonalityID: CHAR(8), **PRIMARY KEY,**
Prospecting: NUMBER
)

PToFeeling (
PersonalityID: CHAR(8), **PRIMARY KEY,**
Feeling: NUMBER
)

PToTurbulent (
PersonalityID: CHAR(8), **PRIMARY KEY,**
Turbulent: NUMBER
)

ExtroToIntro (
Extrovertedness: NUMBER, **PRIMARY KEY,**
Introvertedness: NUMBER
)

IntuitiveToObservant (
Intuitive: NUMBER, **PRIMARY KEY**
Observant: NUMBER
)

JudgingToProspecting (
Judging: NUMBER, **PRIMARY KEY,**
Prospecting: NUMBER
)

ThinkingToFeeling (
Thinking: NUMBER, **PRIMARY KEY,**
Feeling: NUMBER
)

AssertiveToTurbulent (
Assertive: NUMBER, **PRIMARY KEY,**
Turbulent: NUMBER
)

Table definitions before normalizing:

User
(**Email: VARCHAR PRIMARY KEY,**
Name: VARCHAR(255),
Age: INTEGER,
Gender: VARCHAR(255),
Country: VARCHAR(255),
PostalCode: CHAR(6)
PersonalityID: CHAR(8),
ProfileID: CHAR(8),
MailBoxID: CHAR(8),
City: VARCHAR(255),
FOREIGN KEY (ProfileID) REFERENCES Profile(ProfileID) ON DELETE CASCADE,
FOREIGN KEY (PersonalityID) REFERENCES Personality(PersonalityID) ON DELETE CASCADE,
FOREIGN KEY (MailBoxID) REFERENCES Personality(PersonalityID) ON DELETE CASCADE
)

Pact(
User_A_Email: VARCHAR(255),
User_B_Email: VARCHAR(255),
Compatibility_Score: NUMERIC,
User_A_Contract: VARCHAR(255),
User_B_Contract: VARCHAR(255),
FOREIGN KEY (User_A_Email) REFERENCES User(Email),
FOREIGN KEY (User_B_Email) REFERENCES User(Email),
FOREIGN KEY (User_A_Contract) REFERENCES UserAContract(Contract),
FOREIGN KEY (User_B_Contract) REFERENCES UserBContract(Contract),
PRIMARY KEY (User_A_Email, User_B_Email)
)

Mail (
MailID CHAR(20): PRIMARY KEY,
Email: VARCHAR(255),
Message: VARCHAR(255),
FOREIGN KEY (Email) REFERENCES User(Email)
)

UserAnswer (
AnswerID: VARCHAR(255) **PRIMARY KEY,**
AnswerValue: DECIMAL,

FOREIGN KEY (Email) REFERENCES User(Email)
FOREIGN KEY (QuestionID) **REFERENCES Question(QuestionID)**
)

Question
(QuestionID: CHAR(8) **PRIMARY KEY**,
QuestionContent: CHAR(2000)
)

Profile (
ProfileID: CHAR(20) **PRIMARY KEY,**
Name: CHAR(20),
Age: INTEGER,
Sexuality: CHAR(10),
DreamVacation: CHAR(50),
FavouriteHobby: CHAR(30),
FavouriteSport: CHAR(30),
FavouriteMusicGenre: CHAR(30)
)

Personality (
PersonalityID: CHAR(8) **PRIMARY KEY,**
Introvertedness: INTEGER,
Extrovertedness: INTEGER,
Intuitive: INTEGER,
Observant: INTEGER,
Thinking: INTEGER,
Feeling: INTEGER,
Prospecting: INTEGER,
Judging: INTEGER,
Turbulent: INTEGER,
Assertive: INTEGER
)

