

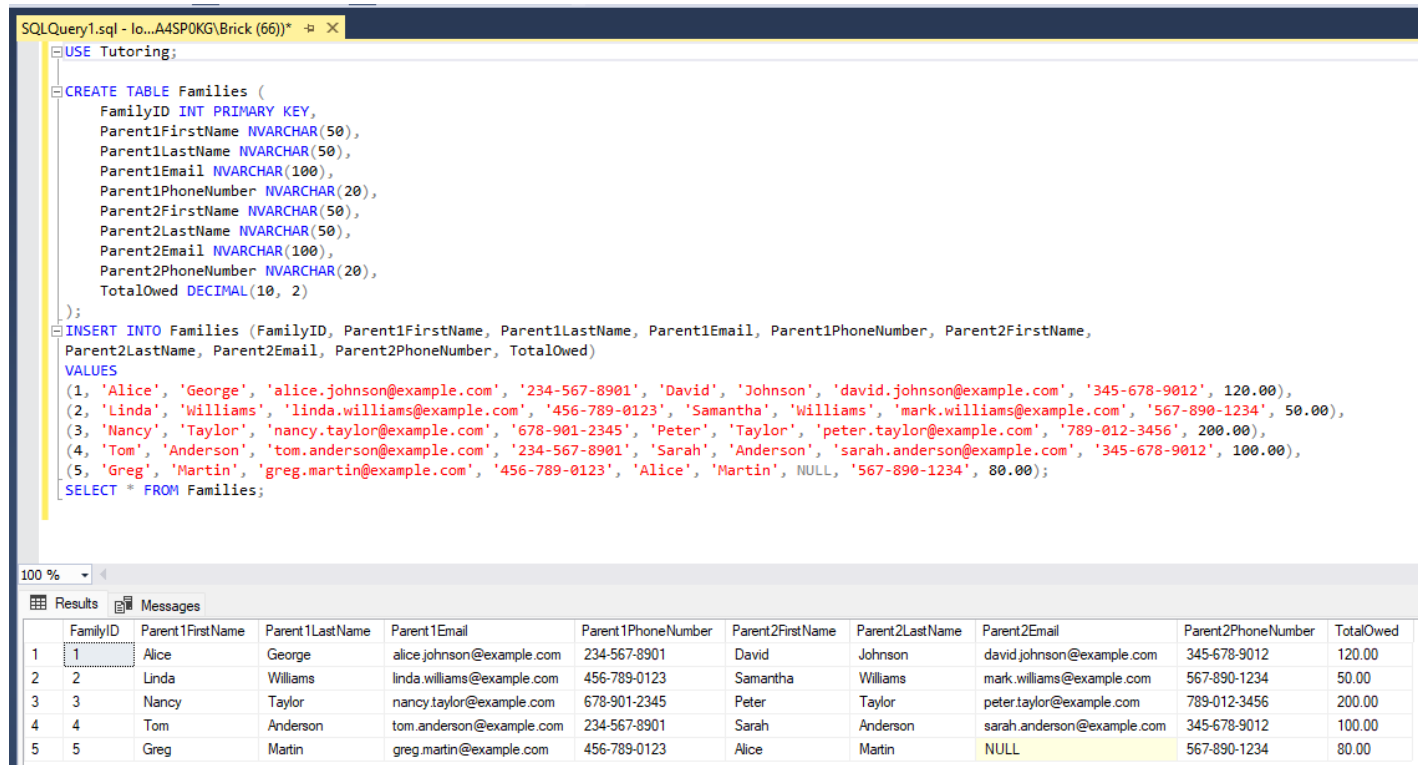
# Databases CS4332 Project 9

Evan Varan, Ize4

## Table Creation and Population:

Display of the user-interface and table creation of the E/R diagram is shown below. Microsoft SQL Server Management Studio was used to create the following:

Family Table:



The screenshot displays the SQL Query window in Microsoft SQL Server Management Studio. The query is as follows:

```
USE Tutoring;

CREATE TABLE Families (
    FamilyID INT PRIMARY KEY,
    Parent1FirstName NVARCHAR(50),
    Parent1LastName NVARCHAR(50),
    Parent1Email NVARCHAR(100),
    Parent1PhoneNumber NVARCHAR(20),
    Parent2FirstName NVARCHAR(50),
    Parent2LastName NVARCHAR(50),
    Parent2Email NVARCHAR(100),
    Parent2PhoneNumber NVARCHAR(20),
    TotalOwed DECIMAL(10, 2)
);

INSERT INTO Families (FamilyID, Parent1FirstName, Parent1LastName, Parent1Email, Parent1PhoneNumber, Parent2FirstName, Parent2LastName, Parent2Email, Parent2PhoneNumber, TotalOwed)
VALUES
(1, 'Alice', 'George', 'alice.johnson@example.com', '234-567-8901', 'David', 'Johnson', 'david.johnson@example.com', '345-678-9012', 120.00),
(2, 'Linda', 'Williams', 'linda.williams@example.com', '456-789-0123', 'Samantha', 'Williams', 'mark.williams@example.com', '567-890-1234', 50.00),
(3, 'Nancy', 'Taylor', 'nancy.taylor@example.com', '678-901-2345', 'Peter', 'Taylor', 'peter.taylor@example.com', '789-012-3456', 200.00),
(4, 'Tom', 'Anderson', 'tom.anderson@example.com', '234-567-8901', 'Sarah', 'Anderson', 'sarah.anderson@example.com', '345-678-9012', 100.00),
(5, 'Greg', 'Martin', 'greg.martin@example.com', '456-789-0123', 'Alice', 'Martin', NULL, '567-890-1234', 80.00);

SELECT * FROM Families;
```

The Results window shows the data inserted into the Families table:

FamilyID	Parent1FirstName	Parent1LastName	Parent1Email	Parent1PhoneNumber	Parent2FirstName	Parent2LastName	Parent2Email	Parent2PhoneNumber	TotalOwed
1	Alice	George	alice.johnson@example.com	234-567-8901	David	Johnson	david.johnson@example.com	345-678-9012	120.00
2	Linda	Williams	linda.williams@example.com	456-789-0123	Samantha	Williams	mark.williams@example.com	567-890-1234	50.00
3	Nancy	Taylor	nancy.taylor@example.com	678-901-2345	Peter	Taylor	peter.taylor@example.com	789-012-3456	200.00
4	Tom	Anderson	tom.anderson@example.com	234-567-8901	Sarah	Anderson	sarah.anderson@example.com	345-678-9012	100.00
5	Greg	Martin	greg.martin@example.com	456-789-0123	Alice	Martin	NULL	567-890-1234	80.00

Schedule Table:

SQLQuery1.sql - lo...A4SP0KG\Brick (66))\*

```
USE Tutoring;
CREATE TABLE Schedule (
    StudentID INT,
    TutorID INT,
    SubjectName NVARCHAR(50),
    SessionOccurance NVARCHAR(50),
    SessionTime TIME,
    PRIMARY KEY (StudentID, TutorID, SubjectName),
    FOREIGN KEY (StudentID) REFERENCES Students(StudentID),
    FOREIGN KEY (TutorID) REFERENCES Tutors(TutorID),
    FOREIGN KEY (SubjectName) REFERENCES Subjects(SubjectName)
);

INSERT INTO Schedule(StudentID, TutorID, SubjectName, SessionOccurance, SessionTime)
VALUES
(1, 1, 'Algebra 1', 'Every Monday', '09:00:00'),
(2, 1, 'Geometry', 'Every other Tuesday', '10:30:00'),
(3, 2, 'Algebra 2', 'Every Wednesday', '13:00:00'),
(4, 3, 'Pre-Calculus', 'Every Friday', '15:00:00'),
(5, 4, 'Calculus', 'Every Thursday', '16:30:00');

select * from Schedule
```

