

Syracuse University

Tutoring

Project Implementation Report

Prepared for: Syracuse University Tutoring

Prepared by: Jack Dolitsky & Avery Curcio

Project Summary

Background

Syracuse University Tutors is a tutoring program designed for Syracuse University students to connect with fellow classmates to tutor them in select subjects. Whether it is tutoring in campus buildings or tutoring in dorms, Syracuse University Tutors wants to sort students and place them at their desired locations with tutors for their subject area of need. Each student's request for a tutor combines many different elements that need to be taken into consideration to connect them with the right tutor. Students provide their student identification number, their name, email, address, major, the subject they need tutoring in, the major they are in, their gpa and grade level. This information is then cross-listed with the tutor data which includes; tutor identification number, tutor name, email, subjects taught, tutor certifications and their average tutor rating. This data needs to be connected in a respected network to create the most logical pairs of tutors and students. We want to make sure this database system is focused on the individual student to avoid inconsistency with incorrect tutoring subjects or tutors that have poor reviews.

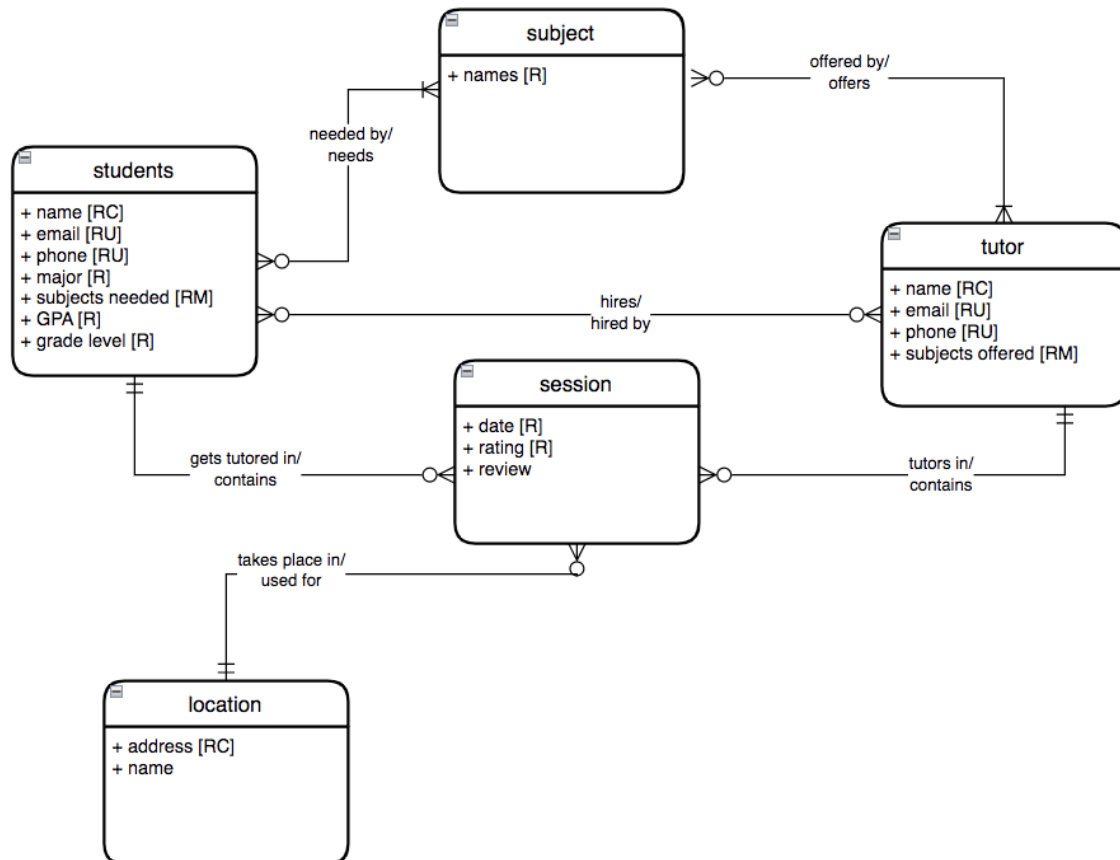
Problem and Solution

The major obstacle that faces Syracuse University Tutors is sorting and analyzing student and tutor data to schedule tutors for students. Syracuse University Tutors needs a revamped system for analyzing these large amounts of data, because each school year, more students need tutoring through Syracuse University Tutors. Student information shows what the students need tutoring in and the tutor information shows what the tutor offers for teaching. Our project aims to make the data concise and readable for Syracuse University Tutors by creating a database system that makes pairing tutors and students accessible. The database system will pair students and tutors and assign them to a said location, in a central location on campus, making it easy to meet.. Also stored in this database system will be tutor reviews so that tutors