7) SQL DDL statements

<u>UnNormalized SQL Statements</u>

```
CREATE TABLE Profile (
        ProfileID CHAR(20) PRIMARY KEY,
        Name VARCHAR(255),
        Age INTEGER,
        Sexuality VARCHAR(255),
        DreamVacation VARCHAR(255),
        FavouriteHobby VARCHAR(255),
        FavouriteSport VARCHAR(255),
        FavouriteMusicGenre VARCHAR(255)
);

CREATE TABLE Personality (
        PersonalityID CHAR(8) PRIMARY KEY,
        Intravertedness INTEGER,
        Extravertedness INTEGER,
        Intuitive INTEGER,
        Observant INTEGER,
        Thinking INTEGER,
        Feeling INTEGER,
        Prospecting INTEGER,
        Judging INTEGER,
        Turbulent INTEGER,
        Assertive INTEGER
);

CREATE TABLE User(
        Email VARCHAR(255) PRIMARY KEY,
        Name VARCHAR(255),
        Age INTEGER,
        Gender VARCHAR(255),
        Country VARCHAR(255),
        PostalCode CHAR(6),
        PersonalityID CHAR(8),
        ProfileID CHAR(8),
        MailBoxID CHAR(8),
        City VARCHAR(255),
        FOREIGN KEY (ProfileID) REFERENCES Profile(ProfileID) ON DELETE CASCADE,
        FOREIGN KEY (PersonalityID) REFERENCES Personality(PersonalityID) ON DELETE
CASCADE,
        FOREIGN KEY (MailBoxID) REFERENCES Personality(PersonalityID) ON DELETE
CASCADE
);

CREATE TABLE Pact (
UserEmailA: VARCHAR(255),
UserEmailB: VARCHAR(255),
UserAContract: VARCHAR(255),
UserBContract:VARCHAR(255),
CompatibilityScore: DECIMAL,
PRIMARY KEY (User_Email, User_Email2),
FOREIGN KEY (User_Email1) REFERENCES User(Email),
FOREIGN KEY (User_Email2) REFERENCES User(Email)
);

CREATE TABLE Mail (
        MailID CHAR(20) PRIMARY KEY,
        Email VARCHAR(255),
        Message VARCHAR(255),
        FOREIGN KEY (Email) REFERENCES User(Email)
);

CREATE TABLE Question (
        QuestionID CHAR(8) PRIMARY KEY,
        QuestionContent VARCHAR(2000)
);

CREATE TABLE UserAnswer (
        AnswerID CHAR(8) PRIMARY KEY,
        QuestionID CHAR(8),
        AnswerValue DECIMAL,
        Email VARCHAR(255),
        FOREIGN KEY (Email) REFERENCES User(Email),
        FOREIGN KEY (QuestionID) REFERENCES Question(QuestionID)
);

CREATE TABLE Pact (
        User_A_Email VARCHAR(255),
        User_B_Email VARCHAR(255),
        User_A_Contract VARCHAR(255),
        User_B_Contract VARCHAR(255),
        CompatabilityScore INTEGER,
        PRIMARY KEY (User_A_Email, User_B_Email)
);
```

