

Group Account: KRC353_1

Faculty of Engineering and Computer Science Expectations of Originality

This form sets out the requirements for originality for work submitted by students in the Faculty of Engineering and Computer Science. Submissions such as assignments, lab reports, project reports, computer programs and take-home exams must conform to the requirements stated on this form and to the Academic Code of Conduct. The course outline may stipulate additional requirements for the course.

1. Your submissions must be your own original work. Group submissions must be the original work of the students in the group.
2. Direct quotations must not exceed 5% of the content of a report, must be enclosed in quotation marks, and must be attributed to the source by a numerical reference citation¹. Note that engineering reports rarely contain direct quotations.
3. Material paraphrased or taken from a source must be attributed to the source by a numerical reference citation.
4. Text that is inserted from a web site must be enclosed in quotation marks and attributed to the web site by numerical reference citation.
5. Drawings, diagrams, photos, maps or other visual material taken from a source must be attributed to that source by a numerical reference citation.
6. No part of any assignment, lab report or project report submitted for this course can be submitted for any other course.
7. In preparing your submissions, the work of other past or present students cannot be consulted, used, copied, paraphrased or relied upon in any manner whatsoever.
8. Your submissions must consist entirely of your own or your group's ideas, observations, calculations, information and conclusions, except for statements attributed to sources by numerical citation.
9. Your submissions cannot be edited or revised by any other student.
10. For lab reports, the data must be obtained from your own or your lab group's experimental work.
11. For software, the code must be composed by you or by the group submitting the work, except for code that is attributed to its sources by numerical reference.

You must write one of the following statements on each piece of work that you submit:

For individual work: **"I certify that this submission is my original work and meets the Faculty's Expectations of Originality"**, with your signature, I.D. #, and the date.

For group work: **"We certify that this submission is the original work of members of the group and meets the Faculty's Expectations of Originality"**, with the signatures and I.D. #s of all the team members and the date.

A signed copy of this form must be submitted to the instructor at the beginning of the semester in each course.

I certify that I have read the requirements set out on this form, and that I am aware of these requirements. I certify that all the work I will submit for this course will comply with these requirements and with additional requirements stated in the course outline.

Course Number: COMP 353
Name: Deniz Akca
Signature: Deniz Akca

Instructor: Khaled Jabado
I.D. #: 26252978
Date: Wednesday, July 17, 2019

Iana Belitchka 40032171 Marc Hegedus

¹ Rules for reference citation can be found in "Form and Style" by Patrich MacDonagh and Jack Bordan, fourth edition, May, 2000, available at <http://www.encs.concordia.ca/scs/Forms/Form&Style.pdf>.

Approved by the ENCS Faculty Council February 10, 2012

David-Etienne Pigeon

40068000

David K...

Luiz Goncalves

26943799

Luiz

Liam Blount-Castagnay

40152713

Liam

1) ER Diagram

Requirement: The E/R diagram of the design of the database given in the project description (or a revised version, if deemed necessary).

