‘Mentor’ Database Project

Professor Thompson
ITSC 3160

Michael Dix
Erik Ellis
Aaron Oldfield
Spencer Scaggs
Tyler Gandy

12/9/18

Summary

The goal of this database project was to design a database that manages data for mentors who are assigned to computer science and information technology science programs at the University of North Carolina at Charlotte. This data consists of information for a Person(student, mentor), different events that are held for students, and the ratings of these events given by either students and mentors. In this project, MySQL was used to visualize the tables, schemas, and database that we have created.

Business Rules and Any Changes or Assumptions

1) Person – Student/Mentor is a generalization-specialization relationship (superclass/subclass)
2) A student is assigned to one and only one mentor per semester but can end up having multiple mentors, mentors advise one or more students over time (many to many relationship)
3) Persons sign up for one or more events, events can have one or more Persons sign up to attend. We want to keep track of the registration for the event and we want to know if the Person attended the event. Some consideration may be given for whether or not the Student attended with the mentor but the faculty committee wasn't sure how to handle this.
4) There needs to be a way to keep up with evaluations for the events –there is an evaluation form that has the following questions: the form is tied to an event when attended by a person and mentor (should have both of these) and it also contains the Person_id – this is tied to the Register/Attend file (Persons have to complete the form only if they attend the event – otherwise it would not make sense).
    1) On a scale of 1 to 5, was the event useful to you?
    2) On a scale of 1 to 5, would you recommend this event to another person?
    3) On a scale of 1 to 5, was the event relative to your major?
    4) Other? To be decided, faculty committee wasn't sure what else to ask

5) Faculty need to receive a report listing their advisees and the events that the advisees attended with their mentors. Faculty also would like a list of assignments each semester.
6) Students need to see a list of the events they have attended with their mentors.
7) Mentors need to see a list of their students and a list of the events attended.

8) Faculty want to see the event ratings to determine which events seem to be valuable. Ratings can come from Mentors and Students and a comparison may be interesting.

9) A student cannot be a mentor and a mentor cannot be a student.

10) Student-mentor pairings are based on semester and can be repeated.

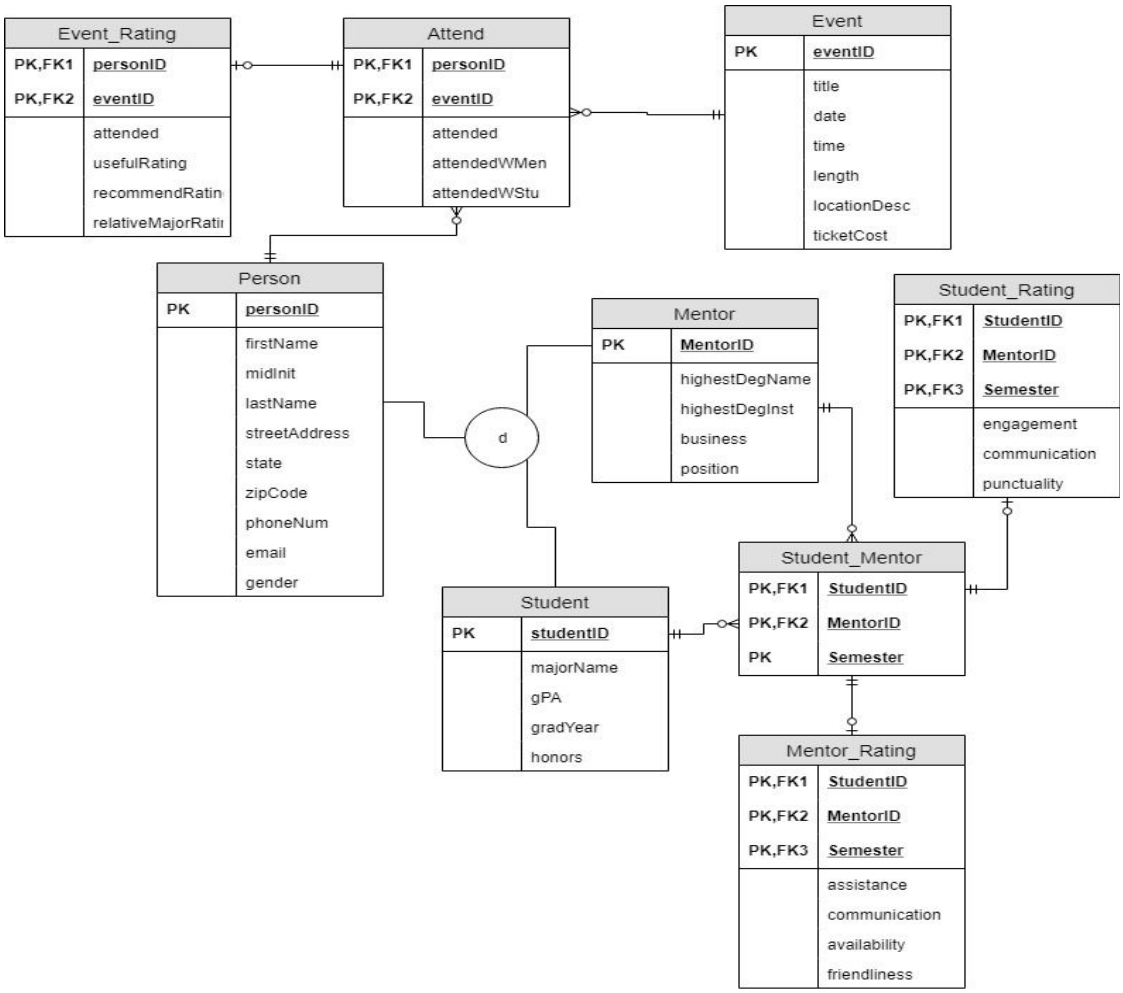
11) Events can be attended by students or mentors alone or together.

