University of British Columbia, Vancouver

Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: 2

Date: Monday, 12 June 2023

Group Number: 10

Name	Student Number	CS Alias (Userid)	Preferred Email Address
Karen Agustino	10782498	q3d6f	karenag@student.ubc.ca
Saren Vathanak	87347696	z8h1l	vathanaksaren@gmail.com
Shaun Foo	55629265	d7i1y	chuanshianfoo@gmail.com

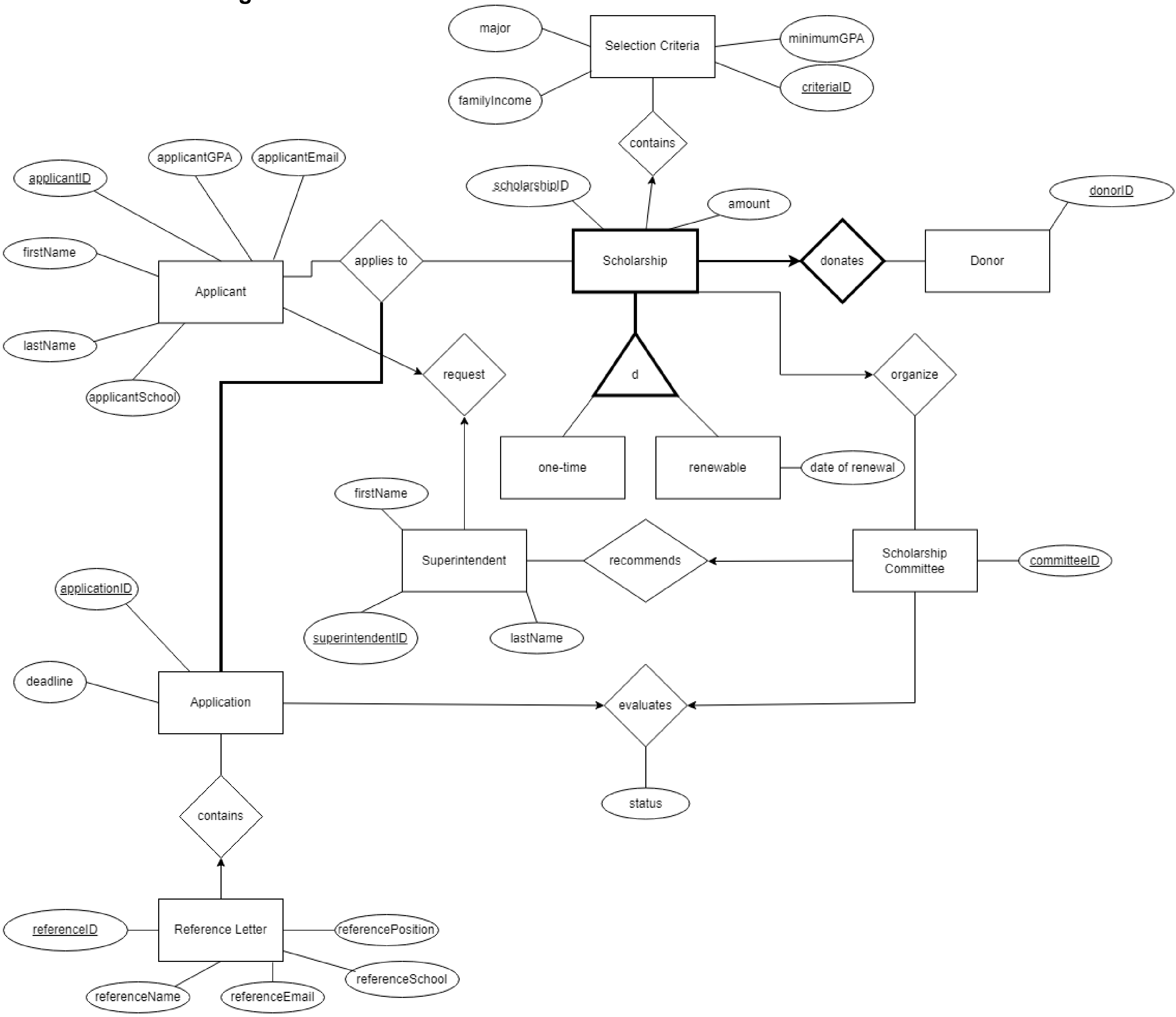
By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your email address,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

Our project is a scholarship portal, where the main entities are applicants who are linked to applications and the scholarship that they are applying to. Scholarships will either be issued on a one-time or a renewable basis (as seen from our ISA). Additionally, a scholarship cannot exist without a donor, and a scholarship committee will evaluate the applicant's application. All entities are primarily identified by their respective IDs stored in the database.