Normalized SQL Statements

```
CREATE TABLE User (
Email VARCHAR PRIMARY KEY,
Name VARCHAR,
PersonalityID CHAR(8),
ProfileID CHAR(8),
MailBoxID CHAR(8),
PostalCode VARCHAR,
FOREIGN KEY (ProfileID) REFERENCES Profile(ProfileID) ON DELETE CASCADE,
FOREIGN KEY (PersonalityID) REFERENCES Personality(PersonalityID) ON DELETE
CASCADE,
FOREIGN KEY (MailBoxID) REFERENCES Personality(PersonalityID) ON DELETE
CASCADE
FOREIGN KEY (PostalCode) REFERENCES PostalCode(PostalCode) ON DELETE
CASCADE
);

CREATE TABLE UserEmailAge(
Email VARCHAR PRIMARY KEY,
Age VARCHAR,
FOREIGN KEY (Email) REFERENCES User(Email) ON DELETE CASCADE
);

CREATE TABLE UserEmailGender(
Email VARCHAR PRIMARY KEY,
Gender VARCHAR,
FOREIGN KEY (Email) REFERENCES User(Email) ON DELETE CASCADE
);

CREATE TABLE UserEmailPostalCode(
Email VARCHAR PRIMARY KEY,
PostalCode VARCHAR,
FOREIGN KEY (Email) REFERENCES User(Email) ON DELETE CASCADE
)

CREATE TABLE PostalCode (
        PostalCode VARCHAR PRIMARY KEY,
        Country VARCHAR,
)

CREATE TABLE PostalCodeCity (
        PostalCode VARCHAR PRIMARY KEY,
        City VARCHAR,
FOREIGN KEY (PostalCode) REFERENCES PostalCode(PostalCode) ON DELETE
CASCADE
)

CREATE TABLE Mail (
        MailID CHAR(20) PRIMARY KEY,
        Email VARCHAR(255),
        Message VARCHAR(255),
        FOREIGN KEY (Email) REFERENCES User(Email)
);

CREATE TABLE Question (
        QuestionID CHAR(8) PRIMARY KEY,
        QuestionContent VARCHAR(2000)
);

CREATE TABLE UserAnswer (
        AnswerID CHAR(8) PRIMARY KEY,
        QuestionID CHAR(8),
        AnswerValue DECIMAL,
        Email VARCHAR(255),
        FOREIGN KEY (Email) REFERENCES User(Email),
        FOREIGN KEY (QuestionID) REFERENCES Question(QuestionID)
);

CREATE TABLE Matches (
        User_Email_A VARCHAR(255),
        User_Email_B VARCHAR(255),
        MatchScore DECIMAL,
        PRIMARY KEY (User_Email_A, User_Email_B),
        FOREIGN KEY (User_Email_A) REFERENCES User(Email),
        FOREIGN KEY (User_Email_B) REFERENCES User(Email)
);



CREATE TABLE PToIntro (
   PersonalityID CHAR(8) PRIMARY KEY,
   Introvertedness DECIMAL(5, 2)
);


CREATE TABLE PToObservant (
   PersonalityID CHAR(8) PRIMARY KEY,
   Observant DECIMAL(5, 2)
);

CREATE TABLE PToProspecting (
   PersonalityID CHAR(8) PRIMARY KEY,
   Prospecting DECIMAL(5, 2)
);

CREATE TABLE PToFeeling (
   PersonalityID CHAR(8) PRIMARY KEY,
   Feeling DECIMAL(5, 2)
);
```

```
CREATE TABLE PToTurbulent (
   PersonalityID CHAR(8) PRIMARY KEY,
   Turbulent DECIMAL(5, 2)
);


CREATE TABLE ExtroToIntro (
   Extrovertedness DECIMAL(5, 2) PRIMARY KEY,
   Introvertedness DECIMAL(5, 2)
);

CREATE TABLE IntuitiveToObservant (
   Intuitive DECIMAL(5, 2) PRIMARY KEY,
   Observant DECIMAL(5, 2)
);

CREATE TABLE JudgingToProspecting (
   Judging DECIMAL(5, 2) PRIMARY KEY,
   Prospecting DECIMAL(5, 2)
);

CREATE TABLE ThinkingToFeeling (
   Thinking DECIMAL(5, 2) PRIMARY KEY,
   Feeling DECIMAL(5, 2)
);

CREATE TABLE AssertiveToTurbulent (
   Assertive DECIMAL(5, 2) PRIMARY KEY,
   Turbulent DECIMAL(5, 2)
);

CREATE TABLE UserAContract (
   User_A_Email VARCHAR(255) PRIMARY KEY,
   Contract VARCHAR(255),
   FOREIGN KEY (User_A_Email) REFERENCES User(Email)
);

CREATE TABLE UserBContract (
   User_B_Email VARCHAR(255) PRIMARY KEY,
   Contract VARCHAR(255),
   FOREIGN KEY (User_B_Email) REFERENCES User(Email)
);
```

```
CREATE TABLE PACT(
        User_A_Email VARCHAR(255),
        User_B_Email VARCHAR(255),
        Compatibility_Score NUMERIC,
        User_A_Contract VARCHAR(255),
        User_B_Contract VARCHAR(255),
        FOREIGN KEY (User_A_Email) REFERENCES User(Email),
        FOREIGN KEY (User_B_Email) REFERENCES User(Email),
        FOREIGN KEY (User_A_Contract) REFERENCES UserAContract(Contract),
        FOREIGN KEY (User_B_Contract) REFERENCES UserBContract(Contract),
        PRIMARY KEY (User_A_Email, User_B_Email)
)
CREATE TABLE UserAContract(
        User_A_Email: VARCHAR(255) PRIMARY KEY,
        Contract: VARCHAR(255),
        FOREIGN KEY (USER_A_EMAIL) REFERENCES User(Email),
)

CREATE TABLE UserBContract(
        User_B_Email: VARCHAR(255) PRIMARY KEY,
        Contract: VARCHAR(255),
        FOREIGN KEY (USER_B_EMAIL) REFERENCES User(Email),
)
```