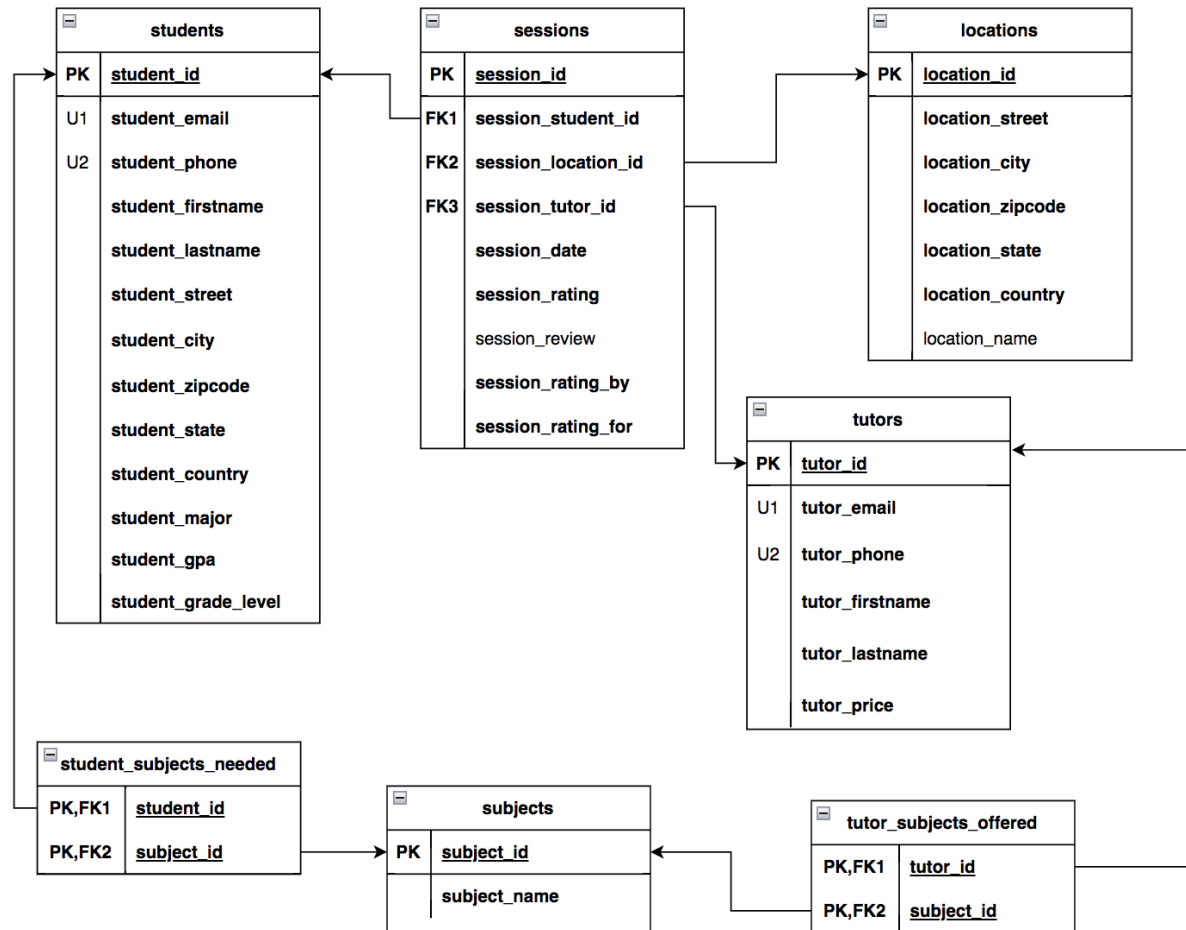
can be filtered based on how well they teach, and students can be sure they are getting the best possible tutor.

Syracuse University Tutors also wants to know things such as which tutor has the most students? Which class requires the most tutoring? Which tutor has the best ratings? What dorm needs the most tutoring? What Major has the highest GPA? By answering these questions through our database system, Syracuse University Tutors will improve their functionality for pairing students and tutors.