3. ER Diagram



ERD Modifications from Milestone 1:

- There are several sets of selection criteria that exist for a scholarship to rely on. A set of selection criteria is uniquely identified by its criteriaID, which determine the other three related criteria, including major, minimum GPA, and family income. A scholarship can only contain one set of selection criteria, while many sets of selection criteria can be used for multiple scholarships (one-to-many).
- Applicant has been updated with additional attributes: applicantSchool, applicantGPA and applicantEmail.
- Reference Letter has been changed to be a regular entity instead of a weak entity, because it is optional for applicants to provide a reference letter for their applications. Additionally, the entity has also been updated to include more attributes: referenceEmail, referenceSchool, and referencePosition.

4. Schema derived from E/R Diagram

Key:
Underlined Attributes = Primary Key
Bolded Attributes = Foreign Key

- Applicant(applicantID: int, firstName: varchar, lastName: varchar, applicantEmail: varchar, applicantSchool: varchar, applicantGPA: decimal)
- Application(applicationID: int, deadline: date)
- ReferenceLetter(referenceID: int, **applicationID:** int, referenceName: varchar, referenceEmail: varchar, referenceSchool : varchar, referencePosition: varchar)
- AppliesTo(**applicantID:** int, **applicationID**: int, **scholarshipID**: int, **donorID**: int)
- OneTime(scholarshipID: int, **donorID:** int, amount:int)
- Renewable(scholarshipID: int, **donorID**: int, amount: int, dateOfRenewal: date)
- ScholarshipCommittee(committeeID: int)
- Donor(donorID: int)
- Superintendent(superintendentID: int, firstName: varchar, lastName: varchar)
- SelectionCriteria(criteriaID: int, major: varchar, minimumGPA: decimal, familyIncome: int)
- evaluates(**applicationID**: int, **committeeID**: int, status: varchar)

5. Functional Dependencies

Applicant
applicantID → firstName, lastName, applicantEmail
applicantEmail → applicantSchool
applicantEmail → applicantGPA

Application
applicationID → deadline

ReferenceLetter
referenceID → applicationID, referenceName, referenceEmail
referenceEmail → referenceSchool, referencePosition

AppliesTo
No FDs as there are no non-key attributes

Superintendent
superintendentID → firstName, lastName

OneTime
scholarshipID, donorID → amount

Renewal
scholarshipID, donorID → amount, dateOfRenewal

ScholarshipCommittee
No FDs as there are no non-key attributes

Donor
No FDs as there are no non-key attributes

SelectionCriteria
criteriaID → major, familyIncome, minimumGPA

Evaluates
ApplicationID, CommitteeID → status

6. Normalization

Applicant table

Step 1: Minimal cover

applicantID → firstName
applicantID → lastName
applicantID → applicantEmail
applicantEmail → applicantSchool
applicantEmail → applicantGPA

Step 2: Determine if any FD violates BCNF, or if all FDs adhere to 3NF

applicantEmail → applicantSchool AND applicantEmail → applicantGPA violate BCNF

Step 3: Perform BCNF decomposition

*First decomposition (*Decompose applicantEmail → applicantSchool):
R1 = (applicantEmail, applicantSchool)
R2 = (applicantEmail, applicantID, firstName, lastName, applicantGPA)

Second decomposition (Decompose applicantEmail → applicantGPA) *:*
R3 = (applicantEmail, applicantGPA)
R4 = (applicantEmail, applicantID, firstName, lastName)

Final relations:
R1 = (applicantEmail, applicantSchool)
R3 = (applicantEmail, applicantGPA)
R4 = (applicantEmail, applicantID, firstName, lastName)

