Erika Francks and Tiffany Ta
Professor Zimbra
OMIS 105
03 June 2016

Global Social Benefit Institute Database

Business Description

The Miller Center for Social Entrepreneurship helps social entrepreneurs around the world scale their businesses. They aim to impact 1 billion lives by 2020. The Global Social Benefit Institute is nested under the Miller Center Social Entrepreneurship, which contains a vast amount of resources for social entrepreneurs, including a mentorship program with Silicon Valley leaders. The GSBI offers a fellowship for students in their junior year at Santa Clara University, called the Global Social Benefit Fellowship (GSBF), where students travel to a developing country to conduct an action research project for a social enterprise (SE). The Miller Center connects these fellows to an SE through an application process, as well as connecting impact investors with SEs and keeping track of these investments. As of now, the Global Social Benefit Institute has served over 731 social entrepreneurs since its inception in 2003.

Our database is based off the relationships and business rules we saw when looking at Miller Center documents and speaking with staff members there. Students are broken down into three types: applicants, fellows and alumni. Student applicants are linked to an SE through their application, and all students are linked to the country they will work in. Each student fellow is linked to a team, which includes the other student fellows traveling with them to do an action research project for a social enterprise. The SEs are linked to the country they operate in, as well as the impact investors who have invested in them and the impacts they have made in their work in social entrepreneurship. Both social enterprises and students have mentors, but the mentors are split out into Silicon Valley mentors for social enterprises and faculty mentors for students. A Silicon Valley mentor could potentially also be a faculty mentor. Silicon Valley mentors give social enterprises advice on how to scale their businesses, and faculty mentors help students prepare for their action research projects during the fellowship.

Our database is created to help streamline the operations of the Miller Center, by storing only useful data and highlighting relevant connections for specific users of the database. We have included data entity types for the social enterprises, students, mentors, impact investors, and impacts the social enterprises have had in the countries in which they operate. We do not have much numerical data in the database, but we will show relationships between the social enterprises and their mentors, the fellows assigned to them, and their investors. We include data from the application process of the enterprises and fellows from the time they are applicants all the way through until they have graduated the program and become alumni. We spoke directly with the Director of Operations and the Associate Director of Social Impact Assessment at the Miller Center, and they explained to us their need to show potential investors and the public the impact of their work with the social enterprises and students. We developed views of our database for the marketing department, impact investors, students, social enterprises and the fellowship director, each highlighting relevant relationships and information.

ER MODEL

Data Dictionary

IMPACT					
Name	Data Type	Constraints	Key	Description	Example Value
ImpactID	bigint	>0	PK	Unique identifier for an impact	12345
Impact Name	nvarchar(50)			Name of impact	Educational
Impact Description	nvarchar(max)			Description of the impact	Graduation rates have gone up
IMPACT_RECOGNITION					
Name	Data Type	Constraints	Key	Description	Example Value
Impact Recognition ID	bigint	>0	PK	Unique identifier for impact recognition	12345
Impact ID	bigint	>0	FK	Unique identifier for impact	12345
Social Enterprise ID	bigint	>0	FK	Unique identifier for each social enterprise	12345
Impact Date	datetime			Date and time the impact was recongized by the Miller Center	1/1/2001 12:30:00
SOCIAL_ENTERPRISE					
Name	Data Type	Constraints	Key	Description	Example Value
Social Enterprise ID	bigint	>0	PK	Unique identifier for each social enterprise	12345
Social Enterprise Name	nvarchar(50)			First and last name of employee in contact with the Miller Center	Shweta Mukherjee
Social Enterprise Title	nvarchar(50)			Name of the social enterprise	ONergy
Social Enterprise Email	nvarchar(50)			Contact email of the social enterprise	shweta@onergy.com
Social Enterprise Phone Number	char(12)			Business or contact person phone number	123456789012
Social Enterprise Gender	char(6)			Description of gender of contact person: Male, Female, or Other	female
Social Enterprise Website	nvarchar(50)			Website of the social enterprise	onergy.com
Social Enterprise Address	nvarchar(50)			Mailing address of the Social Enterprise	123 Maple St, Kolkata, India 12345
Social Enterprise Funds	decimal(10,2)	>0.0		Current amount of funding received the social enterprise	100000
INVESTMENT					
Name	Data Type	Constraints	Key	Description	Example Value
Investment ID	bigint	>0	PK	Unique identifier for investment	12345