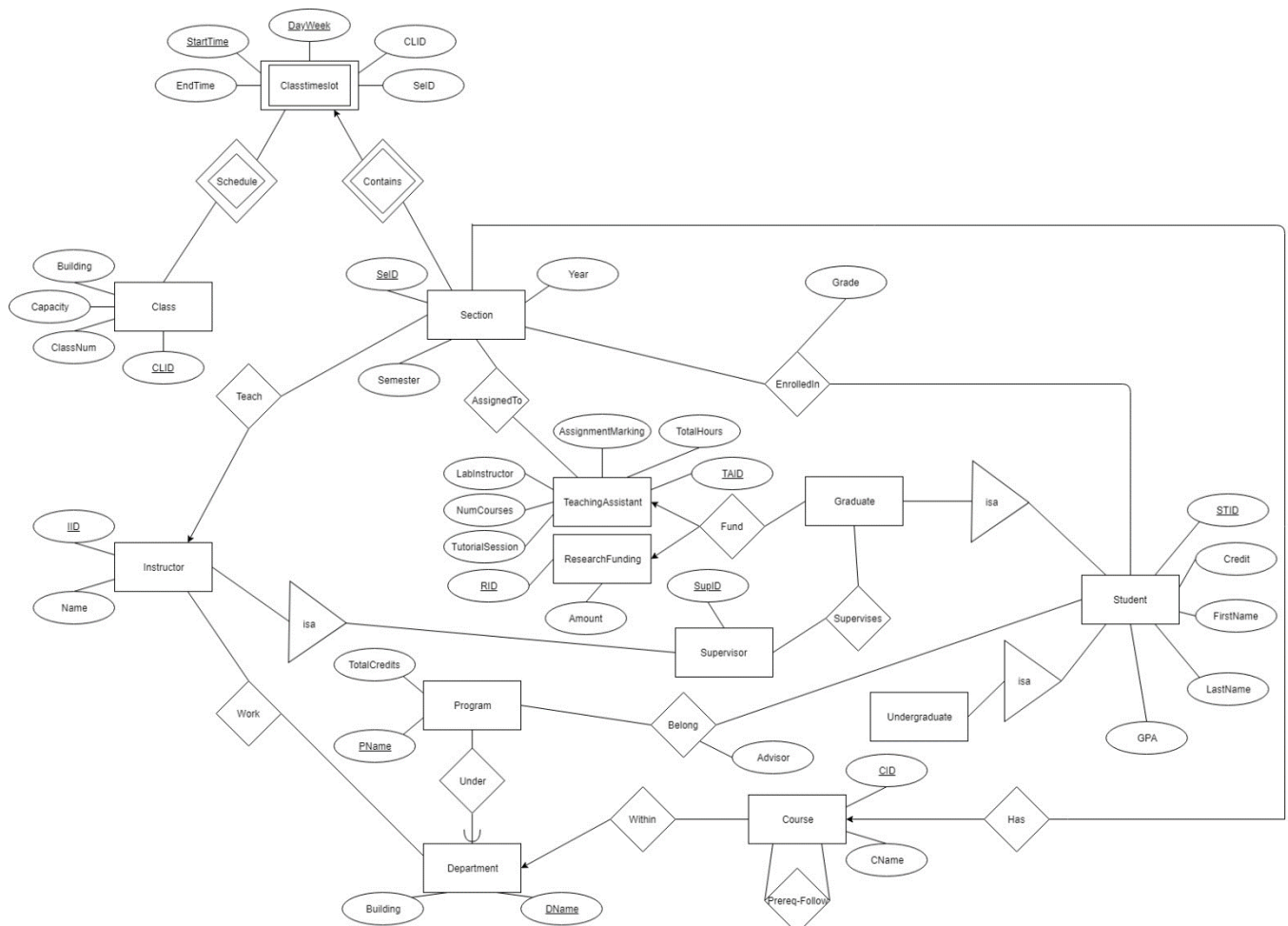


Figure 1: E/R Diagram as described in the project handout. A .JPEG has been included in the .ZIP for clarity.

2) Creating the Database

Requirements: The SQL statements formulated and used to create the database. Pick appropriate data types for the attributes and include them in your report.

The following **CREATE TABLE** statements are in alphabetical order:

```
CREATE TABLE AssignTo
(
    SeID INT,
    TAID INT,
    FOREIGN KEY (SeID) REFERENCES Section (SeID),
    FOREIGN KEY (TAID) REFERENCES TeachingAssistant (TAID)
);
```

```
CREATE TABLE Belong
(
  STID INT,
  PName CHAR (30),
  Advisor CHAR (30),
  FOREIGN KEY (STID) REFERENCES Student (STID),
  FOREIGN KEY (PName) REFERENCES Program (PName)
);
```

```
CREATE TABLE Class
(
  CLID INT PRIMARY KEY,
  ClassNum INT,
  Building CHAR (30),
  Capacity INT
);
```

```
CREATE TABLE ClassTimeslot
(
  StartTime TIME,
  DayWeek CHAR (30),
  CLID INT,
  SeID INT,
  EndTime TIME,
  PRIMARY KEY (StartTime, DayWeek),
  FOREIGN KEY (CLID) REFERENCES Class (CLID),
  FOREIGN KEY (SeID) REFERENCES Section (SeID)
);
```

```
CREATE TABLE Course
(
  CID INT PRIMARY KEY,
  CName CHAR (30) REFERENCES Class (CName),
);
```

```
CREATE TABLE Department
(
  DName CHAR (30) PRIMARY KEY,
  Building CHAR (30)
);
```

```
CREATE TABLE EnrolledIn
(
    STID INT,
    SeID INT,
    Grade CHAR (2),
    FOREIGN KEY (SeID) REFERENCES Section (SeID),
    FOREIGN KEY (STID) REFERENCES Student (STID)
);
```

```
CREATE TABLE Fund
(
    STID INT,
    TAID INT,
    RID INT,
    FOREIGN KEY (STID) REFERENCES Graduate (STID),
    FOREIGN KEY (TAID) REFERENCES TeachingAssistant (TAID),
    FOREIGN KEY (RID) REFERENCES ResearchFunding (RID)
);
```

```
CREATE TABLE Graduate
(
    STID INT PRIMARY KEY,
    FOREIGN KEY (STID) REFERENCES Student (STID)
);
```

```
CREATE TABLE Has
(
    CID INT,
    SeID INT,
    FOREIGN KEY (CID) REFERENCES Course (CID),
    FOREIGN KEY (SeID) REFERENCES Section (SeID)
);
```

```
CREATE TABLE Instructor
(
    IID INT PRIMARY KEY,
    Name CHAR(30),
    Supervised INT
);
```

```
CREATE TABLE Program
(
  PName CHAR (30) PRIMARY KEY,
  TotalCredits INT
);
```

```
CREATE TABLE ResearchFunding
(
  RID INT PRIMARY KEY,
  Amount INT
);
```

```
CREATE TABLE Section
(
  SeID INT PRIMARY KEY,
  Semester CHAR (30),
  Year INT
);
```

```
CREATE TABLE Student
(
  STID INT PRIMARY KEY,
  Credit INT,
  FirstName CHAR (30),
  LastName CHAR (30),
  GPA FLOAT
);
```

```
CREATE TABLE Supervises
(
  SupID INT,
  STID INT,
  FOREIGN KEY (SupID) REFERENCES Supervisor (SupID),
  FOREIGN KEY (STID) REFERENCES Graduate (STID)
);
```

```
CREATE TABLE Supervisor
(
  SupID INT PRIMARY KEY,
  IID INT,
  FOREIGN KEY (IID) REFERENCES Instructor (IID)
);
```

```
CREATE TABLE Teach
(
    SeID INT,
    IID INT,
    FOREIGN KEY (SeID) REFERENCES Section (SeID),
    FOREIGN KEY (IID) REFERENCES Instructor (IID)
);
```

```
CREATE TABLE TeachingAssistant
(
    TAID INT PRIMARY KEY,
    TotalHours INT,
    AssignmentMarking CHAR(1),
    LabInstructor CHAR(1),
    NumCourses INT,
    TutorialSession CHAR(1)
);
```

```
CREATE TABLE Under
(
    DName CHAR(30),
    PName CHAR(30),
    FOREIGN KEY (DName) REFERENCES Department (DName),
    FOREIGN KEY (PName) REFERENCES Program (PName)
);
```

```
CREATE TABLE Undergraduate
(
    STID INT PRIMARY KEY,
    FOREIGN KEY (STID) REFERENCES Student (STID)
);
```

```
CREATE TABLE Within
(
    CID INT,
    DName CHAR(30) REFERENCES Department (DName),
    FOREIGN KEY (CID) REFERENCES Course (CID)
);
```

```
CREATE TABLE Work
(
    DName INT,
    IID INT,
    FOREIGN KEY (DName) REFERENCES Department (DName),
    FOREIGN KEY (IID) REFERENCES Instructor (IID)
);
```