100 %

Results Messages

	StudentID	TutorID	SubjectName	SessionOccurance	SessionTime
1	1	1	Algebra 1	Every Monday	09:00:00.0000000
2	2	1	Geometry	Every other Tuesday	10:30:00.0000000
3	3	2	Algebra 2	Every Wednesday	13:00:00.0000000
4	4	3	Pre-Calculus	Every Friday	15:00:00.0000000
5	5	4	Calculus	Every Thursday	16:30:00.0000000

## Student Table:

SQLQuery1.sql - lo...A4SP0KG\Brick (66))\*

USE Tutoring;

CREATE TABLE Students (

StudentID INT PRIMARY KEY,

FamilyID INT,

StudentFirstName NVARCHAR(50),

StudentLastName NVARCHAR(50),

StudentEmail NVARCHAR(100),

StudentPhoneNumber NVARCHAR(20),

FOREIGN KEY (FamilyID) REFERENCES Families(FamilyID)

);

INSERT INTO Students (StudentID, FamilyID, StudentFirstName, StudentLastName, StudentEmail, StudentPhoneNumber)

VALUES

(1, 1, 'Emma', 'Johnson', 'emma.johnson@example.com', '111-222-3333'),

(2, 1, 'Lucas', 'Johnson', 'lucas.johnson@example.com', '222-333-4444'),

(3, 2, 'Sophia', 'Williams', 'sophia.williams@example.com', '333-444-5555'),

(4, 2, 'Mason', 'Williams', 'mason.williams@example.com', '444-555-6666'),

(5, 3, 'Olivia', 'Taylor', 'olivia.taylor@example.com', '555-666-7777'),

(6, 4, 'Liam', 'Anderson', NULL, '123-456-7890'),

(7, 4, 'Noah', 'Anderson', NULL, '234-567-8901'),

(8, 5, 'Ava', 'Martin', 'ava.martin@example.com', NULL),

(9, 5, 'Ethan', 'Martin', NULL, NULL),

(10, 3, 'Mia', 'Taylor', 'mia.taylor@example.com', '555-666-7777'),

(11, 2, 'Ella', 'Williams', 'ella.williams@example.com', '666-777-8888'),

(12, 1, 'Benjamin', 'Johnson', NULL, '777-888-9999'),

(13, 1, 'Charlotte', 'Johnson', 'charlotte.johnson@example.com', NULL);

select \* from students

100 %

Results Messages

	StudentID	FamilyID	StudentFirstName	StudentLastName	StudentEmail	StudentPhoneNumber
1	1	1	Emma	Johnson	emma.johnson@example.com	111-222-3333
2	2	1	Lucas	Johnson	lucas.johnson@example.com	222-333-4444
3	3	2	Sophia	Williams	sophia.williams@example.com	333-444-5555
4	4	2	Mason	Williams	mason.williams@example.com	444-555-6666
5	5	3	Olivia	Taylor	olivia.taylor@example.com	555-666-7777
6	6	4	Liam	Anderson	NULL	123-456-7890
7	7	4	Noah	Anderson	NULL	234-567-8901
8	8	5	Ava	Martin	ava.martin@example.com	NULL
9	9	5	Ethan	Martin	NULL	NULL
10	10	3	Mia	Taylor	mia.taylor@example.com	555-666-7777
11	11	2	Ella	Williams	ella.williams@example.com	666-777-8888
12	12	1	Benjamin	Johnson	NULL	777-888-9999
13	13	1	Charlotte	Johnson	charlotte.johnson@example.com	NULL

Subject Table:

SQLQuery1.sql - lo...A4SP0KG\Brick (66))\*

```
USE Tutoring;  
CREATE TABLE Subjects (  
    SubjectName NVARCHAR(50) PRIMARY KEY,  
    Charge DECIMAL(10, 2)  
);  
  
INSERT INTO Subjects (SubjectName, Charge)  
VALUES  
    ('Algebra 1', 45.00),  
    ('Algebra 2', 50.00),  
    ('Calculus', 65.00),  
    ('Geometry', 55.00),  
    ('Physics', 70.00),  
    ('Pre-Calculus', 60.00);  
  
select * from Subjects
```

100 %

Results Messages

	SubjectName	Charge
1	Algebra 1	45.00
2	Algebra 2	50.00
3	Calculus	65.00
4	Geometry	55.00
5	Physics	70.00
6	Pre-Calculus	60.00

