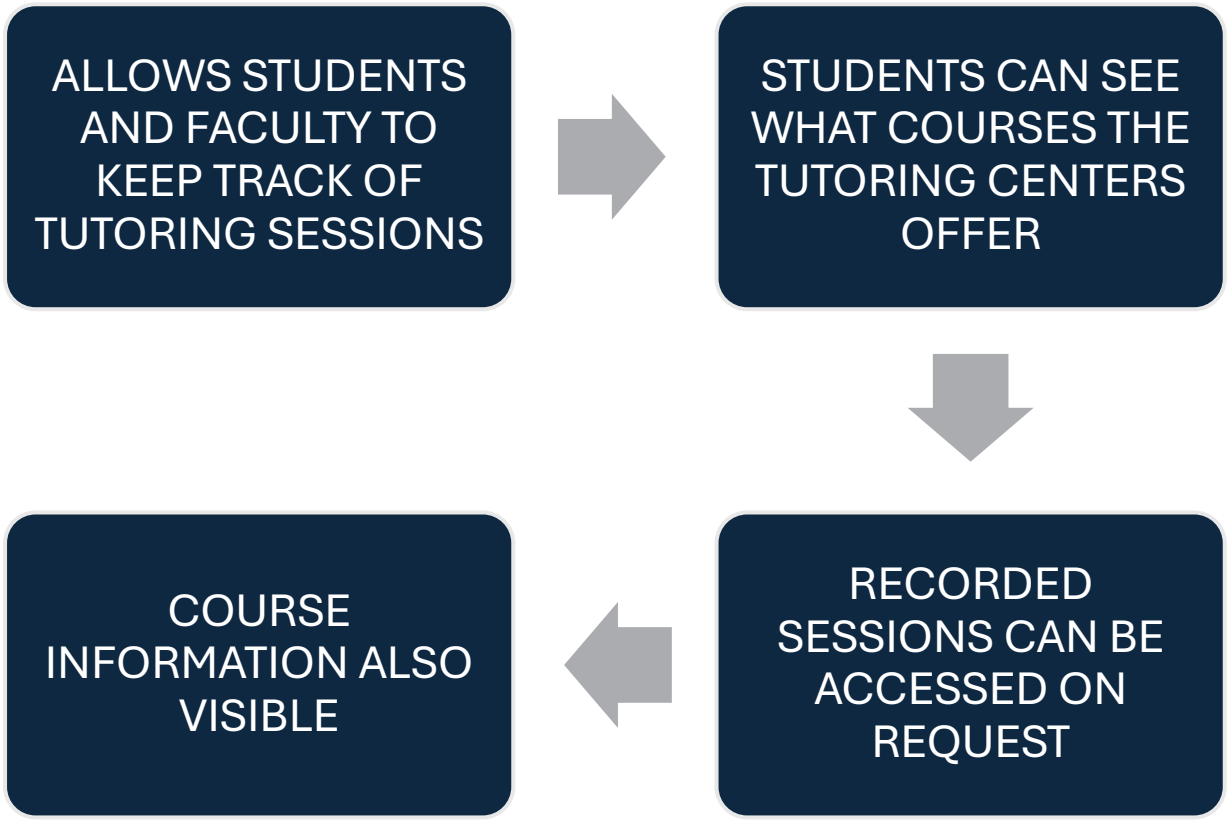
Tutoring Service Application

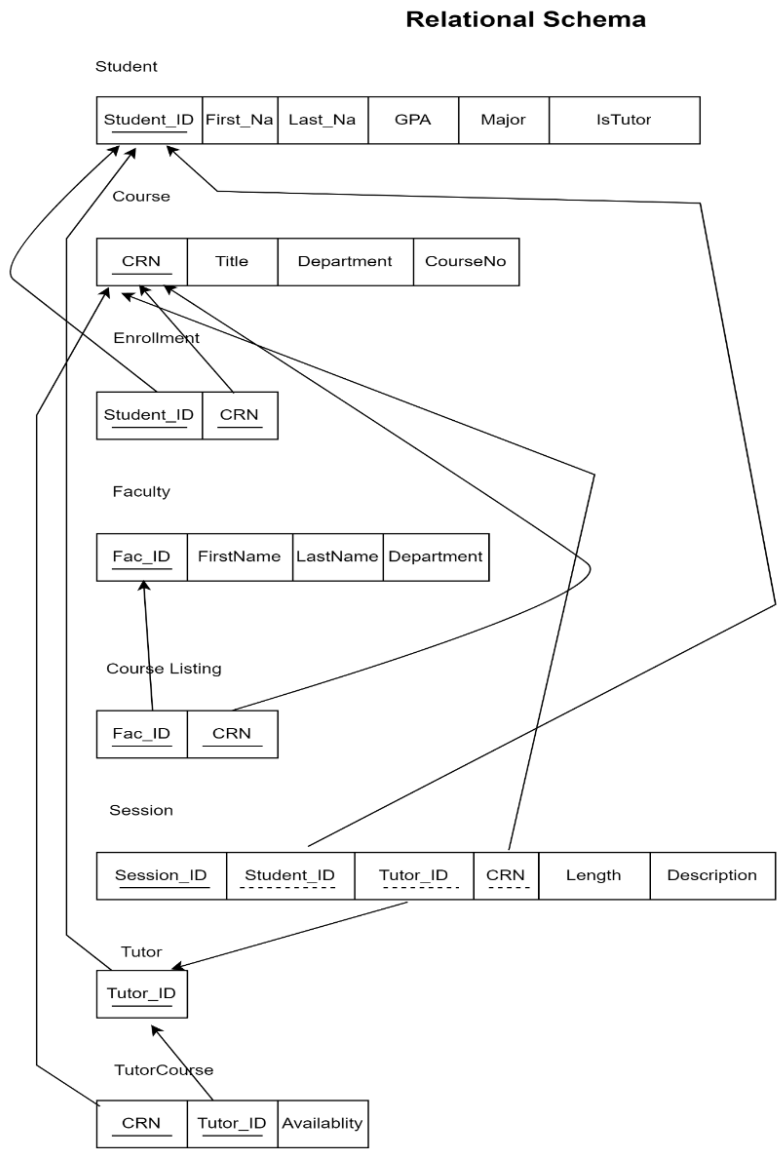
Chad Callender, Patrick Mitchell

May 4, 2023

Application Background

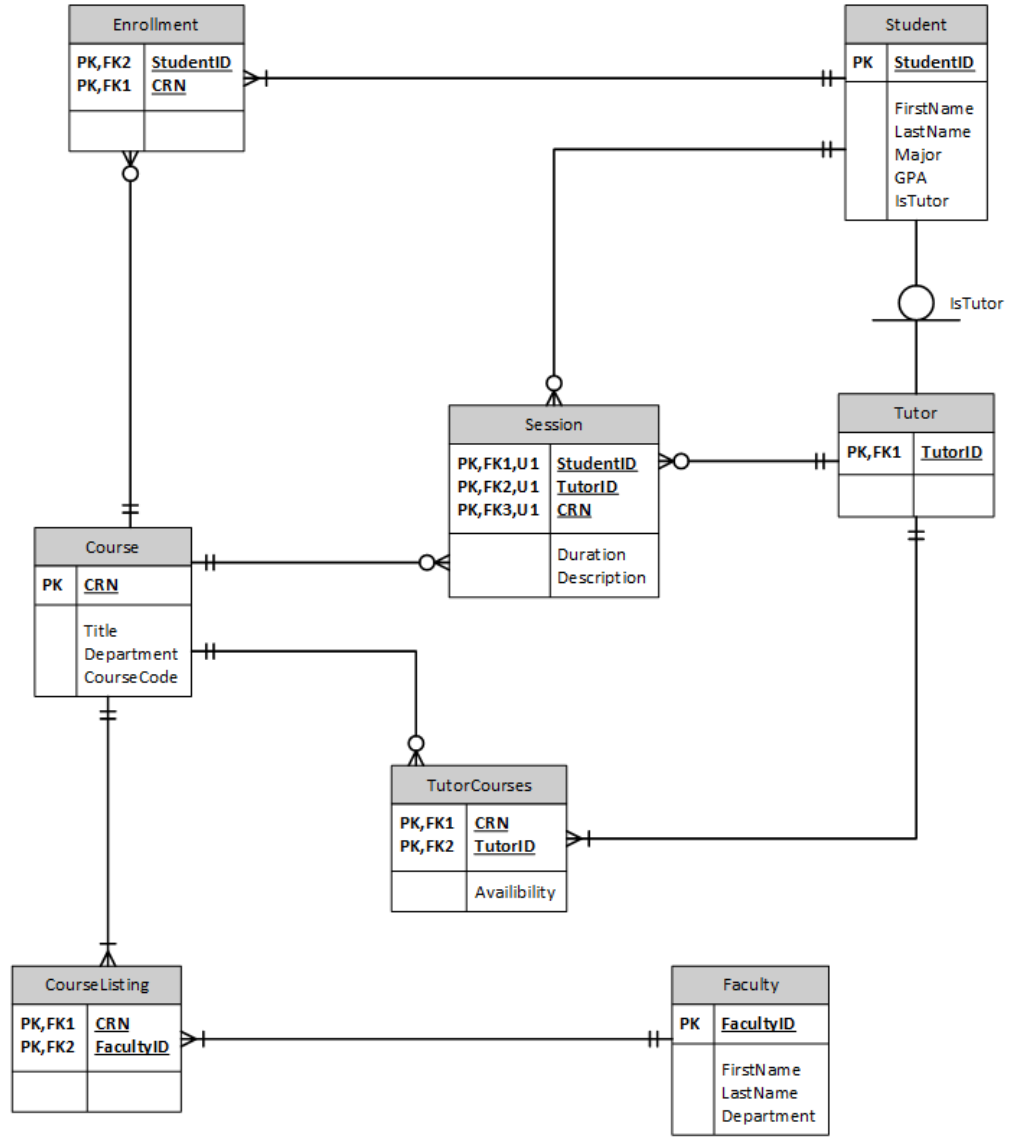


Logical Design



ERD

- Relationships
  - Student must be enrolled in at least one course
  - A tutor is a student; there may be other types
  - A tutor must offer tutoring in at least one course
  - A student may have multiple sessions with multiple tutors for multiple courses
  - A faculty member must teach at least one course



Before we get to it...