3) SQL Queries

Requirements: The SQL statements formulated to express the required queries and transactions mentioned.

Query I:

```
SELECT S.STID as Student_ID, S.FirstName as First_name, S.LastName as Last_name
FROM Student S
INNER JOIN EnrolledIn E ON S.STID = E.STID
INNER JOIN Section Se ON E.SeID = Se.SeID
INNER JOIN Has ON Has.SeID=Se.SeID
INNER JOIN Course C ON Has.CID = C.CID
WHERE C.CName='COMP352' AND (E.Grade='A' OR E.Grade='A+');
```

Result of Query I:

Student_ID	First_name	Last_name
11	Khaled	Jababo
12	Aiman	Hanna
19	Nathan	Mackinnon
25	Smitv	Sam
2	James	Watson

Query II:

```
SELECT Student.STID as Student_ID, FirstName as First_name, LastName as
Last_name, COUNT(Program.PName) as Number_of_Programs
FROM Student
INNER JOIN Belong B ON Student.STID = B.STID
INNER JOIN Program ON B.PName = Program.PName
GROUP BY Student.STID
HAVING COUNT(Program.PName)>1;
```

Result of Query II:

Student_ID	First_name	Last_name	Number_of_Programs
1	Felix	Harris	2

Query III:

```
SELECT I.Name, C.CName, S.Year, S.Semester, I.IID, S.SeID, C.CID
FROM Instructor I
INNER JOIN Teach T ON T.IID = I.IID
INNER JOIN Section S ON T.SeID = S.SeID
INNER JOIN Has H ON S.SeID = H.SeID
INNER JOIN Course C ON H.CID = C.CID
WHERE C.CName = 'COMP352' AND S.Year = 2018 AND S.Semester = 'Fall';
```

Result of Query III:

Name	CName	Year	Semester	IID	SeID	CID
Nicholas Larsen	COMP352	2018	Fall	10		1
Ernest Steig	COMP352	2018	Fall	2	1	1

Query IV:

```
SELECT DISTINCT P.PName as Program, P.TotalCredits as Total_credits
FROM Under
INNER JOIN Program P ON Under.PName = P.PName
WHERE Under.DName='Computer Science';
```

Result of Query IV:

Program	Total_credits
Computer Science	40

Query V:

```
SELECT DISTINCT S.STID as Student_ID, S.firstName as First_name
FROM Student S
INNER JOIN Belong B ON B.STID=S.STID
INNER JOIN Undergraduate U ON U.STID=S.STID
WHERE B.Advisor="";
```

Result of Query V:

Student_ID	First_name
1	Felix
4	Johnny

Query VI:

```
SELECT S.STID as Student_ID, S.FirstName as First_name, S.LastName as Last_name,
TA.AssignmentMarking as assignment_mandate
FROM Section
INNER JOIN Has ON Has.SeID=Section.SeID
INNER JOIN Course ON Has.CID=Course.CID
INNER JOIN AssignTo A ON Section.SeID = A.SeID
INNER JOIN TeachingAssistant TA ON A.TAID = TA.TAID
INNER JOIN Teach T ON Section.SeID = T.SeID
INNER JOIN EnrolledIn EI ON Section.SeID = EI.SeID
INNER JOIN Student S on EI.STID = S.STID
WHERE CName='COMP353' AND Semester='Summer' AND Year=2019;
```

Result of Query VI:

Student_ID	First_name	Last_name	Assignment_mandate
5	John	Smith	n
27	Sara	Slore	n
29	Lucienne	Bouchard	n
30	Esmeralda	Bobo	n

Query VII:

```
SELECT Instructor.Name
FROM Instructor
INNER JOIN Supervisor S ON Instructor.IID = S.IID
INNER JOIN Supervises S2 ON S.SupID = S2.SupID
GROUP BY Instructor.Name
HAVING COUNT(*) >= 20;
```

Result of Query VII:

Name
Lisa Cranterson

Query VIII:

```
SELECT Course.CName as Course_name , Section.SeID as 'Section', Class.Building,
Class.ClassNum as Room_location, Class.Capacity as Room_capacity,
ClassTimeslot.StartTime as Class_start_time, ClassTimeslot.EndTime as
Class_end_time
FROM Course
INNER JOIN Has ON Course.CID = Has.CID
INNER JOIN Section ON Has.SeID = Section.SeID
INNER JOIN EnrolledIn EI ON Section.SeID = EI.SeID
INNER JOIN Student S ON EI.STID = S.STID
INNER JOIN Belong B ON S.STID = B.STID
INNER JOIN Program P ON B.PName = P.PName
INNER JOIN Under U ON P.PName = U.PName
INNER JOIN Department D ON U.DName = D.DName
INNER JOIN ClassTimeslot ON Section.SeID = ClassTimeslot.SeID
INNER JOIN Class ON ClassTimeslot.CLID = Class.CLID
WHERE D.DName = 'Computer Science' AND Section.Semester='Summer' AND
Section.Year=2019;
```

Result of Query VIII:

Course_name	Section	Building	Room_location	Room_capacity	Class_start_time
Tal101	10	I Building	444	121	11:15:00

Query IX

```
SELECT D.DName as Department, COUNT(Course.CID) as Number_of_courses
FROM Course
INNER JOIN Has ON Course.CID = Has.CID
INNER JOIN Section ON Has.SeID = Section.SeID
INNER JOIN EnrolledIn EI ON Section.SeID = EI.SeID
INNER JOIN Student S ON EI.STID = S.STID
INNER JOIN Belong B ON S.STID = B.STID
INNER JOIN Program P ON B.PName = P.PName
INNER JOIN Under U ON P.PName = U.PName
INNER JOIN Department D ON U.DName = D.DName
GROUP BY D.DName;
```

Result of Query IX:

Department	Number_of_Courses
Anthropology	6
Computer Science	4
Engineering	6
Jewish Studies	14
Mathematics	3

Query X

```
SELECT Program.PName as Program, COUNT(S.STID) as Enrolled_Students
FROM Program
INNER JOIN Belong B ON Program.PName = B.PName
INNER JOIN Student S ON B.STID = S.STID
GROUP BY Program.PName;
```

Result of Query X:

Program	Enrolled_Students
Arts	6
Business	5
Computer Architecture	1
Computer Games	4
Computer Hardware	1
Computer Science	4
Economics	2
English	5
Law	2
Mathematics	3

4) Populating Tables

Requirements: Populate each table in the database with at least 10 representative and appropriate tuples.

The following **INSERT INTO** statements are in alphabetical order:

AssignTo table:

```
INSERT INTO AssignTo (SeID, TAID) VALUES (1, 1);
INSERT INTO AssignTo (SeID, TAID) VALUES (2, 2);
INSERT INTO AssignTo (SeID, TAID) VALUES (7, 5);
INSERT INTO AssignTo (SeID, TAID) VALUES (9, 6);
INSERT INTO AssignTo (SeID, TAID) VALUES (4, 7);
INSERT INTO AssignTo (SeID, TAID) VALUES (10, 4);
INSERT INTO AssignTo (SeID, TAID) VALUES (5, 5);
INSERT INTO AssignTo (SeID, TAID) VALUES (2, 3);
INSERT INTO AssignTo (SeID, TAID) VALUES (3, 8);
INSERT INTO AssignTo (SeID, TAID) VALUES (6, 9);
```