Impact Investor ID	bigint	>0	FK	Unique Identifier for impact investors	12345
Social Enterprise ID	bigint	>0	FK	Unique identifier for each social enterprise	12345
Investment Amount	decimal(10,2)	>0.0		Amount impact investor is investing in the specific enterprise	100000
Investment Date	datetime			Date and Time investment recieved by enterprise	5/30/2016 13:00:00
IMPACT_INVESTOR					
Name	Data Type	Constraints	Key	Description	Example Value
Impact Investor ID	bigint	>0	PK	Unique identifier for Impact Investor	12345
Investor Name	nvarchar(50)			First and Last name of Investor	Bill Thompson
Investor Phone Number	char(12)			Phone number of the impact investor	123456789012
Investor Company	nvarchar(50)			Name of investment company	JP Morgan
COUNTRY					
Name	Data Type	Constraints	Key	Description	Example Value
Country ID	bigint	>0	PK	Unique identifier for country	12345
Country Name	nvarchar(50)			Name of the country	India
STUDENT					
Name	Data Type	Constraints	Key	Description	Example Value
Student ID	bigint	>0	PK	Unique identifier for student	12345
Student Name	nvarchar(50)			First and last name of student	Billy Bob
Student Gender	char(6)			Description of gender of student: Male, Female, or Other	female
Student Phone Number	char(12)			Phone number of student	12345678
Student School	nvarchar(50)			School that student is enrolled in at SCU	Arts and Sciences
Student Major	nvarchar(50)			Major(s) that the student has declared at SCU	Biology
Student Minor	nvarchar(50)			Minor(s) that the student has declared at SCU	OMIS
Student Grad Year	numeric(4)			Year that the student graduated or expects to graduate from SCU	2017
Student Email	nvarchar(50)			Email of the student	billybob@scu.edu

Student Type Fellow	nvarchar(1)	('Y', 'N')		Discriminator for student type Fellow	Y
Student Type Applicant	nvarchar(1)	('Y', 'N')		Discriminator for student type Applicant	N
Student Type Alumni	nvarchar(1)	('Y', 'N')		Discriminator for student type Alumni	Y
STUDENT_APPLICANT					
Name	Data Type	Constraints	Key	Description	Example Value
P Student ID	bigint	>0	PK, FK	Unique identifier for student applicant	12345
STUDENT_FELLOW					
Name	Data Type	Constraints	Key	Description	Example Value
F Student ID	bigint	>0	PK, FK	Unique identifier for student fellow	12345
Team ID	bigint	>0	FK	Unique identifier for a team of students	12345
TEAM					
Name	Data Type	Constraints	Key	Description	Example Value
Team ID	bigint	>0	PK	Unique identifier for a team of students	12345
Date Formed	datetime			Date and time the team of students was formed	5/30/2016 13:00:00
STUDENT_ALUMNI					
Name	Data Type	Constraints	Key	Description	Example Value
L Student ID	bigint	>0	PK	Unique identifier for a student alumni	12345
Alumni Company	nvarchar(50)			Company the alumni currently works for	Oracle
Alumni Job Title	nvarchar(50)			Current job title of the alumni	CFO
Alumni Award	nvarchar(100)			Any awards the alumni has received since being in GSBF	Fulbright Scholarship
MENTOR					
Name	Data Type	Constraints	Key	Description	Example Value
Mentor ID	bigint	>0	PK	Unique identifier for a mentor	12345
Mentor Name	nvarchar(50)			First and Last Name of Mentor	Fred Weasley
Contact Info	nvarchar(50)			Prefered mode of contact: email, address, phone number etc.	

Mentor Status	nvarchar(9)			Status of mentor, either active or inactive	active
Mentor Expertise	nvarchar(75)			Field or expertise of mentor that they would mentor in	Water Sanitation
Mentor Type Faculty	char(1)	('Y', 'N')		Discriminator for mentor type Faculty(F)	Y
Mentor Type Silicon Valley	char(1)	('Y', 'N')		Discriminator for mentor type Silicon Valley (‘S)	Y
MENTOR_REASON					
Name	Data Type	Constraints	Key	Description	Example Value
Reason ID	bigint	>0	PK	Unique identifier for a mentor's reason	12345
Reason Name	nvarchar(50)			The name of the reason the mentor decided to become a mentor for GSBI	Friend of the miller center
Reason Description	nvarchar(max)			Explanation of the reason why the mentor chose to become a mentor for GSBI	I was referred by Spencer in the Miller Center
FACULTY_MENTOR					
Name	Data Type	Constraints	Key	Description	Example Value
F Mentor ID	bigint	>0	PK, FK	Unique identifier for a faculty mentor	12345
Faculty Department	nvarchar(50)			Department at SCU that the faculty mentor belongs to	English
SILICON_VALLEY_MENTOR					
Name	Data Type	Constraints	Key	Description	Example Value
S Mentor ID	bigint	>0	PK, FK	Uniqe identifier for Silicon Valley Mentor	12345
Silicon Valley Profession	nvarchar(50)			Silicon Valley Mentor's Job Title	CEO
APPLICATION					
Name	Data Type	Constraints	Key	Description	Example Value
Application ID	bigint	>0	PK	Unique identifier for application	12345
Social Enterprise ID	bigint	>0	FK	Unique identifier for social enterprise	
P Student ID	bigint	>0	FK	Unique identifier for student applicant	
Application Date	datetime			Date and Time the application was submitted	5/30/2016 13:00:00