8.) **INSERT STATEMENTS**

```
INSERT INTO UserAnswer (AnswerID, QuestionID, AnswerValue, Email)
VALUES
(‘1’, ‘2’, “Very2”, “gmail1”),
(‘12’, ‘2’, “Very3”, “gmail2”),
(‘13’, ‘2’, “Ver1y”, “gmail3”),
(‘14’, ‘2’, “Very8”, “gmail4”),
(‘15’, ‘2’, “Ver7y”, “gmail5”);’

INSERT INTO Mail (MailID, Email, Message)
VALUES
(‘12’, ‘gmai1l’, ‘ME342E’),
(‘13’, ‘gmai2l’, ‘ME342E’),
(‘14’, ‘gma3il’, ‘M11EE’),
(‘15’, ‘gm4ail’, ‘ME32E’),
(‘16’, ‘gm5ail’, ‘ME423E’);

INSERT INTO Question (QuestionID, QuestionContent)
VALUES
```

```
(‘1’, ‘You like ice?’),
(‘12’, ‘You like mew?’),
(‘13’, ‘You like weww?’),
(‘14’, ‘You like see?’),
(‘15’, ‘You like dde?’);

INSERT INTO Matches (User_Email_A, User_Email_B )
VALUES
(‘My1 Email’, ‘My Email22’, 0.5),
(‘My2 Email’, ‘My Email222’, 0.6),
(‘My3 Email’, ‘My Email232’, 0.7),
(‘My4 Email’, ‘My Email233’, 0.8);

(‘My5 Email’, ‘My Email243’, 0.9)
INSERT INTO UserBContract (User_B_Email, Contract)
VALUES
(‘jordon@gmail.com’, ‘Marry Me’),
(‘jord23on@gmail.com’, ‘Ma43rry Me’),
(‘jord434on@gmail.com’, ‘M22arry Me’),
(‘jo232rdon@gmail.com’, ‘M33arry Me’),
(‘jord434on@gmail.com’, ‘M44arry Me’);

INSERT INTO UserAContract (User_A_Email, Contract)
VALUES
(‘jordon@outlook.com’, ‘Marry you’),
(‘jord23on@outlook.com’, ‘Ma43rry youMe’),
(‘jord434on@outlook.com’, ‘M22arry Myoue’),
(‘jo232rdon@outlook.com’, ‘M33arry Me you’),
(‘jord434on@outlook.com’, ‘M44arry Me you’);

INSERT INTO PToIntro (PersonalityID, Introvertedness)
VALUES
('Mike1234', 0.25),
('Lily5678', 0.60),
('David910', 0.45),
('Emma2345', 0.99),
('Zoey8765', 0.75);

INSERT INTO PToObservant (PersonalityID, Observant)
VALUES
('Mike1234', 0.25),
('Lily5678', 0.60),
('David910', 0.45),
('Emma2345', 0.99),
('Zoey8765', 0.75);

INSERT INTO PToProspecting (PersonalityID, Prospecting)
VALUES
('Mike1234', 0.25),
('Lily5678', 0.60),
('David910', 0.45),
('Emma2345', 0.99),
('Zoey8765', 0.75);

INSERT INTO PToFeeling (PersonalityID, Feeling)
VALUES
('Mike1234', 0.25),
('Lily5678', 0.60),
('David910', 0.45),
('Emma2345', 0.99),
('Zoey8765', 0.75);

INSERT INTO PToTurbulent (PersonalityID, Turbulent)
VALUES
('Mike1234', 0.25),
('Lily5678', 0.60),
('David910', 0.45),
('Emma2345', 0.99),
('Zoey8765', 0.75);

INSERT INTO ExtroToIntro (Extrovertedness, Introvertedness)
VALUES
(0.2, 0.25),
(0.1, 0.60),
(0.5, 0.45),
(0.31, 0.99),
(0.45, 0.75);

INSERT INTO IntuitiveToObservant (Intuitive, Observant)
VALUES
(0.2, 0.25),
(0.1, 0.60),
(0.5, 0.45),
(0.31, 0.99),
(0.45, 0.75);

INSERT INTO JudgingToProspecting (Judging, Prospecting)