Conceptual Model



Logical Model



Data Dictionary

Students

Entity Name: Students	Entity and Attributes	Field Type	Nullable	Foreign key constraints	Description
Primary Key	student_id	Int	Not Null		The unique number of the student
	student_email	Varchar(50)	Not Null		Student email address
	student_phone	Char(10)	Not Null		Student phone number
	student_firstname	Varchar(25)	Not Null		Student first name
	student_lastname	Varchar(25)			Student last name
	student_street	Varchar(100)	Not Null		Students street address
	student_city	Varchar(25)	Not Null		Student city
	student_zipcode	Char(5)	Not Null		Student zipcode
	student_state	Char(2)	Not Null		Student state of residence
	student_country	Varchar(25)	Not Null		Student country of residence
	student_major	Varchar(50)	Not Null		Student major
	student_gpa	Decimal(3,2)	Null		Student total grade point average

	student_grade_level	Varchar(15)	Not Null		Student grade level
--	---------------------	-------------	----------	--	---------------------

Tutors

Entity Name: Tutors	Entity and Attributes	Field Type	Nullabl e	Foreign key constraints	Description
Primary Key	tutor_id	Int	Not Null		The unique number of the tutor
	tutor_phone	Char(10)	Not Null		Tutor phone number
	tutor_email	Varchar(25)	Not Null		Tutor email address
	tutor_firstname	Varchar(25)	Not Null		Tutor first name
	tutor_lastname	Varchar(25)	Not Null		Tutor last name
	tutor_price	Decimal(5,2)	Not Null		Tutor price per session

Subjects

Entity Name: Subjects	Entity and Attributes	Field Type	Nullable	Foreign key constraints	Description
Primary Key	subject_id	Int	Not Null		The unique number of the subject
	subject_name	Varchar(25)	Not Null		Name of the subject

Tutors Subjects Offered

Entity Name: Tutors Subjects Offered	Entity and Attributes	Field Type	Nullabl e	Foreign key constraints	Description
Primary Key, Foreign Key	tutor_id	Int	Not Null	References tutors(tutor_id)	The unique number of the tutor
Primary Key, Foreign Key	subject_id	Int	Not Null	References subjects(subject_id)	The unique number of the student

Students Subjects Needed

Entity Name: Students Subjects Needed	Entity and Attributes	Field Type	Nullable	Foreign key constraints	Description
Primary Key, Foreign Key	student_id	Int	Not Null	References students(student_id)	The unique number of the student
Primary Key, Foreign Key	subject_id	Int	Not Null	References subjects(subject_id)	The unique number of the subject

Locations

Entity Name: Location	Entity and Attributes	Field Type	Nullable	Foreign key constraints	Description
Primary Key	location_id	Int	Not Null		The unique number of the location
	location_name	Varchar(50)	Null		Location building name
	location_street	Varchar(50)	Not Null		Location street address
	location_city	Varchar(25)	Not Null		The city the location is in
	location_zipcode	Char(5)	Not Null		Zip code of the location
	location_state	Char(2)			State the location is in
	location_country	Varchar(25)	Not Null		Country the location is in

Sessions

Entity Name: Sessions	Entity and Attributes	Field Type	Nullable	Foreign key constraints	Description
Primary Key	session_id	Int	Not Null		The unique number of the session
Foreign Key	session_student_id	Int	Not Null	References students(student_id)	The unique number of the student in the session

Foreign Key	session_tutor_id	Int	Not Null	References tutors(tutor_id)	The unique number of the tutor in the session
Foreign Key	session_location_id	Int	Not Null	References locations(location_id)	The unique number of the location of the session
	session_date	Date	Not Null		Date of the session
	session_rating	Decimal(2,1)	Not Null		Rating for the session
	session_review	Varchar(100)	Null		Review for the session
	session_rating_by	Int	Not Null		Rating written by
	session_rating_for	Int	Not Null		Rating written for

Business Rules

- 1) Tutors can offer up to 2 subjects
- 2) Students can ask for help in up to 2 subjects
- 3) Students must be an undergraduate at Syracuse University
- 4) Students can give any rating of any decimal from 0-5
- 5) Students and Tutors can request or offer from a variety of 6 subjects. The subjects are Math, Science, English, History, Foreign Language, and Computer Science.
- 6) Student can have multiple tutors
- 7) Tutors can tutor multiple students
- 8) Sessions can take place in 1 location out of 3 options
- 9) Location can have many sessions
- 10) Students can have many sessions
- 11) Tutoring is exclusively for one tutor and one student
- 12) Tutor can have many sessions
- 13) A required rating is given out after each session by the student, a review is optional

Database System Infrastructure

We used the following tools to create and implement this project:

1) Draw.io:

We created an entity-relationship diagram using MS Visio. We used this to create both the conceptual and logical model.