Tutor Table:

SQLQuery1.sql - lo...A4SP0KG\Brick (66))\*

```
USE Tutoring;
CREATE TABLE Tutors (
    TutorID INT PRIMARY KEY,
    TutorFirstName NVARCHAR(50),
    TutorLastName NVARCHAR(50)
);

INSERT INTO Tutors (TutorID, TutorFirstName, TutorLastName)
VALUES
(1, 'John', 'Doe'),
(2, 'Jane', 'Smith'),
(3, 'Michael', 'Johnson'),
(4, 'Emily', 'Davis'),
(5, 'Chris', 'Brown');
select * from Tutors
```

100 %

Results Messages

	TutorID	TutorFirstName	TutorLastName
1	1	John	Doe
2	2	Jane	Smith
3	3	Michael	Johnson
4	4	Emily	Davis
5	5	Chris	Brown

## Writing the 10 Queries:

The 10 queries defined in project 7 are shown below in t-SQL with outputs. This is also done in Microsoft SQL Server Management Studio. Note that there are no errors in the code below, the IntelliSense just fails to update before queries are run.

- 1) Display all parents and students of a single family.

SQLQuery1.sql - lo...A4SP0KG\Brick (66))\* -> X

```
USE Tutoring;  
SELECT f.FamilyID, s.StudentID, f.Parent1FirstName, f.Parent1LastName, f.Parent2FirstName,  
f.Parent2LastName, s.StudentFirstName, s.StudentLastName  
FROM Families f  
JOIN students s ON f.FamilyID = s.FamilyID  
WHERE f.FamilyID = 1;
```

100 %

Results Messages

	FamilyID	StudentID	Parent1FirstName	Parent1LastName	Parent2FirstName	Parent2LastName	StudentFirstName	StudentLastName
1	1	1	Alice	George	David	Johnson	Emma	Johnson
2	1	2	Alice	George	David	Johnson	Lucas	Johnson
3	1	12	Alice	George	David	Johnson	Benjamin	Johnson
4	1	13	Alice	George	David	Johnson	Charlotte	Johnson

- 2) Display all students who are taking Algebra 2.

SQLQuery1.sql - lo...A4SP0KG\Brick (66))\* -> X

```
USE Tutoring;  
SELECT s.StudentID, s.StudentFirstName, s.StudentLastName, sc.SubjectName  
FROM Students s  
JOIN Schedule sc ON sc.StudentID = s.StudentID  
WHERE sc.SubjectName = 'Algebra 2';
```

100 %

Results Messages

	StudentID	StudentFirstName	StudentLastName	SubjectName
1	3	Sophia	Williams	Algebra 2

- 3) Display all contact information for a single family.

SQLQuery1.sql - lo...A4SP0KG\Brick (66))\*

```
USE Tutoring;  
SELECT f.Parent1Email, f.Parent1PhoneNumber, f.Parent2Email,  
f.Parent2PhoneNumber, s.StudentEmail, s.StudentPhoneNumber  
FROM Families f  
JOIN Students s ON s.FamilyID = f.FamilyID;
```