VALUES
```

```
(0.2, 0.25),
(0.1, 0.60),
(0.5, 0.45),
(0.31, 0.99),
(0.45, 0.75);

INSERT INTO ThinkingToFeeling (Thinking, Feeling)
VALUES
(0.2, 0.25),
(0.1, 0.60),
(0.5, 0.45),
(0.31, 0.99),
(0.45, 0.75);

INSERT INTO AssertiveToTurbulent (Assertive, Turbulent)
VALUES
(0.2, 0.25),
(0.1, 0.60),
(0.5, 0.45),
(0.31, 0.99),
(0.45, 0.75);

INSERT INTO User (Email, Name, PersonalityID, ProfileID, MailBoxID, PostalCode)
VALUES
('john.doe@example.com', 'John Doe', 'PERS001', 'PRO001', 'MBOX001', 'A1B 2C3'),
('jane.smith@example.com', 'Jane Smith', 'PERS002', 'PRO002', 'MBOX002', 'D4E 5F6'),
('alice.johnson@example.com', 'Alice Johnson', 'PERS003', 'PRO003', 'MBOX003', 'G7H 8I9'),
('bob.brown@example.com', 'Bob Brown', 'PERS001', 'PRO001', 'MBOX001', 'J1K 2L3'),
('charlie.white@example.com', 'Charlie White', 'PERS002', 'PRO004', 'MBOX004', 'M4N 5O6');

INSERT INTO UserEmailAge (Email, Age)
VALUES
('john.doe@example.com', '25'),
('jane.smith@example.com', '30'),
('alex.brown@example.com', '22'),
('emily.jones@example.com', '28'),
('michael.white@example.com', '35');

INSERT INTO UserEmailGender (Email, Gender)
VALUES
('john.doe@example.com', 'Male'),
('jane.smith@example.com', 'Female'),
('alex.brown@example.com', 'Non-binary'),
('emily.jones@example.com', 'Female'),
('michael.white@example.com', 'Male');

INSERT INTO UserEmailPostalCode (Email, PostalCode)
VALUES
('john.doe@example.com', 'R3C 0X7'),
('jane.smith@example.com', 'V5K 0A1'),
('alex.brown@example.com', 'H1A 0B2'),
('emily.jones@example.com', 'K2J 3A4'),
('michael.white@example.com', 'L5B 1Z5');

INSERT INTO PostalCode (PostalCode, Country) VALUES
('R3C 0X7', 'Canada'),
('V5K 0A1', 'Canada'),
('10001', 'United States'),
('SW1A 1AA', 'United Kingdom'),
('A1A 1A1', 'Canada');

INSERT INTO PostalCodeCity (PostalCode, City) VALUES
('R3C 0X7', 'Winnipeg'),
('V5K 0A1', 'Vancouver'),
('10001', 'New York'),
('SW1A 1AA', 'London'),
('A1A 1A1', 'St. John’s');

INSERT INTO PACT (User_A_Email, User_B_Email, Compatibility_Score, User_A_Contract,
User_B_Contract) VALUES
('john.doe@example.com', 'jane.smith@example.com', 85.0, 'Contract1', 'Contract2'),
('alex.brown@example.com', 'emily.jones@example.com', 90.5, 'Contract3', 'Contract4'),
('michael.white@example.com', 'john.doe@example.com', 75.2, 'Contract5', 'Contract1'),
('jane.smith@example.com', 'alex.brown@example.com', 88.3, 'Contract2', 'Contract3'),
('emily.jones@example.com', 'michael.white@example.com', 80.1, 'Contract4', 'Contract5');
```