2) SQL Server:

We used SQL Server as the database that stored all the tables and their data. We created and populate tables in the database using SQL queries.

3) Make.powerapps.com:

We used this website to create the interface for the system. We linked our tables created in SQL Server. Then, we created forms to take user input and display the necessary information.

Creating Tables and Inserting Sample Data

```
--DOWN
DROP VIEW if exists v_session_score
DROP VIEW if exists v_subjects_needed_count
DROP VIEW if exists v_tutor_sessions_given
DROP VIEW if exists v_tutor_average_rating
DROP VIEW if exists v_student_subjects
DROP TABLE if exists sessions
DROP TABLE if exists locations
DROP TABLE if exists students_subjects_needed
DROP TABLE if exists tutors_subjects_offered
DROP TABLE if exists tutor_certifications
DROP TABLE if exists certifications
DROP TABLE if exists subjects
DROP TABLE if exists students
DROP TABLE if exists tutors
```

--UP

```
CREATE TABLE students (  
    student_id int identity NOT NULL,  
    student_email varchar(50) NOT NULL,  
    student_phone char(10) NOT NULL,  
    student_firstname varchar(25) NOT NULL,  
    student_lastname varchar(25) NOT NULL,  
    student_street varchar(100) NOT NULL,  
    student_city varchar(25) NOT NULL,  
    student_zipcode char(5) NOT NULL,  
    student_state char(2),  
    student_country varchar(25) NOT NULL,  
    student_major varchar(50) NOT NULL,  
    student_gpa decimal (3,2) NULL,  
    student_grade_level varchar(15) NOT NULL,  
    CONSTRAINT pk_students_student_id PRIMARY KEY (student_id),  
    CONSTRAINT ck_students_student_gpa CHECK (student_gpa <= 4.0 AND student_gpa >= 0),  
    CONSTRAINT u_students_student_email UNIQUE (student_email),  
    CONSTRAINT u_students_student_phone UNIQUE (student_phone)  
)
```

```
CREATE TABLE tutors(  
    tutor_id int identity NOT NULL,  
    tutor_phone char(10) NOT NULL,  
    tutor_email varchar(50) NOT NULL,  
    tutor_firstname varchar(25) NOT NULL,  
    tutor_lastname varchar(25) NOT NULL,  
    tutor_price DECIMAL (5,2) NOT NULL,  
    CONSTRAINT pk_tutors_tutor_id PRIMARY KEY (tutor_id),  
    CONSTRAINT u_tutors_tutor_email UNIQUE (tutor_email),  
    CONSTRAINT u_tutors_tutor_phone UNIQUE (tutor_phone)  
)
```

```
CREATE TABLE subjects(  
    subject_id int IDENTITY NOT NULL,  
    subject_name varchar(25) NOT NULL  
    CONSTRAINT pk_subjects_subject_id PRIMARY KEY (subject_id),  
    CONSTRAINT u_subjects_subject_name UNIQUE (subject_name)  
)
```

```

CREATE TABLE certifications(
    certification_id int IDENTITY NOT NULL,
    certification_name varchar(100) NULL,
    CONSTRAINT pk_certifications_certification_id PRIMARY KEY (certification_id)
)

CREATE TABLE tutor_certifications(
    tutor_id int NOT NULL,
    certification_id int NOT NULL,
    CONSTRAINT tutor_certifications_tutor_id_certification_id PRIMARY KEY (tutor_id,
certification_id),
    CONSTRAINT fk_tutor_certifications foreign key (tutor_id)
references tutors(tutor_id),
    CONSTRAINT fk_tutor_certifications2 foreign key (certification_id)
references certifications(certification_id)
)

CREATE TABLE tutors_subjects_offered(
    tutor_id int NOT NULL,
    subject_id int NOT NULL
    CONSTRAINT tutors_subjects_offered_tutor_id_subject_id PRIMARY KEY (tutor_id,
subject_id),
    CONSTRAINT fk_tutor_subjects_offered foreign key (tutor_id)
references tutors(tutor_id),
    CONSTRAINT fk_tutor_subjects_offered2 foreign key (subject_id)
references subjects(subject_id)
)

CREATE TABLE students_subjects_needed(
    student_id int NOT NULL,
    subject_id int NOT NULL
    CONSTRAINT students_subjects_needed_student_id_subject_id PRIMARY KEY (student_id,
subject_id),
    CONSTRAINT fk_students_subjects_needed foreign key (student_id)
references students(student_id),
    CONSTRAINT fk_students_subjects_needed2 foreign key (subject_id)
references subjects(subject_id)
)

CREATE TABLE locations(
    location_id int IDENTITY NOT NULL,