Honorable mention to the scripts that saved us time (They're the real MVPs here)

```
/*
c++ script to save time on wriiting insert statements

#include <iostream>
#include <fstream>

using namespace std;

int main()
{
    ifstream in;
    ofstream out;

    int CRN, FacultyID;

    in.open(nums.txt);
    out.open(finally.txt);

    while(in >> CRN, FacultyID)
    {
        out << "insert into courseListing_t (CRN, FacultyID) values (" << CRN << ", " << FacultyID << ")\;";
    }

    in.close();
    out.close();

    return 0;

}
```

```
python script to randomly associate data
import random

crn = [22664, 23142, 24100, 24352, 25287, 26765]

ids = [20270576, 20271293, 20280874]

tups = []

# num_studs = random.randint(1,len(ids))

for course in crn:
  num_studs = random.randint(1,len(ids))
  for i in range(num_studs+1):
    random.shuffle(ids)
    if ((course, ids[0])) not in tups:
      tups.append((course,ids[0]))

# print(tups)
df = open('tups.txt','w')
for tup in tups:
  df.write(f'{tup[0]} {tup[1]}\n')

df.close()
*/
```

Data Defintion

```
CREATE TABLE Student_t(
    StudentID INTEGER NOT NULL,
    FirstName VARCHAR(50) NOT NULL,
    LastName VARCHAR(50) NOT NULL,
    Major VARCHAR(23),
    Department VARCHAR(4),
    GPA DECIMAL(3,2) INVISIBLE,
    IsTutor VARCHAR(10) INVISIBLE,
CONSTRAINT PRIMARY KEY(StudentID));

CREATE TABLE Course_t(
    CRN INTEGER NOT NULL,
    Department VARCHAR(20) NOT NULL,
    CourseCode INTEGER NOT NULL,
    CourseTitle VARCHAR(50),
CONSTRAINT PRIMARY KEY(CRN));

CREATE TABLE Enrollment_t (
    CRN INTEGER NOT NULL,
    StudentID INTEGER NOT NULL,
 CONSTRAINT PRIMARY KEY(StudentID, CRN),
 CONSTRAINT FOREIGN KEY(StudentID) REFERENCES Student_t(StudentID),
CONSTRAINT FOREIGN KEY(CRN) REFERENCES Course_t(CRN));

CREATE TABLE Faculty_t (
    FacultyID INTEGER NOT NULL,
    FirstName VARCHAR(20) NOT NULL,
    LastName VARCHAR(20) NOT NULL,
    Department VARCHAR(4) NOT NULL,
CONSTRAINT PRIMARY KEY(FacultyID));

CREATE TABLE CourseListing_t (
    CRN INTEGER NOT NULL,
    FacultyID INTEGER NOT NULL,
 CONSTRAINT PRIMARY KEY(FacultyID, CRN),
 CONSTRAINT FOREIGN KEY(FacultyID) REFERENCES Faculty_t(FacultyID),
CONSTRAINT FOREIGN KEY(CRN) REFERENCES Course_t(CRN));
```

```
CREATE TABLE Session_t (
    SessionID INTEGER NOT NULL AUTO_INCREMENT,
    StudentID INTEGER NOT NULL,
    TutorID INTEGER NOT NULL,
    CRN INTEGER NOT NULL,
    Date DATE,
    Duration INTEGER CHECK (Duration >= 5),
    Description VARCHAR(50),
CONSTRAINT PRIMARY KEY(SessionID),
CONSTRAINT Session UNIQUE (StudentID, TutorID, CRN, Date),
CONSTRAINT FOREIGN KEY(StudentID) REFERENCES Student_t(StudentID),
CONSTRAINT FOREIGN KEY(TutorID) REFERENCES Student_t(StudentID),
CONSTRAINT FOREIGN KEY(CRN) REFERENCES Course_t(CRN));

CREATE TABLE TutorCourses_t (
    CRN INTEGER NOT NULL,
    TutorID INTEGER NOT NULL,
    Availability VARCHAR(20),
CONSTRAINT PRIMARY KEY(CRN, tutorID),
CONSTRAINT FOREIGN KEY(CRN) REFERENCES Course_t(CRN),
CONSTRAINT FOREIGN KEY(TutorID) REFERENCES student_t(StudentID));

-- TUTOR VIEW
CREATE VIEW tutor_v AS (
    SELECT studentID AS 'tutorID'
    FROM student_t
    WHERE isTutor = 'true');
```