Belong table:

```
INSERT INTO Belong (STID, PName, Advisor) VALUES (1, 'Computer Science', '');
INSERT INTO Belong (STID, PName, Advisor) VALUES (2, 'Arts', 'Elma Aveiro');
INSERT INTO Belong (STID, PName, Advisor) VALUES (3, 'Business', 'Anthony Martial');
INSERT INTO Belong (STID, PName, Advisor) VALUES (4, 'Computer Architecture', '');
INSERT INTO Belong (STID, PName, Advisor) VALUES (5, 'Computer Games', 'Paul Scholes');
INSERT INTO Belong (STID, PName, Advisor) VALUES (6, 'Computer Hardware', '');
INSERT INTO Belong (STID, PName, Advisor) VALUES (7, 'Economics', 'Mason Greenwood');
INSERT INTO Belong (STID, PName, Advisor) VALUES (8, 'English', '');
INSERT INTO Belong (STID, PName, Advisor) VALUES (9, 'Law', 'James Garner');
INSERT INTO Belong (STID, PName, Advisor) VALUES (10, 'Mathematics', 'Gary Neville');
INSERT INTO Belong (STID, PName, Advisor) VALUES (1, 'Arts', 'Katia Aveiro');
INSERT INTO Belong (STID, PName, Advisor) VALUES (11, 'Computer Science', '');
INSERT INTO Belong (STID, PName, Advisor) VALUES (12, 'Computer Science', '');
INSERT INTO Belong (STID, PName, Advisor) VALUES (17, 'English', '');
INSERT INTO Belong (STID, PName, Advisor) VALUES (18, 'Business', '');
INSERT INTO Belong (STID, PName, Advisor) VALUES (19, 'Mathematics', 'Bob Jiye');
INSERT INTO Belong (STID, PName, Advisor) VALUES (20, 'Arts', 'Marc Zuter');
INSERT INTO Belong (STID, PName, Advisor) VALUES (21, 'Computer Science', '');
INSERT INTO Belong (STID, PName, Advisor) VALUES (22, 'Computer Games', '');
INSERT INTO Belong (STID, PName, Advisor) VALUES (13, 'English', 'Bobby Jo');
```

```

INSERT INTO Belong (STID, PName, Advisor) VALUES (14, 'Arts', 'Jack');
INSERT INTO Belong (STID, PName, Advisor) VALUES (15, 'Business', 'Cristina');
INSERT INTO Belong (STID, PName, Advisor) VALUES (16, 'English', 'Louise');
INSERT INTO Belong (STID, PName, Advisor) VALUES (23, 'Business', null);
INSERT INTO Belong (STID, PName, Advisor) VALUES (24, 'Mathematics', null);
INSERT INTO Belong (STID, PName, Advisor) VALUES (25, 'Computer Games', null);
INSERT INTO Belong (STID, PName, Advisor) VALUES (26, 'Economics', 'Lucy');
INSERT INTO Belong (STID, PName, Advisor) VALUES (27, 'Law', null);
INSERT INTO Belong (STID, PName, Advisor) VALUES (28, 'Business', null);
INSERT INTO Belong (STID, PName, Advisor) VALUES (29, 'Computer Games',
'Bertrand');
INSERT INTO Belong (STID, PName, Advisor) VALUES (30, 'Arts', null);
INSERT INTO Belong (STID, PName, Advisor)
VALUES (31, 'Arts', 'Marc Luver');
INSERT INTO Belong (STID, PName, Advisor) VALUES (32, 'English', 'Bobby Jo');

```

Class table:

```

INSERT INTO Class (CLID, ClassNum, Building, Capacity) VALUES (1, 737, 'A Building',
150);
INSERT INTO Class (CLID, ClassNum, Building, Capacity) VALUES (2, 434, 'B Building',
125);
INSERT INTO Class (CLID, ClassNum, Building, Capacity) VALUES (3, 736, 'C Building',
100);
INSERT INTO Class (CLID, ClassNum, Building, Capacity) VALUES (4, 907, 'D Building',
145);
INSERT INTO Class (CLID, ClassNum, Building, Capacity) VALUES (5, 675, 'E Building',
190);
INSERT INTO Class (CLID, ClassNum, Building, Capacity) VALUES (6, 234, 'F Building',
250);
INSERT INTO Class (CLID, ClassNum, Building, Capacity) VALUES (7, 943, 'G Building',
203);
INSERT INTO Class (CLID, ClassNum, Building, Capacity) VALUES (8, 390, 'H Building',
109);
INSERT INTO Class (CLID, ClassNum, Building, Capacity) VALUES (9, 444, 'I Building',
121);
INSERT INTO Class (CLID, ClassNum, Building, Capacity) VALUES (10, 656, 'J Building',
153);