Application Status	nvarchar(7)			Status of the Application either submitted or approved	submitted
Honors Program	nvarchar(1)			Whether or not student is involved in the honors program. 'Y' or 'N'	Y
Schedule Conflicts	nvarchar(1)			Whether or not student is has scheduling conflicts'Y' or 'N'	N
Spanish	nvarchar(1)			Whether or not student can speak Spanish fluently. 'Y' or 'N'	Y
COUNTRY_SOCIAL_ENTERPRISE					
Name	Data Type	Constraints	Key	Description	Example Value
Country Social Enterprise ID	bigint	>0	PK	Unique identifier for a social enterprise located in a country	12345
Country ID	bigint	>0	FK	Unique identifier for a country	12345
Social Enterprise ID	bigint	>0	FK	Unique identifier for a social enterprise	12345
STUDENT_COUNTRY					
Name	Data Type	Constraints	Key	Description	Example Value
Student Country ID	bigint	>0	PK	Unique identifier for a student working in a country	12345
Student ID	bigint	>0	FK	Unique identifier for a student	12345
Country ID	bigint	>0	FK	Unique identifier for a country	12345
MENTOR_MENTOR_REASON					
Name	Data Type	Constraints	Key	Description	Example Value
Mentor Mentor Reason ID	bigint	>0	PK	Unique identifier for connecting a mentor to a reason	12345
Mentor ID	bigint	>0	FK	Unique identifier for a mentor	12345
Reason ID	bigint	>0	FK	Unique identifier for a reason	12345
SILICON_VALLEY_MENTOR_SOCIAL_ENTERPRISE					
Name	Data Type	Constraints	Key	Description	Example Value
SV Mentor Social Enterprise ID	bigint	>0	PK	Unique identifier for connecting a silicon valley mentor to a social enterprise	12345
S Mentor ID	bigint	>0	FK	Unique identifier for a mentor	12345
Social Enterprise ID	bigint	>0	FK	Unique identifier for a social enterprise	12345

FACULTY_MENTOR_STUDENT					
Name	Data Type	Constraints	Key	Description	Example Value
Faculty Mentor Student ID	bigint	>0	PK	Unique identifier for connecting a faculty mentor to a student	12345
F Mentor ID	bigint	>0	FK	Unique identifier for a mentor	12345
Student ID	bigint	>0	FK	Unique identifier for a student	12345