Data Defintion

```
CREATE TABLE Student_t(
    StudentID INTEGER NOT NULL,
    FirstName VARCHAR(50) NOT NULL,
    LastName VARCHAR(50) NOT NULL,
    Major VARCHAR(23),
    Department VARCHAR(4),
    GPA DECIMAL(3,2) INVISIBLE,
    IsTutor VARCHAR(10) INVISIBLE,
CONSTRAINT PRIMARY KEY(StudentID));

CREATE TABLE
    CRN INTEG
    Departmen
    CourseCod
    CourseTit
CONSTRAINT PRIMARY KEY(CRN));

CREATE TABLE Enrollment_t (
    CRN INTEGER NOT NULL,
    StudentID INTEGER NOT NULL,
 CONSTRAINT PRIMARY KEY(StudentID, CRN),
 CONSTRAINT FOREIGN KEY(StudentID) REFERENCES Student_t(StudentID),
CONSTRAINT FOREIGN KEY(CRN) REFERENCES Course_t(CRN));

CREATE TABLE Faculty_t (
    FacultyID INTEGER NOT NULL,
    FirstName VARCHAR(20) NOT NULL,
    LastName VARCHAR(20) NOT NULL,
    Department VARCHAR(4) NOT NULL,
CONSTRAINT PRIMARY KEY(FacultyID));

CREATE TABLE CourseListing_t (
    CRN INTEGER NOT NULL,
    FacultyID INTEGER NOT NULL,
 CONSTRAINT PRIMARY KEY(FacultyID, CRN),
 CONSTRAINT FOREIGN KEY(FacultyID) REFERENCES Faculty_t(FacultyID),
CONSTRAINT FOREIGN KEY(CRN) REFERENCES Course_t(CRN));
```

```
GPA DECIMAL(3,2) INVISIBLE,
IsTutor VARCHAR(10) INVISIBLE,
```

```
CREATE TABLE Session_t (
    SessionID INTEGER NOT NULL AUTO_INCREMENT,
    StudentID INTEGER NOT NULL,
    TutorID INTEGER NOT NULL,
    CRN INTEGER NOT NULL,
    Date DATE,
    Duration INTEGER CHECK (Duration >= 5),
    Description VARCHAR(50),
CONSTRAINT PRIMARY KEY(SessionID),
CONSTRAINT Session UNIQUE (StudentID, TutorID, CRN, Date),
CONSTRAINT FOREIGN KEY(StudentID) REFERENCES Student_t(StudentID),
CONSTRAINT FOREIGN KEY(TutorID) REFERENCES Student_t(StudentID),
CONSTRAINT FOREIGN KEY(CRN) REFERENCES Course_t(CRN));

CREATE TABLE TutorCourses_t (
    CRN INTEGER NOT NULL,
    TutorID INTEGER NOT NULL,
    Availability VARCHAR(20),
CONSTRAINT PRIMARY KEY(CRN, tutorID),
CONSTRAINT FOREIGN KEY(CRN) REFERENCES Course_t(CRN),
CONSTRAINT FOREIGN KEY(TutorID) REFERENCES student_t(StudentID));

-- TUTOR VIEW
CREATE VIEW tutor_v AS (
    SELECT studentID AS 'tutorID'
    FROM student_t
    WHERE isTutor = 'true');
```

Data Defintion

```
CREATE TABLE Student_t(
    StudentID INTEGER NOT NULL,
    FirstName VARCHAR(50) NOT NULL,
    LastName VARCHAR(50) NOT NULL,
    Major VARCHAR(23),
    Department VARCHAR(4),
    GPA DECIMAL(3,2) INVISIBLE,
    IsTutor VARCHAR(10) INVISIBLE,
CONSTRAINT PRIMARY KEY(StudentID));

CREATE TABLE Course_t(
    CRN INTEGER NOT NULL,
    Department VARCHAR(20) NOT NULL,
    CourseCode INTEGER NOT NULL,
    CourseTitle VARCHAR(50),
CONSTRAINT PRIMARY KEY(CRN));

CREATE TABLE Enrollment_t (
    CRN INTEGER NOT NULL,
    StudentID INTEGER NOT NULL,
 CONSTRAINT PRIMARY KEY(StudentID, CRN),
 CONSTRAINT FOREIGN KEY(StudentID) REFERENCES Student_t(StudentID),
CONSTRAINT FOREIGN KEY(CRN) REFERENCES Course_t(CRN));

CREATE TABLE Faculty_t (
    FacultyID INTEGER NOT NULL,
    FirstName VARCHAR(20) NOT NULL,
    LastName VARCHAR(20) NOT NULL,
    Department VARCHAR(4) NOT NULL,
CONSTRAINT PRIMARY KEY(FacultyID));

CREATE TABLE CourseListing_t (
    CRN INTEGER NOT NULL,
    FacultyID INTEGER NOT NULL,
 CONSTRAINT PRIMARY KEY(FacultyID, CRN),
 CONSTRAINT FOREIGN KEY(FacultyID) REFERENCES Faculty_t(FacultyID),
CONSTRAINT FOREIGN KEY(CRN) REFERENCES Course_t(CRN));
```

```
CREATE TABLE Session_t (
    SessionID INTEGER NOT NULL AUTO_INCREMENT,
    StudentID INTEGER NOT NULL,
    TutorID INTEGER NOT NULL,
    CRN INTEGER NO
    Date DATE,
    Duration INTEG
    Description VA
CONSTRAINT PRIMARY
CONSTRAINT Session UNIQUE (StudentID, TutorID, CRN, Date),
CONSTRAINT FOREIGN KEY(StudentID) REFERENCES Student_t(StudentID),
CONSTRAINT FOREIGN KEY(TutorID) REFERENCES Student_t(StudentID),
CONSTRAINT FOREIGN KEY(CRN) REFERENCES Course_t(CRN));

CREATE TABLE TutorCourses_t (
    CRN INTEGER NOT NULL,
    TutorID INTEGER NOT NULL,
    Availability VARCHAR(20),
CONSTRAINT PRIMARY KEY(CRN, tutorID),
CONSTRAINT FOREIGN KEY(CRN) REFERENCES Course_t(CRN),
CONSTRAINT FOREIGN KEY(TutorID) REFERENCES student_t(StudentID));

-- TUTOR VIEW
CREATE VIEW tutor_v AS (
    SELECT studentID AS 'tutorID'
    FROM student_t
    WHERE isTutor = 'true');
```

```
ATE TABLE Session_t (
 SessionID INTEGER NOT NULL AUTO_INCREMENT,
 StudentID INTEGER NOT NULL,
 TutorID INTEGER NOT NULL,
```