```

ClassTimeslot table:

```

INSERT INTO ClassTimeslot (StartTime, DayWeek, CLID, SeID, EndTime) VALUES
('08:45:00', 'Monday', 1, 1, '09:45:00');
INSERT INTO ClassTimeslot (StartTime, DayWeek, CLID, SeID, EndTime) VALUES
('09:30:00', 'Wednesday', 2, 2, '10:45:00');
INSERT INTO ClassTimeslot (StartTime, DayWeek, CLID, SeID, EndTime) VALUES
('10:00:00', 'Tuesday', 3, 3, '11:45:00');

```

```

INSERT INTO ClassTimeslot (StartTime, DayWeek, CLID, SeID, EndTime) VALUES
('10:30:00', 'Monday', 6, 7, '12:45:00');
INSERT INTO ClassTimeslot (StartTime, DayWeek, CLID, SeID, EndTime) VALUES
('10:45:00', 'Wednesday', 8, 4, '11:45:00');
INSERT INTO ClassTimeslot (StartTime, DayWeek, CLID, SeID, EndTime) VALUES
('11:00:00', 'Monday', 7, 9, '12:45:00');
INSERT INTO ClassTimeslot (StartTime, DayWeek, CLID, SeID, EndTime) VALUES
('11:15:00', 'Thursday', 9, 10, '13:45:00');
INSERT INTO ClassTimeslot (StartTime, DayWeek, CLID, SeID, EndTime) VALUES
('11:45:00', 'Thursday', 4, 6, '12:45:00');
INSERT INTO ClassTimeslot (StartTime, DayWeek, CLID, SeID, EndTime) VALUES
('12:30:00', 'Friday', 10, 5, '14:45:00');
INSERT INTO ClassTimeslot (StartTime, DayWeek, CLID, SeID, EndTime) VALUES
('12:45:00', 'Friday', 5, 8, '13:45:00');

```

Course table:

```

INSERT INTO Course (CID, CName) VALUES (1, 'COMP352');
INSERT INTO Course (CID, CName) VALUES (2, 'Mech300');
INSERT INTO Course (CID, CName) VALUES (3, 'Fr465');
INSERT INTO Course (CID, CName) VALUES (4, 'Sex230');
INSERT INTO Course (CID, CName) VALUES (5, 'Bio202');
INSERT INTO Course (CID, CName) VALUES (6, 'COMP353');
INSERT INTO Course (CID, CName) VALUES (7, 'His375');
INSERT INTO Course (CID, CName) VALUES (8, 'Engr301');
INSERT INTO Course (CID, CName) VALUES (9, 'COMP352');
INSERT INTO Course (CID, CName) VALUES (10, 'Tal101');

```

Department table:

```

INSERT INTO Department (DName, Building) VALUES ('Anthropology', 'I Building');
INSERT INTO Department (DName, Building) VALUES ('Biology', 'C Building');
INSERT INTO Department (DName, Building) VALUES ('Computer Science', 'A
Building');
INSERT INTO Department (DName, Building) VALUES ('Engineering', 'E Building');
INSERT INTO Department (DName, Building) VALUES ('French', 'F Building');
INSERT INTO Department (DName, Building) VALUES ('Gender Studies', 'G Building');
INSERT INTO Department (DName, Building) VALUES ('History', 'H Building');
INSERT INTO Department (DName, Building) VALUES ('Jewish Studies', 'J Building');
INSERT INTO Department (DName, Building) VALUES ('Mathematics', 'B Building');
INSERT INTO Department (DName, Building) VALUES ('Physics', 'D Building');