100 %

Results Messages

	Parent1Email	Parent1PhoneNumber	Parent2Email	Parent2PhoneNumber	StudentEmail	StudentPhoneNumber
1	alice.johnson@example.com	234-567-8901	david.johnson@example.com	345-678-9012	emma.johnson@example.com	111-222-3333
2	alice.johnson@example.com	234-567-8901	david.johnson@example.com	345-678-9012	lucas.johnson@example.com	222-333-4444
3	linda.williams@example.com	456-789-0123	mark.williams@example.com	567-890-1234	sophia.williams@example.com	333-444-5555
4	linda.williams@example.com	456-789-0123	mark.williams@example.com	567-890-1234	mason.williams@example.com	444-555-6666
5	nancy.taylor@example.com	678-901-2345	peter.taylor@example.com	789-012-3456	olivia.taylor@example.com	555-666-7777
6	tom.anderson@example.com	234-567-8901	sarah.anderson@example.com	345-678-9012	NULL	123-456-7890
7	tom.anderson@example.com	234-567-8901	sarah.anderson@example.com	345-678-9012	NULL	234-567-8901
8	greg.martin@example.com	456-789-0123	NULL	567-890-1234	ava.martin@example.com	NULL
9	greg.martin@example.com	456-789-0123	NULL	567-890-1234	NULL	NULL
10	nancy.taylor@example.com	678-901-2345	peter.taylor@example.com	789-012-3456	mia.taylor@example.com	555-666-7777
11	linda.williams@example.com	456-789-0123	mark.williams@example.com	567-890-1234	ella.williams@example.com	666-777-8888
12	alice.johnson@example.com	234-567-8901	david.johnson@example.com	345-678-9012	NULL	777-888-9999
13	alice.johnson@example.com	234-567-8901	david.johnson@example.com	345-678-9012	charlotte.johnson@example.com	NULL

- 4) Count how many tutoring sessions are currently in the schedule.

SQLQuery1.sql - lo...A4SP0KG\Brick (66))\*

```
USE Tutoring;  
SELECT COUNT(StudentID) AS TotalNumberOfSession  
FROM Schedule;
```

100 %

Results Messages

	TotalNumberOfSession
1	5

- 5) Display all subjects that cost more than \$50.

SQLQuery1.sql - lo...A4SP0KG\Brick (66))\*

```
USE Tutoring;  
SELECT SubjectName, Charge  
FROM Subjects  
WHERE Charge > 50.00;
```

100 %

Results Messages

	SubjectName	Charge
1	Calculus	65.00
2	Geometry	55.00
3	Physics	70.00
4	Pre-Calculus	60.00

- 6) Display all families that currently owe more than \$100.

SQLQuery1.sql - lo...A4SP0KG\Brick (66))\*

```
USE Tutoring;  
SELECT FamilyID, TotalOwed  
FROM Families  
WHERE TotalOwed > 100.00;
```

100 %

Results Messages

	FamilyID	TotalOwed
1	1	120.00
2	3	200.00



- 7) Find the number of students that have sessions scheduled on Monday.

SQLQuery1.sql - lo...A4SP0KG\Brick (66))\*

```
USE Tutoring;
SELECT COUNT(StudentID) AS NumberOfStudentWithSessionOnMonday
FROM Schedule
WHERE SessionOccurance LIKE '%Monday%';
```

100 %

Results Messages

	NumberOfStudentWithSessionOnMonday
1	1

- 8) Find the number of tutors currently employed by the company.

SQLQuery1.sql - lo...A4SP0KG\Brick (66))\*

```
USE Tutoring;
SELECT COUNT(TutorID) AS NumberOfTutorsEmployed
FROM Tutors;
```

100 %

Results Messages

	NumberOfTutorsEmployed
1	5

- 9) List the Names of the tutors who are currently scheduled for more than one subject.

SQLQuery1.sql - lo...A4SP0KG\Brick (66))\*

```
USE Tutoring;
SELECT t.TutorFirstName, t.TutorLastName, COUNT(DISTINCT s.SubjectName) AS NumberOfSubjects
FROM Tutors t
JOIN Schedule s ON s.TutorID = t.TutorID
GROUP BY t.TutorFirstName, t.TutorLastName, t.TutorID
HAVING COUNT(DISTINCT s.SubjectName) > 1;
```

100 %

Results Messages

	TutorFirstName	TutorLastName	NumberOfSubjects
1	John	Doe	2

10) List the names of the tutors who are currently scheduled to teach Pre-cal.

SQLQuery1.sql - lo...A4SP0KG\Brick (66))\*

```
USE Tutoring;  
SELECT t.TutorFirstName, t.TutorLastName, s.SubjectName  
FROM Tutors t  
JOIN Schedule s ON s.TutorID = t.TutorID  
WHERE s.SubjectName LIKE '%Pre-Calc%';
```

100 %

Results Messages

	TutorFirstName	TutorLastName	SubjectName
1	Michael	Johnson	Pre-Calculus