Data Defintion

```
CREATE TABLE Student_t(
    StudentID INTEGER NOT NULL,
    FirstName VARCHAR(50) NOT NULL,
    LastName VARCHAR(50) NOT NULL,
    Major VARCHAR(23),
    Department VARCHAR(4),
    GPA DECIMAL(3,2) INVISIBLE,
    IsTutor VARCHAR(10) INVISIBLE,
CONSTRAINT PRIMARY KEY(StudentID));

CREATE TABLE Course_t(
    CRN INTEGER NOT NULL,
    Department VARCHAR(20) NOT NULL,
    CourseCode INTEGER NOT NULL,
    CourseTitle VARCHAR(50),
CONSTRAINT PRIMARY KEY(CRN));

CREATE TABLE Enrollment_t (
    CRN INTEGER NOT NULL,
    StudentID INTEGER NOT NULL,
 CONSTRAINT PRIMARY KEY(StudentID, CRN),
 CONSTRAINT FOREIGN KEY(StudentID) REFERENCES Student_t(StudentID),
CONSTRAINT FOREIGN KEY(CRN) REFERENCES Course_t(CRN));

CREATE TABLE Faculty_t (
    FacultyID INTEGER NOT NULL,
    FirstName VARCHAR(20) NOT NULL,
    LastName VARCHAR(20) NOT NULL,
    Department VARCHAR(4) NOT NULL,
CONSTRAINT PRIMARY KEY(FacultyID));

CREATE TABLE CourseListing_t (
    CRN INTEGER NOT NULL,
    FacultyID INTEGER NOT NULL,
 CONSTRAINT PRIMARY KEY(FacultyID, CRN),
 CONSTRAINT FOREIGN KEY(FacultyID) REFERENCES Faculty_t(FacultyID),
CONSTRAINT FOREIGN KEY(CRN) REFERENCES Course_t(CRN));
```

```
Date DATE,
Duration INTEGER CHECK (Duration >= 5),
```

```
CREATE TABLE Session_t (
    SessionID INTEGER NOT NULL AUTO_INCREMENT,
    StudentID INTEGER NOT NULL,
    TutorID INTEGER NOT NULL,
    CRN INTEGER NOT NULL,
    Date DATE,
    Duration INTEGER CHECK (Duration >= 5),
    Description VARCHAR(50),
    RY KEY(SessionID),
    on UNIQUE (StudentID, TutorID, CRN, Date),
    GN KEY(StudentID) REFERENCES Student_t(StudentID),
CONSTRAINT FOREIGN KEY(TutorID) REFERENCES Student_t(StudentID),
CONSTRAINT FOREIGN KEY(CRN) REFERENCES Course_t(CRN));

CREATE TABLE TutorCourses_t (
    CRN INTEGER NOT NULL,
    TutorID INTEGER NOT NULL,
    Availability VARCHAR(20),
CONSTRAINT PRIMARY KEY(CRN, tutorID),
CONSTRAINT FOREIGN KEY(CRN) REFERENCES Course_t(CRN),
CONSTRAINT FOREIGN KEY(TutorID) REFERENCES student_t(StudentID));

-- TUTOR VIEW
CREATE VIEW tutor_v AS (
    SELECT studentID AS 'tutorID'
    FROM student_t
    WHERE isTutor = 'true');
```