ReferenceLetter table

Step 1: Minimal cover

referenceID → applicationID
referenceID → referenceName
referenceID → referenceEmail
referenceEmail → referenceSchool
referenceEmail→ referencePosition

Step 2: Determine if any FD violates BCNF, or if all FDs adhere to 3NF referenceEmail → referenceSchool AND referenceEmail → referencePosition violate BCNF

Step 3: Perform BCNF decomposition

First Decomposition (Decompose referenceEmail → referenceSchool):
R1 = (referenceEmail, referenceSchool)
R2 = (referenceEmail, referenceID, applicationID, referenceName, referencePosition)

Second Decomposition (Decompose referenceEmail → referencePosition):
R3 = (referenceEmail, referencePosition)
R4 = (referenceEmail, referenceID, applicationID, referenceName)

Final Relations:
R1 = (referenceEmail, referenceSchool)
R3 = (referenceEmail, referencePosition)
R4 = (referenceEmail, referenceID, applicationID, referenceName)

7. SQL DDL statements

```
CREATE TABLE Applicant (
      ApplicantID      INTEGER,
      firstName        VARCHAR(40),
      lastName         VARCHAR(40),
      applicantSchool  VARCHAR(40),
      applicantGPA     DECIMAL(4,2),
      applicantEmail   VARCHAR(40),
      PRIMARY KEY (ApplicantID)
)


CREATE TABLE Application (
      ApplicationID    INTEGER,
      deadline         DATE,
      ApplicantID      INTEGER,
      PRIMARY KEY (ApplicationID),
      FOREIGN KEY (ApplicantID)
         REFERENCES Applicant(ApplicantID)
            ON DELETE CASCADE
)
```

```
CREATE TABLE AppliesTo (
      ApplicationID    INTEGER,
      ScholarshipID    INTEGER,
      ApplicantID      INTEGER,
      donorID           INTEGER,
      PRIMARY KEY (ApplicationID, ScholarshipID, ApplicantID,
donorID),
      FOREIGN KEY (ApplicationID)
         REFERENCES Application(ApplicationID)
            ON DELETE CASCADE,
      FOREIGN KEY (ScholarshipID)
         REFERENCES OneTime(ScholarshipID)
            ON DELETE CASCADE,
      FOREIGN KEY (ScholarshipID)
         REFERENCES Renewable(ScholarshipID)
            ON DELETE CASCADE,
      FOREIGN KEY (ApplicantID)
         REFERENCES Applicant(ApplicantID)
            ON DELETE CASCADE,
      FOREIGN KEY (donorID)
         REFERENCES Donor(donorID)
           ON DELETE CASCADE
)

CREATE TABLE Evaluates (
      ApplicationID    INTEGER,
      committeeID      INTEGER,
      status            VARCHAR(40),
      PRIMARY KEY (ApplicationID, committeeID),
      FOREIGN KEY (ApplicationID)
         REFERENCES Application(ApplicationID)
            ON DELETE CASCADE,
      FOREIGN KEY (committeeID)
         REFERENCES ScholarshipCommittee(committeeID)
            ON DELETE CASCADE
)

CREATE TABLE SelectionCriteria (
      criteriaID       INTEGER,
      minimumGPA       DECIMAL(4,2),
      major            VARCHAR(40),
      familyIncome     VARCHAR(40),
      PRIMARY KEY (criteriaID)
)

CREATE TABLE ScholarshipCommittee (
      committeeID      INTEGER,
      PRIMARY KEY (committeeID)
)

CREATE TABLE Superintendent (
      superintendentID INTEGER,
      firstName        VARCHAR(40),
      secondName       VARCHAR(40),
      PRIMARY KEY (superintendentID)
)

CREATE TABLE Renewable (
      scholarshipID    INTEGER,
      amount           INTEGER,
      dateOfRenewal    DATE,
      donorID          INTEGER NOT NULL,
      PRIMARY KEY (scholarshipID),
      FOREIGN KEY (donorID)
         REFERENCES Donor(donorID)
           ON DELETE CASCADE
)

CREATE TABLE OneTime (
      scholarshipID    INTEGER,
      amount           INTEGER,
      donorID          INTEGER NOT NULL,
      PRIMARY KEY (scholarshipID),
      FOREIGN KEY (donorID)
         REFERENCES Donor(donorID)
           ON DELETE CASCADE
)
```

```
CREATE TABLE Donor (
      donorID          INTEGER,
      PRIMARY KEY (donorID)
)

CREATE TABLE ReferenceLetter (
      referenceID           INTEGER,
      applicationID         INTEGER,
      referenceName         VARCHAR(40),
      referenceEmail        VARCHAR(40),
      referenceSchool       VARCHAR(40),
      referencePosition     VARCHAR(40),
      PRIMARY KEY (referenceID),
      FOREIGN KEY (applicationID)
         REFERENCES Application (applicationID)
           ON DELETE CASCADE
)
```