Logical Schema

Create Table Statements

IMPACT	**ImpactID**	ImpactName	ImpactDescription		
IMPACT_RECOGNITION	**ImpactRecognitionID**	***ImpactID***	***SocialEnterpriseID***	ImpactDate	
SOCIAL_ENTERPRISE	**SocialEnterpriseID**	SocialEnterpriseName	SocialEnterpriseTitle	SocialEnterpriseEmail	SocialEnterprisePhoneNumber
	SocialEnterpriseGender	SocialEnterpriseWebsit	SocialEnterpriseAddress	SocialEnterpriseFunds	
COUNTRY_SOCIAL_ENTERPRISE	**CountrySocialEnterpriseID**	***SocialEnterpriseID***	***CountryID***		
INVESTMENT	**InvestmentID**	***SocialEnterpriseID***	***ImpactInvestorID***	InvestmentAmount	InvestmentDate
IMPACT_INVESTOR	**ImpactInvestorID**	ImpactInvestorName	ImpactInvestorPhoneNumbe	ImpactInvestorCompany	
COUNTRY	**CountryID**	CountryName			
STUDENT_COUNTRY	**StudentCountryID**	***StudentID***	***CountryID***		
APPLICATION	**ApplicationID**	***SocialEnterpriseID***	***PStudentID***	ApplicationDate	ApplicationStatus
	ScheduleConflicts	**HonorsProgram**	**Spanish**		
STUDENT	**StudentID**	StudentName	StudentGender	StudentPhoneNumber	StudentSchool
	StudentMajor	StudentMinor	StudentGradYear	StudentEmail	**StudentTypeFellow**
		StudentTypeApplicant	**StudentTypeAlumni**		
STUDENT_FELLOW	***FStudentID***	***TeamID***			
TEAM	**TeamID**	DateFormed			
STUDENT_APPLICANT	***PStudentID***				
STUDENT_ALUMNI	***LStudentID***	AlumniCompany	AlumniJobTitle	AlumniAward	
MENTOR	**MentorID**	MentorName	MentorContactInfo	MentorStatus	MentorExpertise
	MentorTypeSiliconValley	**MentorTypeFaculty**			
MENTOR_MENTOR_REASON	**MentorMentorReasonID**	***MentorID***	***ReasonID***		
MENTOR_REASON	**ReasonID**	ReasonName	ReasonDescription		
FACULTY_MENTOR	***FMentorID***	FacultyDepartment			
SILICON_VALLEY_MENTOR	***SMentorID***	SiliconValleyProfession			
SILICON_VALLEY_MENTOR_SOCIAL_ENTERPRIS	**SVMentorSocialEnterpriseID**	***SMentorID***	***SocialEnterpriseID***		
FACULTY_MENTOR_STUDENT	**FacultyMentorStudentID**	***FMentorID***	***StudentID***		

Impact Investor

```
CREATE TABLE ImpactInvestor_T
(ImpactInvestorID               INT NOT NULL IDENTITY(1,1)
                                        CHECK (ImpactInvestorID > 0),
InvestorName                    NVARCHAR(50),
InvestorPhoneNumber             CHAR(12),
InvestorCompany                 NVARCHAR(50),
CONSTRAINT ImpactInvestor_PK PRIMARY KEY(ImpactInvestorID));
```

Impact

```
CREATE TABLE Impact_T
(ImpactID                       INT NOT NULL IDENTITY(1,1)
                                        CHECK (ImpactID > 0),
ImpactName                      NVARCHAR(50),
ImpactDescription               NVARCHAR(max),
CONSTRAINT Impact_PK PRIMARY KEY(ImpactID));
```

Social Enterprise

```
CREATE TABLE SocialEnterprise_T
(SocialEnterpriseID             INT NOT NULL IDENTITY(1,1)
                                        CHECK (SocialEnterpriseID > 0),
SocialEnterpriseName            NVARCHAR(50),
SocialEnterpriseTitle           NVARCHAR(50),
SocialEnterpriseEmail           NVARCHAR(50),
SocialEnterprisePhoneNumber     CHAR(12),
SocialEnterpriseGender          CHAR(6),
SocialEnterpriseWebsite         NVARCHAR(50),
SocialEnterpriseAddress         NVARCHAR(50),
SocialEnterpriseFunds           DECIMAL(10,2),
CONSTRAINT SocialEnterprise_PK PRIMARY KEY(SocialEnterpriseID));
```

Country

```
CREATE TABLE Country_T
(CountryID                      INT NOT NULL IDENTITY(1,1)
                                        CHECK (CountryID > 0),
CountryName                     NVARCHAR(50),
CONSTRAINT Country_PK PRIMARY KEY(CountryID));
```

Student

```
CREATE TABLE Student_T
(StudentID                      INT NOT NULL IDENTITY(1,1)
                                        CHECK (StudentID > 0),
StudentName                     NVARCHAR(50) NOT NULL,
StudentGender                   CHAR(6),
                                        CHECK (StudentGender IN ('Male', 'Female', 'Other')),
StudentPhoneNumber              CHAR(12),
StudentSchool                   NVARCHAR(50),
StudentMajor                    NVARCHAR(50),
StudentMinor                    NVARCHAR(50),
StudentGradYear                 NUMERIC(4),
```

```
StudentEmail                    NVARCHAR(50),
StudentTypeFellow               NVARCHAR(1)
                                        CHECK(StudentTypeFellow IN ('Y', 'N')) NOT NULL,
StudentTypeApplicant            NVARCHAR(1)
                                        CHECK(StudentTypeApplicant IN ('Y', 'N')) NOT NULL,
StudentTypeAlumni               NVARCHAR(1)
                                        CHECK(StudentTypeAlumni IN ('Y', 'N')) NOT NULL,
CONSTRAINT Student_PK PRIMARY KEY(StudentID));
```