```

EnrolledIn table:

```
INSERT INTO EnrolledIn(STID, SeID, Grade)VALUES(10, 1, 'F');
INSERT INTO EnrolledIn(STID, SeID, Grade)VALUES(9, 2, 'D-');
INSERT INTO EnrolledIn(STID, SeID, Grade)VALUES(8, 3, 'D');
INSERT INTO EnrolledIn(STID, SeID, Grade)VALUES(7, 4, 'D+');
INSERT INTO EnrolledIn(STID, SeID, Grade)VALUES(6, 5, 'A+');
INSERT INTO EnrolledIn(STID, SeID, Grade)VALUES(5, 6, 'C');
INSERT INTO EnrolledIn(STID, SeID, Grade)VALUES(4, 7, 'C+');
INSERT INTO EnrolledIn(STID, SeID, Grade)VALUES(3, 8, 'A');
INSERT INTO EnrolledIn(STID, SeID, Grade)VALUES(2, 9, 'A');
INSERT INTO EnrolledIn(STID, SeID, Grade)VALUES(1, 10, 'A+');
INSERT INTO EnrolledIn(STID, SeID, Grade)VALUES(11, 1, 'A+');
INSERT INTO EnrolledIn(STID, SeID, Grade)VALUES(12, 1, 'A+');
INSERT INTO EnrolledIn(STID, SeID, Grade)VALUES(17, 2, 'A');
INSERT INTO EnrolledIn(STID, SeID, Grade)VALUES(18, 3, 'B');
INSERT INTO EnrolledIn(STID, SeID, Grade)VALUES(19, 1, 'A+');
INSERT INTO EnrolledIn(STID, SeID, Grade)VALUES(20, 5, 'C-');
INSERT INTO EnrolledIn(STID, SeID, Grade)VALUES(21, 7, 'B-');
INSERT INTO EnrolledIn(STID, SeID, Grade)VALUES(22, 8, 'E');
INSERT INTO EnrolledIn(STID, SeID, Grade)VALUES(13, 1, 'E');
INSERT INTO EnrolledIn(STID, SeID, Grade)VALUES(14, 3, 'D');
INSERT INTO EnrolledIn(STID, SeID, Grade)VALUES(15, 2, 'B');
INSERT INTO EnrolledIn(STID, SeID, Grade)VALUES(16, 5, 'B+');
INSERT INTO EnrolledIn(STID, SeID, Grade)VALUES(23, 4, 'F');
INSERT INTO EnrolledIn(STID, SeID, Grade)VALUES(24, 5, 'F');
INSERT INTO EnrolledIn(STID, SeID, Grade)VALUES(25, 1, 'A+');
INSERT INTO EnrolledIn(STID, SeID, Grade)VALUES(26, 7, 'C');
INSERT INTO EnrolledIn(STID, SeID, Grade)VALUES(27, 6, 'C-');
INSERT INTO EnrolledIn(STID, SeID, Grade)VALUES(28, 1, 'F');
INSERT INTO EnrolledIn(STID, SeID, Grade)VALUES(29, 6, 'B+');
INSERT INTO EnrolledIn(STID, SeID, Grade)VALUES(30, 6, 'B+');
INSERT INTO EnrolledIn(STID, SeID, Grade)VALUES(31, 4, 'A-');
INSERT INTO EnrolledIn(STID, SeID, Grade)VALUES(32, 10, 'B');
```

Fund table:

```
INSERT INTO Fund (STID, TAID, RID) VALUES (6, 3, null);
INSERT INTO Fund (STID, TAID, RID) VALUES (7, 2, null);
INSERT INTO Fund (STID, TAID, RID) VALUES (8, 1, 1);
INSERT INTO Fund (STID, TAID, RID) VALUES (9, 5, 2);
INSERT INTO Fund (STID, TAID, RID) VALUES (10, 2, 3);
INSERT INTO Fund (STID, TAID, RID) VALUES (11, 3, null);
INSERT INTO Fund (STID, TAID, RID) VALUES (12, 6, null);
INSERT INTO Fund (STID, TAID, RID) VALUES (17, 3, 2);
INSERT INTO Fund (STID, TAID, RID) VALUES (18, 7, 1);
INSERT INTO Fund (STID, TAID, RID) VALUES (19, 8, null);
```

```
INSERT INTO Fund (STID, TAID, RID) VALUES (20, 2, 3);
INSERT INTO Fund (STID, TAID, RID) VALUES (21, 3, 1);
INSERT INTO Fund (STID, TAID, RID) VALUES (22, 9, null);
INSERT INTO Fund (STID, TAID, RID) VALUES (13, 4, null);
INSERT INTO Fund (STID, TAID, RID) VALUES (14, 4, null);
INSERT INTO Fund (STID, TAID, RID) VALUES (15, 6, null);
INSERT INTO Fund (STID, TAID, RID) VALUES (16, 4, null);
INSERT INTO Fund (STID, TAID, RID) VALUES (23, 2, null);
INSERT INTO Fund (STID, TAID, RID) VALUES (24, 8, null);
INSERT INTO Fund (STID, TAID, RID) VALUES (25, 5, null);
INSERT INTO Fund (STID, TAID, RID) VALUES (26, 5, null);
INSERT INTO Fund (STID, TAID, RID) VALUES (27, 9, null);
```

Graduate table:

```
INSERT INTO Graduate (STID) VALUES (6);
INSERT INTO Graduate (STID) VALUES (7);
INSERT INTO Graduate (STID) VALUES (8);
INSERT INTO Graduate (STID) VALUES (9);
INSERT INTO Graduate (STID) VALUES (10);
INSERT INTO Graduate (STID) VALUES (11);
INSERT INTO Graduate (STID) VALUES (12);
INSERT INTO Graduate (STID) VALUES (13);
INSERT INTO Graduate (STID) VALUES (14);
INSERT INTO Graduate (STID) VALUES (15);
INSERT INTO Graduate (STID) VALUES (16);
INSERT INTO Graduate (STID) VALUES (17);
INSERT INTO Graduate (STID) VALUES (18);
INSERT INTO Graduate (STID) VALUES (19);
INSERT INTO Graduate (STID) VALUES (20);
INSERT INTO Graduate (STID) VALUES (21);
INSERT INTO Graduate (STID) VALUES (22);
INSERT INTO Graduate (STID) VALUES (23);
INSERT INTO Graduate (STID) VALUES (24);
INSERT INTO Graduate (STID) VALUES (25);
INSERT INTO Graduate (STID) VALUES (26);
INSERT INTO Graduate (STID) VALUES (27);
```