Data Defintion

```
CREATE TABLE Student_t(
    StudentID INTEGER NOT NULL,
    FirstName VARCHAR(50) NOT NULL,
    LastName VARCHAR(50) NOT NULL,
    Major VARCHAR(23),
    Department VARCHAR(4),
    GPA DECIMAL(3,2) INVISIBLE,
    IsTutor VARCHAR(10) INVISIBLE,
CONSTRAINT PRIMARY KEY(StudentID));

CREATE TABLE Course_t(
    CRN INTEGER NOT NULL,
    Department VARCHAR(20) NOT NULL,
    CourseCode INTEGER NOT NULL,
    CourseTitle VARCHAR
CONSTRAINT PRIMARY KEY

CREATE TABLE Enrollmen
    CRN INTEGER NOT NU
    StudentID INTEGER NOT NULL,
 CONSTRAINT PRIMARY KEY(StudentID, CRN),
 CONSTRAINT FOREIGN KEY(StudentID) REFERENCES Student_t(StudentID),
CONSTRAINT FOREIGN KEY(CRN) REFERENCES Course_t(CRN));

CREATE TABLE Faculty_t (
    FacultyID INTEGER NOT NULL,
    FirstName VARCHAR(20) NOT NULL,
    LastName VARCHAR(20) NOT NULL,
    Department VARCHAR(4) NOT NULL,
CONSTRAINT PRIMARY KEY(FacultyID));

CREATE TABLE CourseListing_t (
    CRN INTEGER NOT NULL,
    FacultyID INTEGER NOT NULL,
 CONSTRAINT PRIMARY KEY(FacultyID, CRN),
 CONSTRAINT FOREIGN KEY(FacultyID) REFERENCES Faculty_t(FacultyID),
CONSTRAINT FOREIGN KEY(CRN) REFERENCES Course_t(CRN));
```

```
CONSTRAINT PRIMARY KEY(SessionID),
CONSTRAINT Session UNIQUE (StudentID, TutorID, CRN, Date),
```

```
CREATE TABLE Session_t (
    SessionID INTEGER NOT NULL AUTO_INCREMENT,
    StudentID INTEGER NOT NULL,
    TutorID INTEGER NOT NULL,
    CRN INTEGER NOT NULL,
    Date DATE,
    Duration INTEGER CHECK (Duration >= 5),
    Description VARCHAR(50),
CONSTRAINT PRIMARY KEY(SessionID),
CONSTRAINT Session UNIQUE (StudentID, TutorID, CRN, Date),
CONSTRAINT FOREIGN KEY(StudentID) REFERENCES Student_t(StudentID),
                                  EFERENCES Student_t(StudentID),
                                  NCES Course_t(CRN));

    CRN INTEGER NOT NULL,
    TutorID INTEGER NOT NULL,
    Availability VARCHAR(20),
CONSTRAINT PRIMARY KEY(CRN, tutorID),
CONSTRAINT FOREIGN KEY(CRN) REFERENCES Course_t(CRN),
CONSTRAINT FOREIGN KEY(TutorID) REFERENCES student_t(StudentID));

-- TUTOR VIEW
CREATE VIEW tutor_v AS (
    SELECT studentID AS 'tutorID'
    FROM student_t
    WHERE isTutor = 'true');
```

Data Defintion

```
CREATE TABLE Student_t(
    StudentID INTEGER NOT NULL,
    FirstName VARCHAR(50) NOT NULL,
    LastName VARCHAR(50) NOT NULL,
    Major VARCHAR(23),
    Department VARCHAR(4),
    GPA DECIMAL(3,2) INVISIBLE,
    IsTutor VARCHAR(10) INVISIBLE,
CONSTRAINT PRIMARY KEY(StudentID));

CREATE TABLE Course_t(
    CRN INTEGER NOT NULL,
    Department VARCHAR(20) NOT NULL,
    CourseCode INTEGER NOT NULL,
    CourseTitle VARCHAR(50)
CONSTRAINT PRIMARY KEY(

CREATE TABLE Enrollment
    CRN INTEGER NOT NUL
    StudentID INTEGER N
 CONSTRAINT PRIMARY KEY(
 CONSTRAINT FOREIGN KEY(
CONSTRAINT FOREIGN KEY(

CREATE TABLE Faculty_t
    FacultyID INTEGER N
    FirstName VARCHAR(2
    LastName VARCHAR(20) NOT NULL,
    Department VARCHAR(4) NOT NULL,
CONSTRAINT PRIMARY KEY(FacultyID));

CREATE TABLE CourseListing_t (
    CRN INTEGER NOT NULL,
    FacultyID INTEGER NOT NULL,
 CONSTRAINT PRIMARY KEY(FacultyID, CRN),
 CONSTRAINT FOREIGN KEY(FacultyID) REFERENCES Faculty_t(FacultyID),
CONSTRAINT FOREIGN KEY(CRN) REFERENCES Course_t(CRN));
```

```
-- TUTOR VIEW
CREATE VIEW tutor_v AS (
    SELECT studentID AS 'tutorID'
    FROM student_t
    WHERE isTutor = 'true');
```

```
CREATE TABLE Session_t (
    SessionID INTEGER NOT NULL AUTO_INCREMENT,
    StudentID INTEGER NOT NULL,
    TutorID INTEGER NOT NULL,
    CRN INTEGER NOT NULL,
    Date DATE,
    Duration INTEGER CHECK (Duration >= 5),
    Description VARCHAR(50),
CONSTRAINT PRIMARY KEY(SessionID),
CONSTRAINT Session UNIQUE (StudentID, TutorID, CRN, Date),
CONSTRAINT FOREIGN KEY(StudentID) REFERENCES Student_t(StudentID),
AINT FOREIGN KEY(TutorID) REFERENCES Student_t(StudentID),
AINT FOREIGN KEY(CRN) REFERENCES Course_t(CRN));

 TABLE TutorCourses_t (
N INTEGER NOT NULL,
torID INTEGER NOT NULL,
ailability VARCHAR(20),
AINT PRIMARY KEY(CRN, tutorID),
AINT FOREIGN KEY(CRN) REFERENCES Course_t(CRN),
AINT FOREIGN KEY(TutorID) REFERENCES student_t(StudentID));

-- TUTOR VIEW
CREATE VIEW tutor_v AS (
    SELECT studentID AS 'tutorID'
    FROM student_t
    WHERE isTutor = 'true');
```