Student Applicant

```
CREATE TABLE StudentApplicant_T
(PStudentID                     INT NOT NULL
                                        CHECK(PStudentID > 0),
CONSTRAINT PStudent_PK PRIMARY KEY(PStudentID),
CONSTRAINT StudentApplicant_FK FOREIGN KEY(PStudentID) REFERENCES
Student_T(StudentID));
```

Team

```
CREATE TABLE Team_T
(TeamID                         INT NOT NULL IDENTITY(1,1)
                                        CHECK (TeamID > 0),
DateFormed                      datetime DEFAULT GETDATE(),
CONSTRAINT Team_PK PRIMARY KEY(TeamID));
```

Student Fellow

```
CREATE TABLE StudentFellow_T
(FStudentID                     INT NOT NULL,
TeamID                          INT NOT NULL,
CONSTRAINT FStudent_PK PRIMARY KEY(FStudentID),
CONSTRAINT StudentFellow_FK1 FOREIGN KEY(FStudentID) REFERENCES
Student_T(StudentID),
CONSTRAINT StudentFellow_FK2 FOREIGN KEY(TeamID) REFERENCES
Team_T(TeamID));
```

Student Alumni

```
CREATE TABLE StudentAlumni_T
(LStudentID                     INT NOT NULL,
AlumniCompany                   NVARCHAR(50),
AlumniJobTitle                  NVARCHAR(50),
AlumniAward                     NVARCHAR(100),
CONSTRAINT LStudent_PK PRIMARY KEY(LStudentID),
CONSTRAINT StudentAlumni_FK FOREIGN KEY(LStudentID) REFERENCES
Student_T(StudentID));
```

Mentor

```
CREATE TABLE Mentor_T
(MentorID                       INT NOT NULL IDENTITY(1,1)
                                        CHECK (MentorID > 0),
MentorName                      NVARCHAR(50),
ContactInfo                     NVARCHAR(50),
MentorStatus                    NVARCHAR(9),
```

```
MentorExpertise                 NVARCHAR(75),
MentorTypeFaculty               CHAR(1)
                                        CHECK(MentorTypeFaculty IN ('Y', 'N')) NOT NULL,
MentorTypeSiliconValley         CHAR(1)
                                        CHECK(MentorTypeSiliconValley IN ('Y', 'N')) NOT NULL,
CONSTRAINT Mentor_PK PRIMARY KEY(MentorID));
```

Faculty Mentor

```
CREATE TABLE FacultyMentor_T
(FMentorID                      INT NOT NULL,
FacultyDepartment               NVARCHAR(50),
CONSTRAINT FacultyMentor_PK PRIMARY KEY(FMentorID),
CONSTRAINT FacultyMentor_FK FOREIGN KEY(FMentorID) REFERENCES
Mentor_T(MentorID));
```

Silicon Valley Mentor

```
CREATE TABLE SiliconValleyMentor_T
(SMentorID                      INT NOT NULL,
SiliconValleyProfession         NVARCHAR(50),
CONSTRAINT SiliconValleyMentor_PK PRIMARY KEY(SMentorID),
CONSTRAINT SiliconValleyMentor_FK FOREIGN KEY(SMentorID) REFERENCES
Mentor_T(MentorID));
```

Application

```
CREATE TABLE Application_T
(ApplicationID                  INT NOT NULL IDENTITY(1,1)
CHECK (ApplicationID > 0),
SocialEnterpriseID              INT NOT NULL,
PStudentID                      INT NOT NULL,
ApplicationDate                 datetime DEFAULT GETDATE(),
ApplicationStatus               NVARCHAR(7),
HonorsProgram                   NVARCHAR(1) NOT NULL,
ScheduleConflict                NVARCHAR(1),
Spanish                         NVARCHAR(1) NOT NULL,
CONSTRAINT Application_PK PRIMARY KEY(ApplicationID),
CONSTRAINT Application_FK1 FOREIGN KEY(PStudentID) REFERENCES
StudentApplicant_T(PStudentID),
CONSTRAINT Application_F2K FOREIGN KEY(SocialEnterpriseID) REFERENCES
SocialEnterprise_T(SocialEnterpriseID));
```