```

```

location_name varchar(50) NULL,
location_street varchar(50) NOT NULL,
location_city varchar(25) NOT NULL,
location_zipcode char(5) NOT NULL,
location_state char(2),
location_country varchar(25) NOT NULL,
CONSTRAINT pk_locations_location_id PRIMARY KEY (location_id)

)

CREATE TABLE sessions(
    session_id int identity NOT NULL,
    session_student_id int NOT NULL,
    session_tutor_id int NOT NULL,
    session_location_id int NOT NULL,
    session_date date NOT NULL,
    session_rating decimal (2,1) NOT NULL,
    session_review varchar(100) NULL,
    session_rating_by int NOT NULL,
    session_rating_for int NOT NULL,
    CONSTRAINT pk_sessions_session_id PRIMARY KEY (session_id),
    CONSTRAINT ck_sessions_session_rating CHECK (session_rating <= 5.0 AND
session_rating >= 0),
    CONSTRAINT fk_sessions_student_id foreign KEY (session_student_id)
references students(student_id),
    CONSTRAINT fk_sessions_tutor_id foreign KEY (session_tutor_id)
references tutors(tutor_id),
    CONSTRAINT fk_sessions_location_id foreign KEY (session_location_id)
references locations(location_id),

)

----- INSERTS

INSERT INTO subjects(subject_name)
VALUES
('Math'),
('Science'),
('English'),
('Foreign Language'),
('History'),

```

```

('Computer Science');

INSERT INTO students(student_email, student_phone, student_firstname,
student_lastname, student_street, student_city, student_zipcode, student_state,
student_country, student_major, student_gpa, student_grade_level)
VALUES
('johnsmith@gmail.com', 5165698876, 'John', 'Smith', '11 Cypress Ave', 'Cedarhurst',
11516, 'NY', 'USA', 'Statistics', '3.4', 'Sophomore'),
('maryjohnson@gmail.com', 7734456666, 'Mary', 'Johnson', '103 Maple St', 'Chicago',
60007, 'IL', 'USA', 'Art History', '3.6', 'Senior'),
('juliad123@gmail.com', 8568877654, 'Julia', 'Denni', '9 Lexington St', 'Medford',
08055, 'NJ', 'USA', 'Biochemistry', '3.5', 'Junior'),
('antedwards@gmail.com', 2318867054, 'Anthony', 'Edwards', '1103 Rose Ct', 'Minnesota',
55106, 'MN', 'USA', 'Physics', '2.9', 'Junior'),
('rgould@gmail.com', 5161109767, 'Rose', 'Gould', '11 Great Oaks Rd', 'Roslyn', 11577,
'NY', 'USA', 'Communications', '3.8', 'Senior'),
('jnzippin@yahoo.com', 5543320091, 'Jared', 'Zippin', '45 Westport Rd', 'Long Meadow',
01001, 'MA', 'USA', 'Nutrition', '3.1', 'Sophomore'),
('bzoot@gmail.com', 4567891234, 'Benjamin', 'Zootson', '101 Maple Dr', 'Boston',
02101, 'MA', 'USA', 'Forensic Science', '3.1', 'Sophomore'),
('quinnlaurent@yahoo.com', 4587384458, 'Quinn', 'Laurent', '4 Doolittle Rd', 'New
Rochelle', 10538, 'NY', 'USA', 'Spanish', '3.4', 'Senior'),
('sydjones12@gmail.com', 0987651234, 'Sydney', 'Jones', '2341 Oswald St', 'Chicago',
60007, 'IL', 'USA', 'Accounting', '3.9', 'Junior'),
('aacclark@syr.edu', 3156831230, 'Anna', 'Clark', 'South Crouse Ave', 'Syracuse', 13210,
'NY', 'USA', 'Math', '4.0', 'Senior'),
('bewatson@syr.edu', 3159419573, 'Ben', 'Watson', 'Waverly Ave', 'Syracuse', 13210,
'NY', 'USA', 'Computer Science', '3.4', 'Junior'),
('canydem@syr.edu', 3039992437, 'Cara', 'Nydem', 'Ostrom Ave', 'Syracuse', 13210,
'NY', 'USA', 'Computer Science', '3.0', 'Junior'),
('molaw@syr.edu', 3158733754, 'Molly', 'Law', 'Ackerman Ave', 'Syracuse', 13210,
'NY', 'USA', 'English', '3.1', 'Freshman'),
('gobarns@syr.edu', 2182013156, 'Greg', 'Barns', 'Waverly Ave', 'Syracuse', 13201,
'NY', 'USA', 'Foreign Language', '2.5', 'Freshman'),
('llbaron@syr.edu', 3032337755, 'Loren', 'Baron', 'Ostrom Ave', 'Syracuse', 13204,
'NY', 'USA', 'English', '2.8', 'Sophomore'),
('kmloy@syr.edu', 2156734599, 'Kaitlyn', 'Loy', 'Ackerman Ave', 'Syracuse', 13210,
'NY', 'USA', 'History', '3.9', 'Sophomore'),
('mlmichael@syr.edu', 3154357786, 'Maddie', 'Michael', 'Irving Ave', 'Syracuse', 13210,
'NY', 'USA', 'Math', '3.0', 'Junior'),