Tables and Views

```
+-----------+-----------+------------+------------------------+------------+
| StudentID | FirstName | LastName   | Major                  | Department |
+-----------+-----------+------------+------------------------+------------+
|  20206275 | Guthrie   | Dargan     | Business Administration | BUAD       |
|  20208477 | Fan       | Smallcombe | Biology                | BIOL       |
|  20209667 | Jackelyn  | Craigg     | Business Administration | BUAD       |
|  20215007 | Oona      | Laxen      | Business Administration | BUAD       |
|  20218294 | Jolyn     | Zieme      | Political Science      | POLS       |
|  20224532 | Babb      | Wylder     | Biology                | BIOL       |
|  20225509 | Tamiko    | Lewing     | Computer Science       | CMPS       |
|  20227594 | Gregorius | Trask      | Mechanical Engineering | MENG       |
|  20228794 | Rriocard  | Titley     | Computer Science       | CMPS       |
|  20230527 | Teodorico | Cranch     | Business Administration | BUAD       |
+-----------+-----------+------------+------------------------+------------+
10 rows in set (0.00 sec)

mysql> select * from course_t limit 5;
+-------+------------+------------+-------------------------------+
| CRN   | Department | CourseCode | CourseTitle                   |
+-------+------------+------------+-------------------------------+
| 20169 | NURS       |       1143 | Introduction to Nursing       |
| 22664 | ENGL       |       1143 | Intro Crit Read & Acad Writing |
| 23011 | MATH       |       1243 | Plane Trigonometry            |
| 23142 | ENGL       |       1143 | Intro Crit Read & Acad Writing |
| 23503 | NURS       |       2284 | Mother and Baby               |
+-------+------------+------------+-------------------------------+
5 rows in set (0.00 sec)
```

Student_t, course_t

```
mysql> select * from faculty_t limit 5;
+-----------+-----------+------------+------------+
| FacultyID | FirstName | LastName   | Department |
+-----------+-----------+------------+------------+
|  20005093 | Karil     | Millthorpe | MATH       |
|  20023140 | Evanne    | Iles       | NURS       |
|  20027880 | Imogene   | Le Fleming | CMPS       |
|  20028804 | Ashely    | McIlmorie  | BIOL       |
|  20039236 | Brig      | Benion     | MENG       |
+-----------+-----------+------------+------------+
5 rows in set (0.00 sec)

mysql> select * from tutor_t limit 5;
ERROR 1146 (42S02): Table 'tutoring.tutor_t' doesn't exist
mysql> select * from tutor_v limit 5;
+----------+
| tutorID  |
+----------+
| 20208477 |
| 20215007 |
| 20230527 |
| 20230929 |
| 20232283 |
+----------+
5 rows in set (0.00 sec)
```

Faculty_t, tutor_v

```
mysql> select count(*) from enrollment_t;
+----------+
| count(*) |
+----------+
|      202 |
+----------+
1 row in set (0.01 sec)

mysql> select count(*) from session_t;
+----------+
| count(*) |
+----------+
|      275 |
+----------+
1 row in set (0.00 sec)
```

Enrollment_t, session_t counts

Data Manipulation (SQL Queries)

Students enrolled in Calculus I by course title and course code

```sql
-- 1a. Students enrolled in a particular course code:

SELECT s.studentID, FirstName, LastName
FROM student_t s, enrollment_t e, course_t c
wHERE e.crn = c.crn AND e.studentid = s.studentid
AND c.coursecode = 2214;

-- 1b. Students enrolled in a particular coursetitle:

SELECT s.studentID, FirstName, LastName
FROM student_t s, enrollment_t e, course_t c
wHERE e.crn = c.crn AND e.studentid = s.studentid
AND c.coursetitle = 'Calculus I';
```

```
mysql> -- 1a. Students enrolled in a particular course code:
mysql>
mysql> SELECT s.studentID, FirstName, LastName
    -> FROM student_t s, enrollment_t e, course_t c
    -> wHERE e.crn = c.crn AND e.studentid = s.studentid
    -> AND c.coursecode = 2214;
+-----------+-----------+--------------+
| studentID | FirstName | LastName     |
+-----------+-----------+--------------+
|  20225509 | Tamiko    | Lewing       |
|  20228794 | Rriocard  | Titley       |
|  20270195 | Ladonna   | Shyre        |
|  20279328 | Pearce    | Maggorini    |
|  20280052 | Milly     | Basini-Gazzi |
|  20299748 | Aile      | Carey        |
+-----------+-----------+--------------+
6 rows in set (0.00 sec)

mysql>
mysql> -- 1b. Students enrolled in a particular coursetitle:
mysql>
mysql> SELECT s.studentID, FirstName, LastName
    -> FROM student_t s, enrollment_t e, course_t c
    -> wHERE e.crn = c.crn AND e.studentid = s.studentid
    -> AND c.coursetitle = 'Calculus I';
+-----------+-----------+--------------+
| studentID | FirstName | LastName     |
+-----------+-----------+--------------+
|  20225509 | Tamiko    | Lewing       |
|  20228794 | Rriocard  | Titley       |
|  20270195 | Ladonna   | Shyre        |
|  20279328 | Pearce    | Maggorini    |
|  20280052 | Milly     | Basini-Gazzi |
|  20299748 | Aile      | Carey        |
+-----------+-----------+--------------+
6 rows in set (0.00 sec)
```