12) A person must attend an event in order to rate it (boolean field designated within table).

13) Students and mentors can both exist without the assignment of the student-mentor relationship.

14) Students and mentors are able to rate each other once per semester pairing.

EERD



Data Dictionary (Meta Data)

Table	Field Name	Data Type	PK	FK	Not Null	Description
Person	personID	int	Y		Y	Person's ID
Person	firstName	varchar(63)			Y	Person's first name
Person	lastName	varchar(63)			Y	Person's last name
Person	streetAddress	varchar(255)				Street address of person
Person	state	varchar(127)				State of person
Person	zipCode	int				Area code of person
Person	phoneNum	varchar(31)				Phone number of person
Person	email	varchar(255)			Y	Email of person
Person	gender	varchar(63)				Gender of person
Student	studentID	int			Y	References personID
Student	majorName	varchar(255)				Major of student
Student	gPA	int				Student's GPA
Student	gradYear	int				Student's graduation year
Student	honors	varchar(8)				If student is honors or not
Mentor	mentorID	int			Y	References personID
Mentor	highestDegName	varchar(255)				Highest degree of mentor
Mentor	highestDegInst	varchar(255)				
Mentor	business	varchar(255)			Y	Business of mentor
Mentor	jobTitle	varchar(255)				Job title of mentor

Table	Field Name	Data Type	PK	FK	Not Null	Description
Student_ Mentor	studentID	int		Y	Y	References PersonID
Student_ Mentor	mentorID	int		Y	Y	References PersonID
Student_ Mentor	semester	varchar(127)			Y	Semester of mentor assignment
Student_ Rating	studentID	int		Y	Y	References PersonID
Student_ Rating	mentorID	int		Y	Y	References PersonID
Student_ Rating	semester	varchar(127)			Y	Semester of mentor assignment
Student_ Rating	engagementSc ore	int				Rating of student engagement
Student_ Rating	communicatio nScore	int				Rating of communication between student and mentor
Student_ Rating	punctualitySc ore	int				Rating of student punctuality
Mentor_ Rating	studentID	int		Y	Y	References PersonID
Mentor_ Rating	mentorID	int		Y	Y	References PersonID
Mentor_ Rating	semester	varchar(127)			Y	Semester of mentor assignment
Mentor_ Rating	communicatio nScore	int				Rating for mentor communication
Mentor_ Rating	availabilitySc ore	int				Rating for mentor availability