Impact Recognition

```
CREATE TABLE ImpactRecognition_T
(ImpactRecognitionID            INT NOT NULL IDENTITY(1,1)
                                        CHECK (ImpactRecognitionID > 0),
ImpactID                        INT NOT NULL,
SocialEnterpriseID              INT NOT NULL,
CONSTRAINT ImpactRecognition_PK PRIMARY KEY(ImpactRecognitionID),
CONSTRAINT ImpactRecognition_FK1 FOREIGN KEY(ImpactID) REFERENCES
Impact_T(ImpactID),
CONSTRAINT ImpactRecognition_FK2 FOREIGN KEY(SocialEnterpriseID) REFERENCES
SocialEnterprise_T(SocialEnterpriseID));
```

Investment

```
CREATE TABLE Investment_T
(InvestmentID                   INT NOT NULL IDENTITY(1,1)
                                        CHECK (InvestmentID > 0),
ImpactInvestorID                INT NOT NULL,
SocialEnterpriseID              INT NOT NULL,
InvestmentAmount                DECIMAL(10,2),
InvestmentDate                  datetime DEFAULT GETDATE(),
CONSTRAINT Investment_PK PRIMARY KEY(InvestmentID),
CONSTRAINT Investment_FK1 FOREIGN KEY(ImpactInvestorID) REFERENCES
ImpactInvestor_T(ImpactInvestorID),
CONSTRAINT Investment_FK2 FOREIGN KEY(SocialEnterpriseID) REFERENCES
SocialEnterprise_T(SocialEnterpriseID));
```

Mentor Reason

```
CREATE TABLE MentorReason_T
(MentorReasonID                 INT NOT NULL IDENTITY(1,1)
                                        CHECK (MentorReasonID > 0),
ReasonName                      NVARCHAR(50),
ReasonDescription               NVARCHAR(max),
CONSTRAINT MentorReason_PK PRIMARY KEY(MentorReasonID));
```

Country Social Enterprise

```
CREATE TABLE CountrySocialEnterprise_T
(CountrySocialEnterpriseID      INT NOT NULL IDENTITY(1,1)
                                        CHECK (CountrySocialEnterpriseID > 0),
StudentID                       INT NOT NULL,
CountryID                       INT NOT NULL,
CONSTRAINT CountrySocialEnterprise_PK PRIMARY KEY(CountrySocialEnterpriseID),
CONSTRAINT CountrySocialEnterprise_FK1 FOREIGN KEY(StudentID) REFERENCES
Student_T(StudentID),
CONSTRAINT CountrySocialEnterprise_FK2 FOREIGN KEY(CountryID) REFERENCES
Country_T(CountryID));
```

Student Country

```
CREATE TABLE StudentCountry_T
(StudentCountryID               INT NOT NULL IDENTITY(1,1)
                                        CHECK (StudentCountryID > 0),
StudentID                       INT NOT NULL,
CountryID                       INT NOT NULL,
CONSTRAINT StudentCountry_PK PRIMARY KEY(StudentCountryID),
CONSTRAINT StudentCountry_FK1 FOREIGN KEY(StudentID) REFERENCES
Student_T(StudentID),
CONSTRAINT StudentCountry_FK2 FOREIGN KEY(CountryID) REFERENCES
Country_T(CountryID));
```

Mentor Mentor Reason

```
CREATE TABLE MentorMentorReason_T
(MentorMentorReasonID           INT NOT NULL IDENTITY(1,1)
                                        CHECK (MentorMentorReasonID > 0),
```

```
MentorID                        INT NOT NULL,
ReasonID                        INT NOT NULL,
CONSTRAINT MentorMentorReason_PK PRIMARY KEY(MentorMentorReasonID),
CONSTRAINT MentorMentorReason_FK1 FOREIGN KEY(MentorID) REFERENCES
Mentor_T(MentorID),
CONSTRAINT MentorMentorReason_FK2 FOREIGN KEY(ReasonID) REFERENCES
MentorReason_T(MentorReasonID));
```

Silicon Valley Mentor Social Enterprise

```
CREATE TABLE SiliconValleyMentorSocialEnterprise_T
(SiliconValleyMentorSocialEnterpriseID   INT NOT NULL IDENTITY(1,1)
                                            CHECK (SiliconValleyMentorSocialEnterpriseID > 0),
SMentorID                               INT NOT NULL,
SocialEnterpriseID                      INT NOT NULL,
CONSTRAINT SiliconValleyMentorSocialEnterprise_PK PRIMARY
KEY(SiliconValleyMentorSocialEnterpriseID),
CONSTRAINT SiliconValleyMentorSocialEnterprise_FK1 FOREIGN KEY(SMentorID) REFERENCES
SiliconValleyMentor_T(SMentorID),
CONSTRAINT SiliconValleyMentorSocialEnterprise_FK2 FOREIGN KEY(SocialEnterpriseID)
REFERENCES
SocialEnterprise_T(SocialEnterpriseID));
```