```

```

('crsanchez@syr.edu', 315443812, 'Chris','Sanchez', 'Waverly Ave','Syracuse', 13210,
'NY','USA', 'Accounting', '2.9', 'Freshman'),
('ajkortz@syr.edu', 9934432565, 'Anthony','Kortz', '1st E Ave','Syracuse', 13210,
'NY','USA', 'Pre-Medicine', '3.2', 'Junior'),
('sagordon@syr.edu', 3756737749, 'Sarah','Gordon', 'Dorothy Dr','Syracuse', 13203,
'NY','USA', 'Accounting', '2.9', 'Senior'),
('njplatt@syr.edu', 2747848923, 'Noah','Platt', 'Irving Ave','Syracuse', 13203,
'NY','USA', 'Finance', '3.6', 'Sophomore'),
('ehapple@syr.edu', 2154473872, 'Evan','Apple', 'Crouse Ave','Syracuse', 13210,
'NY','USA', 'Pre-Medicine', '3.2', 'Freshman'),
('vbroch@syr.edu', 7437582281, 'Victoria','Roch', 'Waverly Ave','Syracuse', 13210,
'NY','USA', 'Foreign Language', '3.6', 'Sophomore'),
('alplaza@syr.edu', 7497548982, 'Aubrey','Plaza', 'Fairfield Ave','Syracuse', 13203,
'NY','USA', 'Foreign Language', '3.9', 'Senior');

```

```

INSERT INTO locations(location_street, location_city, location_zipcode,
location_state, location_country, location_name)

```

```
VALUES
```

```

('222 Waverly Ave','Syracuse', 13244, 'NY', 'USA', 'Bird Library'),
('130 Sims Dr','Syracuse', 13244, 'NY', 'USA', 'Carnegie Library'),
('200 Waverly Ave','Syracuse', 13244, 'NY', 'USA', 'Schine Student Center');

```

```
INSERT INTO students_subjects_needed(student_id, subject_id)
```

```
VALUES
```

```

(1,1),
(2,5),
(3,2),
(4,2),
(5,1),
(6,4),
(6,2),
(7,2),
(8,4),
(9,1),
(9,5),
(10,1),
(11,6),
(12,6),
(13,3),
(14,4),
(15,3),