Courses with no students enrolled

```sql
-- 2. Courses that have no enrollment:

SELECT crn, coursecode, coursetitle
FROM course_t
WHERE crn NOT IN
(SELECT distinct crn FROM enrollment_t);
```

```
mysql> SELECT crn, coursecode, coursetitle
    -> FROM course_t
    -> WHERE crn NOT IN
    -> (SELECT distinct crn FROM enrollment_t);
+-------+------------+--------------------------------+
| crn   | coursecode | coursetitle                    |
+-------+------------+--------------------------------+
| 23011 |       1243 | Plane Trigonometry             |
| 25514 |       3313 | Physical Chemistry             |
| 26660 |       3313 | Intro Read & Writ - Literature |
| 29494 |       2563 | Fluid Dynamics                 |
+-------+------------+--------------------------------+
4 rows in set (0.00 sec)
```

Tutor courses with students enrolled

```sql
-- 5. Courses offered at the tutoring center that have students enrolled
SELECT c.crn, tutorID, firstName Tutor, courseTitle
FROM student_t s, tutorcourses_t tc, course_t c
WHERE studentID = tutorID
    AND tc.crn = c.crn
    AND courseTitle NOT IN (select courseTitle from course_t where crn not in (select crn from enrollment_t))
ORDER BY courseTitle;
```

```
mysql> SELECT c.crn, tutorID, firstName Tutor, courseTitle
    -> FROM student_t s, tutorcourses_t tc, course_t c
    -> WHERE studentID = tutorID
    -> AND tc.crn = c.crn
    -> AND courseTitle NOT IN (select courseTitle from course_t where
    -> ORDER BY courseTitle;
+-------+----------+-----------+---------------------------------+
| crn   | tutorID  | Tutor     | courseTitle                     |
+-------+----------+-----------+---------------------------------+
| 26567 | 20264548 | Emmalynne | Calculus I                      |
| 26567 | 20293929 | Chaim     | Calculus I                      |
| 26567 | 20299748 | Aile      | Calculus I                      |
| 25313 | 20272587 | Adi       | Calculus II                     |
| 24077 | 20208477 | Fan       | Cells II                        |
| 24077 | 20264548 | Emmalynne | Cells II                        |
| 25287 | 20264548 | Emmalynne | College Algebra                 |
| 25287 | 20272587 | Adi       | College Algebra                 |
| 25287 | 20293929 | Chaim     | College Algebra                 |
| 28816 | 20299748 | Aile      | Computer Science I              |
| 26765 | 20215007 | Oona      | Foundations of Business         |
| 26765 | 20230527 | Teodorico | Foundations of Business         |
| 26986 | 20208477 | Fan       | General Chemistry I             |
| 26986 | 20293929 | Chaim     | General Chemistry I             |
| 23142 | 20208477 | Fan       | Intro Crit Read & Acad Writing  |
| 23142 | 20215007 | Oona      | Intro Crit Read & Acad Writing  |
| 23890 | 20272587 | Adi       | Intro to Mechanical Engineering |
| 23890 | 20293929 | Chaim     | Intro to Mechanical Engineering |
| 20169 | 20232283 | Horatio   | Introduction to Nursing         |
| 20169 | 20258701 | Wanids    | Introduction to Nursing         |
| 24370 | 20298247 | Charmion  | Introduction to Nursing         |
+-------+----------+-----------+---------------------------------+
21 rows in set (0.00 sec)
```

Number of sessions by department

```sql
-- 4. Metrics for sessions by department
SELECT department, COUNT(department) 'Number of Sessions' ,
    SUM(s.duration) 'Duration in Minutes',
    SUM(duration) / COUNT(department) 'Average Session Length'
FROM session_t s INNER JOIN course_t c ON s.crn = c.crn
GROUP BY department
ORDER BY SUM(duration) desc;
```

```
mysql> SELECT department, COUNT(department) 'Number of Sessions' , SUM(s.duration) '
    -> FROM session_t s INNER JOIN course_t c ON s.crn = c.crn
    -> GROUP BY department
    -> ORDER BY SUM(duration) desc;
+------------+--------------------+--------------------+-----------------------+
| department | Number of Sessions | Duration in Minutes | Average Session Length |
+------------+--------------------+--------------------+-----------------------+
| MATH       |                122 |               4173 |               34.2049 |
| CHEM       |                 45 |               1675 |               37.2222 |
| ENGL       |                 38 |               1335 |               35.1316 |
| BUAD       |                 30 |               1058 |               35.2667 |
| NURS       |                 23 |                766 |               33.3043 |
| CMPS       |                 17 |                467 |               27.4706 |
+------------+--------------------+--------------------+-----------------------+
6 rows in set (0.00 sec)
```

Total time students spent in tutoring for a particular course

```sql
-- 3. The total time spent in tutor session for a particular course

SELECT studentid, SUM(duration) 'Session Length in minutes'
FROM session_t
WHERE crn IN (select crn from course_t where courseTitle = 'Computer Science I')
group by studentid;
```

```
mysql> SELECT studentid, SUM(duration) 'Session Length in minutes'
    -> FROM session_t
    -> WHERE crn IN (select crn from course_t where courseTitle = 'Computer Science I')
    -> group by studentid;
+-----------+---------------------------+
| studentid | Session Length in minutes |
+-----------+---------------------------+
|  20244742 |                       123 |
|  20246928 |                       167 |
|  20270576 |                        40 |
|  20259851 |                         6 |
|  20270195 |                       131 |
+-----------+---------------------------+
5 rows in set (0.00 sec)
```