8. Populating tables

```
INSERT INTO Applicant
VALUES (12348, 'Jessica', 'Jones', 'jjones@ubc.ca', 'University of
British Columbia', 3.69);
(45636, 'Emily', 'In Paris', 'ouibaguette@parisu.fr', 'Paris
University', 4.00);
(55555, 'Michael', 'Michaels', 'mmichaels@ubc.ca', 'University of
British Columbia', 2.41);
(98452, 'Elizabeth', 'Queen', 'lizzie@ubristol.uk', 'University of
Bristol', 4.20);
(12345, 'seuss', 'doctor', 'heisadoctor@uvic.ca', 'University of
Victoria', 3.52);

INSERT INTO Application
VALUES (69421, 29/03/2021);
(11111, 08/09/2023);
(90002, 31/01/2024);
(67492, 02/04/2023);
```

```
(96477, 15/11/2021);

INSERT INTO ReferenceLetter
VALUES (29568, 69421, 'Bob Wales', 'whales@faculty.uwales.uk',
'University of Wales', 'Professor');
(40539, 11111, 'Monica Nguyen', 'monican@faculty.britu.uk' , 'Britain
University', 'Professor');
(65043, 90002, 'Tom Cat', 'ihatejerry@cat.hollywoodcol.us',
'Hollywood College', 'Housecat');
(85968, 67492, 'Sam Lee', 'leevemealone@stu.socal.us', 'University of
Southern California', 'Student');
(92045, 96477, 'George Bay', 'baywatch@dean.georgetown.us',
'Georgetown University', 'Dean');

INSERT INTO SelectionCriteria
VALUES (001, ‘Computer Science’, 3.20, 30000);
(002, ‘Computer Science’, 4.00, NULL);
(003, ‘Accounting’, 3.00, NULL);
(004, ‘History’, 2.40, 10000);
(011, ‘Political Science’, 4.0, NULL);

INSERT INTO ScholarshipCommittee
VALUES (123);
(234);
(345);
(456);
(567);

INSERT INTO Evaluates
VALUES (69421, 123, 'accepted');
(11111, 234, 'accepted');
(90002, 345, 'declined');
(67492, 456, 'accepted');
(96477, 567, 'declined');

INSERT INTO OneTime
VALUES (178904, 1001, 1000000);
(236049, 1001, 2000000);
(306946, 1003, 1000000);
```

```
(405069, 1003, 500000);
(507968, 1004, 500000);

INSERT INTO Renewable
VALUES (200358,1002, 5000, 09/04);
(200438,1002, 10000, 09/04);
(222118,1002, 20000, 17/02);
(500332,1001, 25000, 31/01);
(653453,1000, 1000000, 25/05);

INSERT INTO Donor
VALUES (1000);
(1001);
(1002);
(1003);
(1004);

INSERT INTO Superintendent
VALUES (02, 'James', 'Jameson');
(03, 'Jack', 'Johnson');
(04, 'Jon', 'Jackson');
(05, 'Jill', 'Jilly');
(06, 'Sarah', 'Sarahon');

INSERT INTO AppliesTo
VALUES (69421,200438,12348,1002);
(11111,405069,12348,1003);
(90002,178904,98452,1004);
(96477,178904,55555,1001);
(67492,222118,45636,1002);
```