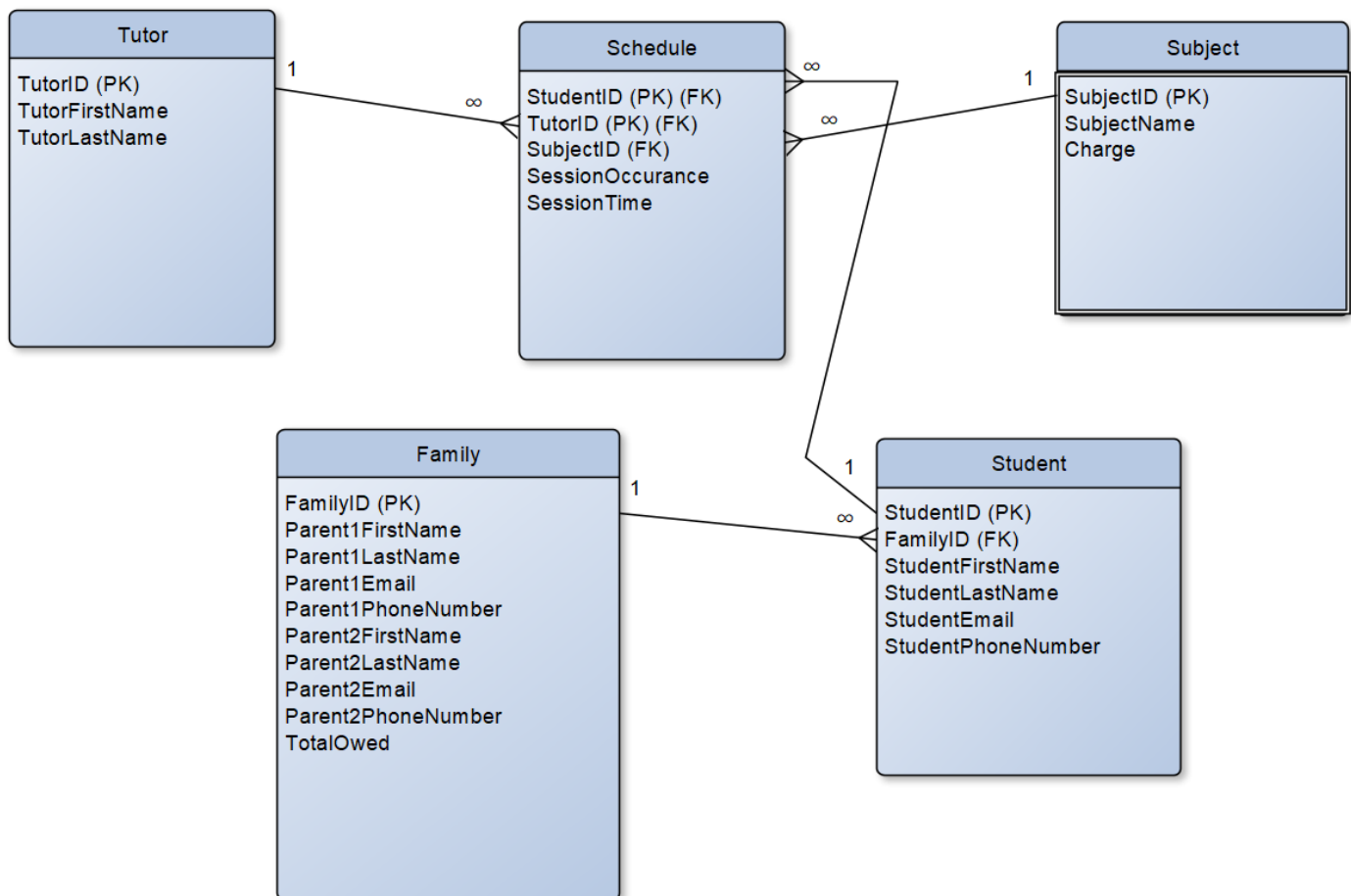


Databases CS4332 Project 8

Evan Varan, lze4

Part I:

The E/R Diagram has been revised. I added a SubjectID as the primary key for the Subject Table. This is referenced by the SubjectID foreign key in the Schedule table. SubjectName is now only in the subject table and is not referenced by anything. This was done to avoid update and insert anomalies as having a name as a primary key could cause a problem if a user inputs "Algebra II" and another user inputs "Algebra 2". The updated E/R diagram is shown below.



Part II:

i)

Display of the user-interface and table creation of the E/R diagram is shown below. Xampp was used to create the following:

Family 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:

StudentID	TutorID	SubjectName	SessionOccurance	SessionTime
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

Student Table:

StudentID	FamilyID	StudentFirstName	StudentLastName	StudentEmail	StudentPhoneNumber
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	liam.anderson@example.com	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

Subject Table:

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

Tutor Table:

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

ii) The 10 queries defined are shown below in SQL with outputs.