Mentor_ Rating	friendlinessSc ore	int				Rating of mentor friendliness

Table	FieldName	Data Type	PK	FK	Not Null	Description
Campus Event	eventID	int	Y		Y	Event ID
Campus Event	title	varchar(255)			Y	Title of campus event
Campus Event	eventDate	date			Y	Date of event
Campus Event	eventTime	time			Y	Time of event
Campus Event	eventLength	float				Length of event
Campus Event	eventLocation	varchar(255)				Location of event
Campus Event	ticketCost	varchar(127)				Cost of ticket for event
Attend	personID	int		Y	Y	References PersonID
Attend	eventID	int		Y	Y	References eventID
Attend	attended	varchar(8)			Y	If event has been attended
Attend	attendedWMe nt	varchar(8)				If event has been attended with mentor
Attend	attendedWStu	varchar(8)				If event has been attended with student
Event_R ating	personID	int		Y	Y	References PersonID
Event_R ating	eventID	int		Y	Y	References EventID
Event_R ating	attended	varchar(8)			Y	Determines if event was attended

Event_Rating	usefulRating	varchar(127)				Rating of usefulness of event
Event_Rating	recommendRating	varchar(127)				Rating of how much the event would be recommended
Event_Rating	relativeMajorRating	varchar(127)				Rating of the event's relativity to the student's major

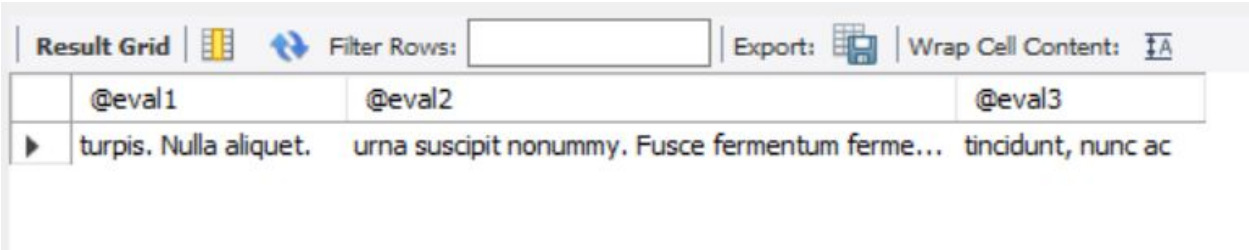
Stored Procedure

A stored procedure is similar to Java method as it is a set of SQL statements that is saved to the DMS. These stored procedures can be called for later use. Stored procedures are executed far more than they are written.

```
use  mentorDB;

DROP PROCEDURE IF EXISTS evaluations;

DELIMITER //

CREATE PROCEDURE evaluations(

IN id1 INT,

OUT eval1 varchar(127),

OUT eval2 varchar(127),

OUT eval3 varchar(127) )

BEGIN

SELECT usefulRating FROM Event_Rating

WHERE Event_Rating.personID = id1 INTO eval1;
```

```
SELECT recommendRating FROM Event_Rating

WHERE Event_Rating.personID = id1 INTO eval2;

SELECT relativeMajorRating FROM Event_Rating

WHERE Event_Rating.personID = id1 INTO eval3;

END //

DELIMITER ;

CALL evaluations(12, @eval1, @eval2, @eval3);

SELECT @eval1, @eval2, @eval3;
```

Here are the results of the stored procedure. This would be the student's evaluations of the events that they have attended:



Trigger

A trigger is a form of stored procedure that occurs when data modification events occur (INSERT, UPDATE, etc.). They are different from stored procedures because stored procedures have to be called in order to function.

Example of Delete and Update

Update: Updates Campus Event table and sets title of event for the event with ID 13 to "CompSci Employer Meet and Greet"

```
UPDATE mentordb.CampusEvent
SET title = 'CompSci Employer Meet and Greet'
WHERE eventID = 13;
```

DELETE: Deletes row from Person table where personID is null

```
DELETE FROM person
WHERE personID = null;
```

SQL for 3 Reports

Report 1

Report Faculty - Advisees Events

Faculty need to receive a report listing their advisees and the events that the advisees attended with their mentors. Faculty also would like a list of assignments each semester.

```
SELECT  persontID, eventID
FROM Attend
JOIN Student ON personID = student.ID;
```

Report 2

Report Student - Student Attended Events

Students need to see a list of the events they have attended with their mentors

```
SELECT eventID
FROM Attend
WHERE attended = “Yes”;
```

Report 3

Report Mentor - Mentor’s Students and Events

Mentors need to see a list of their students and a list of the events attended.

```
SELECT studentID, firstName, lastName, eventID, title, eventTitle
FROM Student, CampusEvent
JOIN Student_Mentor ON Student_Mentor.studentID = studentID;
```