```



```

(16,5),
(17,1),
(18,1),
(19,2),
(20,1),
(21,1),
(22,2),
(23,4),
(24,4);

INSERT INTO tutors(tutor_firstname, tutor_lastname, tutor_email, tutor_phone,
tutor_price)
VALUES
('Tim', 'Berman', 'timberman@gmail.com', 9178830989, 50.00),
('John', 'Quinn', 'jquinn3@gmail.com', 8048831900, 100.00),
('Mary', 'Jane', 'mfjane@yahoo.com', 4456860977, 75.00),
('Catherine', 'Reynolds', 'catreynolds@yahoo.com', 9172032067, 40.00),
('Heather', 'Sampson', 'heathers56@gmail.com', 5166635505, 60.00),
('Mac', 'Ruby', 'mc99@gmail.com', 5136433505, 60.00);

INSERT INTO tutors_subjects_offered(tutor_id, subject_id)
VALUES
(1,1),
(2,2),
(3,3),
(3,1),
(4,4),
(5,5),
(6,6);

INSERT INTO sessions(session_student_id, session_tutor_id, session_location_id,
session_date, session_rating, session_review, session_rating_by, session_rating_for)
VALUES
(1,1,1,'2021-09-21',5.0,'Very helpful',1,1),
(2,5,3,'2021-09-23',4.0, NULL ,2,5),
(3,2,2,'2021-09-28',2.5,'Not very patient',3,2),
(4,2,2,'2021-10-15',4.0,'Got an 80 on my test after session',4,2),
(5,1,1,'2021-10-15',4.0,'Good',5,1),
(6,4,1,'2021-09-21',4.5, NULL ,6,4),
(7,2,2,'2021-09-12',5.0,'Very helpful',7,2),
(8,4,3,'2021-09-04',4.0, NULL ,8,4),

```

```
(9,5,2,'2021-09-09',2.5,'Not good.',9,5),
(10,1,1,'2021-10-13',4.0,'Good.',10,1),
(11,6,1,'2021-09-20',4.5, NULL, 11,6),
(12,6,2,'2021-09-29',5.0,'Helpful',12,6),
(13,3,3,'2021-09-14',4.0, NULL ,13,3),
(14,4,3,'2021-09-16',5.0,'Very patient',14,4),
(15,3,1,'2021-10-15',2.0,'Was very late',15,3),
(16,5,2,'2021-11-01',4.5, 'Good', 16,5),
(17,1,1,'2021-11-01',3.0, 'Not terrible, but goes too fast', 17,1),
(18,1,1, '2021-11-11',4.5, 'Helped a lot ', 18,1),
(19,2,2,'2021-11-10',4.5, 'Learned a lot', 19,2),
(20,1,1,'2021-11-05',4.0, 'Good', 20,1),
(21,1,1,'2021-11-04',3.5, 'He was nice ', 21,1),
(22,2,2,'2021-11-03',4.0, 'I liked it', 22,2),
(23,4,3,'2021-11-23',5.0, 'Great tutor', 23,4),
(24,4,2, '2021-11-22',5.0, 'Nice', 24,4);
GO
```

Data Questions

1. Which tutor has the highest average rating?
2. Which tutor has tutored the most sessions?
3. What subject needed the most help?
4. What session received the lowest rating and why?
5. What location was used the most often?

```
-- Which tutor has the highest average rating?  
CREATE VIEW v_tutor_average_rating AS  
select tutor_id, tutor_firstname, tutor_lastname, avg(session_rating) as  
average_rating  
from tutors  
join sessions on tutor_id = session_tutor_id  
group by tutor_id, tutor_firstname, tutor_lastname  
order by average_rating  
GO
```

From this table, we can tell that Mac Ruby has the highest average tutor rating.

	tutor_id	tutor_firstname	tutor_lastname	average_rating
1	3	Mary	Jane	3.000000
2	5	Heather	Sampson	3.666666
3	1	Tim	Berman	4.000000
4	2	John	Quinn	4.000000
5	4	Catherine	Reynolds	4.700000
6	6	Mac	Ruby	4.750000

```
-- Which tutor has tutored the most sessions?
CREATE VIEW v_tutor_sessions_given AS
select tutor_firstname, tutor_lastname session_tutor_id, count(session_tutor_id) as
sessions_given from sessions
JOIN tutors on session_tutor_id = tutor_id
group by session_tutor_id, tutor_firstname, tutor_lastname
GO
```

From this table, we can see that the tutor named Tim Berman has given the most sessions out of any tutor.

	tutor_firstname ▾	session_tutor_id ▾	sessions_given ▾
1	Mary	Jane	2
2	Mac	Ruby	2
3	Heather	Sampson	3
4	Catherine	Reynolds	5
5	John	Quinn	5
6	Tim	Berman	7

```
-- What subject needed the most help?
CREATE VIEW v_subjects_needed_count AS
select subject_name, count(subject_id) as count from v_student_subjects
GROUP BY subject_name
GO
```

From this table, we can see that Math needs the most help because the count is the highest out of the subjects.