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

```
1 select f.FamilyID, s.StudentID, f.Parent1FirstName,f.Parent1LastName,f.Parent2FirstName,f.Parent2LastName,s.StudentFirstName, s.StudentLastName
2 from family f
3 join student s on f.FamilyID = s.FamilyID
4 where f.familyid = 1;
```

☒ Enable foreign key checks

Go Cancel

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

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

2) Display all students who are taking Algebra 2.

```
1 select s.StudentID,s.StudentFirstName, s.StudentLastName, sc.SubjectName
2 from student s
3 join Schedule sc on sc.StudentID = s.StudentID
4 where sc.SubjectName = "Algebra 2";
```

☒ Enable foreign key checks

Go Cancel

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

☐ Show all | Number of rows: 25 | Filter rows: Search this table

Extra options

StudentID	StudentFirstName	StudentLastName	SubjectName
3	Sophia	Williams	Algebra 2

3) Display all contact information for a single family.

```
select f.Parent1Email, f.Parent1PhoneNumber, f.Parent2Email, f.Parent2PhoneNumber, s.StudentEmail, s.StudentPhoneNumber from family f join student s on s.FamilyID = f.FamilyID;
```

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

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

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

```
select count(studentID) as TotalNumberOfSession from Schedule;
```

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

☐ Show all | Number of rows: 25 | Filter rows: Search this table

Extra options

TotalNumberOfSession

5

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

```
select SubjectName, Charge from subject where Charge > 50.00;
```

☐ Profiling [[Edit inline](#)] [[Edit](#)] [[Explain SQL](#)] [[Create PHP code](#)] [[Refresh](#)]

☐ Show all | Number of rows: 25 | Filter rows: Sort by

Extra options

					SubjectName	Charge		
<input type="checkbox"/>		Edit		Copy		Delete	Calculus	65.00
<input type="checkbox"/>		Edit		Copy		Delete	Geometry	55.00
<input type="checkbox"/>		Edit		Copy		Delete	Physics	70.00
<input type="checkbox"/>		Edit		Copy		Delete	Pre-Calculus	60.00

☐ Check all With selected: [Edit](#) [Copy](#) [Delete](#) [Export](#)

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

```
SELECT FamilyID, TotalOwed from family where TotalOwed > 100.00;
```

☐ Profiling [[Edit inline](#)] [[Edit](#)] [[Explain SQL](#)] [[Create PHP code](#)] [[Refresh](#)]

☐ Show all | Number of rows: 25 | Filter rows: S

Extra options

						FamilyID	TotalOwed	
<input type="checkbox"/>		Edit		Copy		Delete	1	120.00
<input type="checkbox"/>		Edit		Copy		Delete	3	200.00

☐ Check all With selected: [Edit](#) [Copy](#) [Delete](#) [Export](#)

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

✓ Showing rows 0 - 0 (1 total, Query took 0.0002 seconds.)

```
SELECT count(StudentID) as NumberOfStudentWithSessionOnMonday from schedule where SessionOccurance like '%Monday%';
```

☐ Profiling [[Edit inline](#)] [[Edit](#)] [[Explain SQL](#)] [[Create PHP code](#)] [[Refresh](#)]

☐ Show all | Number of rows: 25 | Filter rows:

Extra options

NumberOfStudentWithSessionOnMonday
1

☐ Show all | Number of rows: 25 | Filter rows:

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

✓ Showing rows 0 - 0 (1 total, Query took 0.0002 seconds.)

```
SELECT count(TutorID) as NumberOfTutorsEmployed from tutor;
```

☐ Profiling [[Edit inline](#)] [[Edit](#)] [[Explain SQL](#)] [[Create PHP code](#)] [[Refresh](#)]

☐ Show all | Number of rows: 25 | Filter rows:

Extra options

NumberOfTutorsEmployed
5

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

```
1 select t.TutorFirstName, t.TutorLastName, count(distinct s.SubjectName) as NumberOfSubjects
2 from tutor t
3 join Schedule s on s.TutorID = t.TutorID
4 group by t.TutorID
5 having count(distinct s.SubjectName) > 1;
```

☒ Enable foreign key checks

☐ Profiling [[Edit inline](#)] [[Edit](#)] [[Explain SQL](#)] [[Create PHP code](#)] [[Refresh](#)]

☐ Show all | Number of rows: 25 | Filter rows:

Extra options

TutorFirstName	TutorLastName	NumberOfSubjects
John	Doe	2

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

```
1 select t.TutorFirstName, t.TutorLastName, s.SubjectName
2 from tutor t
3 join Schedule s on s.TutorID = t.TutorID
4 where s.SubjectName like '%Pre-Cal%';
```

☒ Enable foreign key checks

Go

Cancel

☐ Profiling [[Edit inline](#)] [[Edit](#)] [[Explain SQL](#)] [[Create PHP code](#)] [[Refresh](#)]



Show all

Number of rows:

25



Filter rows:

Extra options

TutorFirstName	TutorLastName	SubjectName
Michael	Johnson	Pre-Calculus