Faculty Mentor Student

```
CREATE TABLE FacultyMentorStudent_T
(FacultyMentorStudentID                 INT NOT NULL IDENTITY(1,1)
                                            CHECK (FacultyMentorStudentID > 0),
FMentorID                               INT NOT NULL,
SocialEnterpriseID                      INT NOT NULL,
CONSTRAINT FacultyMentorStudent_PK PRIMARY KEY(FacultyMentorStudentID),
CONSTRAINT FacultyMentorStudent_FK1 FOREIGN KEY(FMentorID) REFERENCES
FacultyMentor_T(FMentorID),
CONSTRAINT FacultyMentorStudent_FK2 FOREIGN KEY(SocialEnterpriseID) REFERENCES
SocialEnterprise_T(SocialEnterpriseID));
```

Views

1. **Impact Investor View**

Create View ImpactInvestor as

Select SocialEnterprise_T.SocialEnterpriseTitle, Country_T.CountryName,
Impact_T.ImpactName, Impact_T.ImpactDescription,
sum(SocialEnterprise_T.SocialEnterpriseFunds) as TotalFundsRaised,
sum(Investment_T.InvestmentAmount) as TotalInvestmentDollars,
count(distinct(Investment_T.InvestmentID)) as TotalNumberofInvestments,
sum(Investment_T.InvestmentAmount)/count(Investment_T.InvestmentID) as
AvgInvestmentAmount

From Investment_T, SocialEnterprise_T, Impact_T, ImpactRecognition_T, Country_T,
CountrySocialEnterprise_T

Where Investment_T.SocialEnterpriseID = SocialEnterprise_T.SocialEnterpriseID
and Impact_T.ImpactID = ImpactRecognition_T.ImpactID
and ImpactRecognition_T.SocialEnterpriseID = SocialEnterprise_T.SocialEnterpriseID
and Country_T.CountryID = CountrySocialEnterprise_T.CountryID
and CountrySocialEnterprise_T.SocialEnterpriseID = SocialEnterprise_T.SocialEnterpriseID

Group by SocialEnterprise_T.SocialEnterpriseID, SocialEnterprise_T.SocialEnterpriseTitle,
Country_T.CountryName, Impact_T.ImpactName, Impact_T.ImpactDescription

We created this view for impact investors looking to invest in a social enterprise through the Miller Center. Their view shows the name of the social enterprise and the impact they have had, the amount of funding each social enterprise has raised, the total investment dollars they have received so far, the total number of investments made to them, and the average amount invested in them. This is valuable for the Miller Center because if an impact investor were to ask, “On average, how much are people investing in ____ social enterprise?”, the view would display this information and that could inform their decision to invest. This view also hides all other data that the Miller Center might not want the investors to have access to.

2. **Marketing View**

Create View Marketing as

Select Country_T.countryname, Impact_T.impactdescription, count(Student_T.StudentID)
as NumStudentsVisited, SocialEnterprise_T.SocialEnterpriseTitle,
SocialEnterprise_T.SocialEnterpriseFunds

```
From Country_T, Impact_T, SocialEnterprise_T, ImpactRecognition_T,
Student_T, CountrySocialEnterprise_T, StudentCountry_T

Where Country_T.CountryID = CountrySocialEnterprise_T.CountryID
and CountrySocialEnterprise_T.SocialEnterpriseID = SocialEnterprise_T.SocialEnterpriseID
and Impact_T.ImpactID = ImpactRecognition_T.ImpactID
and ImpactRecognition_T.SocialEnterpriseID = SocialEnterprise_T.SocialEnterpriseID
and StudentCountry_T.countryid = Country_T.countryid
and StudentCountry_T.studentid = Student_T.studentid

GROUP BY Country_T.countryname, Impact_T.impactdescription,
SocialEnterprise_T.SocialEnterpriseTitle, SocialEnterprise_T.SocialEnterpriseFunds
```

We created this view for the marketing department of the Miller Center so that they can show potential investors, partners, students and mentors the accomplishments of the GSBI program. This view shows the various countries that social enterprises work in, a description what impacts have been made in these countries, and how many students have visited these countries on the fellowship. This is a good marketing tool for GSBI because the Director of Operations told us that they would like to be able to show how many students have been to a particular place on the program and what sort of impact is being made in that place.