	subject_name ▾	count ▾
1	Computer Science	2
2	English	2
3	Foreign Language	5
4	History	3
5	Math	8
6	Science	6

```
-- What session received the lowest rating and why?
CREATE VIEW v_session_score as
select TOP 5 student_firstname + ' ' + student_lastname as student_name,
tutor_firstname + ' ' + tutor_lastname as tutor_name, session_date,
session_rating, session_review
from sessions
join tutors on tutor_id = session_tutor_id
join students on student_id = session_student_id
order by session_rating asc
GO
```

From this table, we can see that the tutor named Loren Baron received the lowest rating of 2.0 for the reason that they were very late.

	student_name	tutor_name	session_date	session_rating	session_review
1	Loren Baron	Mary Jane	2021-10-15	2.0	Was very late
2	Julia Denni	John Quinn	2021-09-28	2.5	Not very patient
3	Sydney Jones	Heather Sampson	2021-09-09	2.5	Not good.
4	Maddie Michael	Tim Berman	2021-11-01	3.0	Not terrible, but goes to...
5	Noah Platt	Tim Berman	2021-11-04	3.5	He was nice

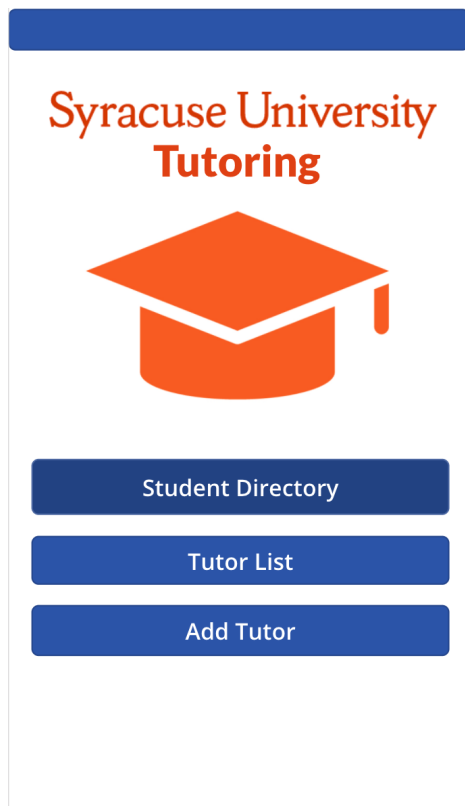
```
-- What location was used the most often?
select location_name, count(session_location_id) as count from locations
join sessions on session_location_id = location_id
group by location_name
```

This table shows us that Bird Library is the most frequently used tutoring location for sessions.





	location_name	count
1	Bird Library	10
2	Carnegie Library	9
3	Schine Student Center	5

Forms

Our Main Menu screen allows students to browse the tutor list and find the best tutor for them, while tutors can browse the student list and find students who need their assistance with a certain subject.



Below is an example of what the student list looks like.

Students				
 Search items				
Anna Clark	Math	>		
Anthony Edwards	Science	>		
Anthony Kortz	Science	>		
Aubrey Plaza	Foreign Language	>		
Ben Watson	Computer Science	>		
Benjamin Zootson	Science	>		
Cara Nydem	Computer Science	>		
Chris Sanchez	Math	>		
Evan Apple	Science	>		
Greg Barnes		>		

Student lists can be expanded to show more student information including email and phone numbers.

✕ Student Information ✓

student_name

Anna Clark

student_email

aaclark@syr.edu

subject_name


Math


student_phone

3156831230




Next, we have a screen that displays the list of tutors and their emails.


 Tutor List

 Search items


Tim
Berman
timberman@gmail.com




John
Quinn
jquinn3@gmail.com




Mary
Jane
mfjane@yahoo.com




Catherine
Reynolds
catreynolds@yahoo.com



Heather
Sampson
heathers56@gmail.com



Mac
Ruby
mc99@gmail.com



Plus, we have a screen that shows more tutor information.

✕ Tutor Info

tutor_firstname

Tim

tutor_lastname

Berman

tutor_email

timberman@gmail.com



tutor_phone

9178830989

tutor_price

50

We also have a screen where tutors can add their own information into the app.

 Add Tutor 

* Enter First Name:

* Enter Last Name:

* Enter Email:

* Enter Phone Number (10 Digits, No Dashes)

* Enter Price Per Hour

Log

Jack and Avery worked equally on all parts of the project.