3. Student View

```
CREATE VIEW Student as

SELECT Team_T.TeamID, Student_T.StudentName, Mentor_T.MentorName,
Mentor_T.mentorcontactinfo

FROM StudentFellow_T, Team_T, Student_T,
FacultyMentor_T, Mentor_T, FacultyMentorStudent_T

WHERE StudentFellow_T.Teamid = Team_T.Teamid
and Student_T.studentID = StudentFellow_T.fstudentid
and FacultyMentorStudent_T.StudentID = Student_T.StudentID
and Mentor_T.mentorid = FacultyMentor_T.fmentorid
and FacultyMentorStudent_T.FMentorID = FacultyMentor_T.FMentorID

GROUP BY Student_T.StudentName, Team_T.TeamID, Mentor_T.MentorName,
 Mentor_T.mentorcontactinfo
```

We created this view for students in the program so they can keep track of their application, their team, and their mentor's name and contact info. This view shows the status of their application (submitted or approved), all the members of each team of fellows, the faculty mentor for each fellow and their contact information so the student can get in touch with them to start planning their research. This is useful because during the GSBF class in the spring, the fellow has to reach out to their own mentor to gain insight into the organization and how they should approach their research project. This view gives the student only the information they would need to get started on their project, and no unnecessary fields.This view also hides all other data that the Miller Center might not want the students to have access to.

4. Social Enterprise View

```
CREATE VIEW SocialEnterprise as

SELECT ImpactInvestor_T.ImpactInvestorName,
sum(investment_t.investmentamount) as TotalInvestmentAmount,
socialenterprise_t.socialenterprisetitle,
Impact_T.ImpactName

FROM Impact_T, ImpactInvestor_T, SocialEnterprise_T,
ImpactRecognition_T, Investment_T

WHERE impactinvestor_t.impactinvestorid = investment_t.impactinvestorid
and SocialEnterprise_T.socialenterpriseid = ImpactRecognition_T.SocialEnterpriseID
and ImpactRecognition_T.ImpactID = Impact_T.ImpactID
and investment_t.socialenterpriseid = SocialEnterprise_T.socialenterpriseid

GROUP BY ImpactInvestor_T.ImpactInvestorName, socialenterprise_t.socialenterprisetitle,
Impact_T.ImpactName
```

We created this view for social enterprises so that they can strategize how to get more investments. We are showing them impact investors names, how much those investors have invested in SEs, which SEs they have invested in, and which impacts those SEs have accomplished. This will help the SEs because if they are looking for someone to invest in their enterprise that focuses on water, they can look at the different impact investors to see which ones invest in enterprises with water impacts and how much they have invested. They can choose to reach out to those investors who have put a lot of money towards similar enterprises

to themselves. This view also hides all other data that the Miller Center might not want the SEs to have access to.

5. Fellowship Director View

```
CREATE VIEW FellowshipDirector as

SELECT student_t.studentname, student_t.studentmajor,
socialenterprise_t.socialenterprisetitle,
mentor_t.mentorname, mentor_t.mentorexpertise, country_t.countryname

FROM Student_T, SocialEnterprise_T, Mentor_T, country_t, StudentCountry_T,
CountrySocialEnterprise_T, FacultyMentor_T, FacultyMentorStudent_T

WHERE
student_t.studentid = studentcountry_t.studentid
and studentcountry_t.countryid = country_t.countryid
and SocialEnterprise_t.SocialEnterpriseID = CountrySocialEnterprise_T.SocialEnterpriseID
and Country_T.CountryID = CountrySocialEnterprise_T.SocialEnterpriseID
and FacultyMentorStudent_T.StudentID = Student_T.StudentID
and Mentor_T.mentorid = FacultyMentor_T.fmentorid
and FacultyMentorStudent_T.FMentorID = FacultyMentor_T.FMentorID
```

We created this view for the director of the fellowship. This view shows him student's names, majors, the social enterprise they're working with, their mentor name and expertise, and the country name. This view is meant to help the fellowship director pair mentors with students by matching expertise to student majors and the social enterprises they're working for. The fellowship director also has access to which countries the students are going to take that into consideration when matching mentors with students. A certain expertise could be more relevant in certain countries. Looking at the country name also helps the director keep track of teams and where students are headed.