Has table:

```
INSERT INTO Has (CID, SeID) VALUES (1, 1);
INSERT INTO Has (CID, SeID) VALUES (2, 2);
INSERT INTO Has (CID, SeID) VALUES (3, 3);
INSERT INTO Has (CID, SeID) VALUES (4, 4);
INSERT INTO Has (CID, SeID) VALUES (5, 5);
INSERT INTO Has (CID, SeID) VALUES (6, 6);
INSERT INTO Has (CID, SeID) VALUES (7, 7);
```



```
INSERT INTO Has (CID, SelD) VALUES (8, 8);
INSERT INTO Has (CID, SelD) VALUES (1, 9);
INSERT INTO Has (CID, SelD) VALUES (10, 10);
INSERT INTO Has (CID, SelD) VALUES (1, 11);
```

Instructor table:

```
INSERT INTO Instructor (IID, Name, Supervised) VALUES (1, 'Lisa Cranterson', 35);
INSERT INTO Instructor (IID, Name, Supervised) VALUES (2, 'Ernest Steig', 7);
INSERT INTO Instructor (IID, Name, Supervised) VALUES (3, 'Melissa Roberts', 10);
INSERT INTO Instructor (IID, Name, Supervised) VALUES (4, 'Jake Ralph', 8);
INSERT INTO Instructor (IID, Name, Supervised) VALUES (5, 'Claire Devons', 3);
INSERT INTO Instructor (IID, Name, Supervised) VALUES (6, 'Jeremy Kudo', 30);
INSERT INTO Instructor (IID, Name, Supervised) VALUES (7, 'Linda Torrents', 20);
INSERT INTO Instructor (IID, Name, Supervised) VALUES (8, 'Jessica Stevenson', 54);
INSERT INTO Instructor (IID, Name, Supervised) VALUES (9, 'Isaac Harrison', 2);
INSERT INTO Instructor (IID, Name, Supervised) VALUES (10, 'Nicholas Larsen', 15);
INSERT INTO Instructor (IID, Name, Supervised) VALUES (11, 'Cristiano Ronaldo', 4);
```

Program table:

```
INSERT INTO Program (PName, TotalCredits) VALUES ('Arts', 40);
INSERT INTO Program (PName, TotalCredits) VALUES ('Business', 22);
INSERT INTO Program (PName, TotalCredits) VALUES ('Computer Architecture', 15);
INSERT INTO Program (PName, TotalCredits) VALUES ('Computer Games', 30);
INSERT INTO Program (PName, TotalCredits) VALUES ('Computer Hardware', 33);
INSERT INTO Program (PName, TotalCredits) VALUES ('Computer Science', 40);
INSERT INTO Program (PName, TotalCredits) VALUES ('Economics', 6);
INSERT INTO Program (PName, TotalCredits) VALUES ('English', 12);
INSERT INTO Program (PName, TotalCredits) VALUES ('Law', 14);
INSERT INTO Program (PName, TotalCredits) VALUES ('Mathematics', 23);
```

ResearchFunding table:

```
INSERT INTO ResearchFunding (RID, Amount) VALUES (1, 1000);
INSERT INTO ResearchFunding (RID, Amount) VALUES (2, 10000);
INSERT INTO ResearchFunding (RID, Amount) VALUES (3, 8000);
INSERT INTO ResearchFunding (RID, Amount) VALUES (4, 200);
INSERT INTO ResearchFunding (RID, Amount) VALUES (5, 100);
INSERT INTO ResearchFunding (RID, Amount) VALUES (6, 6000);
INSERT INTO ResearchFunding (RID, Amount) VALUES (7, 3000);
INSERT INTO ResearchFunding (RID, Amount) VALUES (8, 1500);
INSERT INTO ResearchFunding (RID, Amount) VALUES (9, 100);
INSERT INTO ResearchFunding (RID, Amount) VALUES (10, 20000);
```

Section table:

```
INSERT INTO Section (SeID, Semester, Year) VALUES (1, 'Fall', 2018);
INSERT INTO Section (SeID, Semester, Year) VALUES (2, 'Winter', 2020);
INSERT INTO Section (SeID, Semester, Year) VALUES (3, 'Summer', 2019);
INSERT INTO Section (SeID, Semester, Year) VALUES (4, 'Fall', 2019);
INSERT INTO Section (SeID, Semester, Year) VALUES (5, 'Winter', 2020);
INSERT INTO Section (SeID, Semester, Year) VALUES (6, 'Summer', 2019);
INSERT INTO Section (SeID, Semester, Year) VALUES (7, 'Fall', 2019);
INSERT INTO Section (SeID, Semester, Year) VALUES (8, 'Winter', 2020);
INSERT INTO Section (SeID, Semester, Year) VALUES (9, 'Summer', 2019);
INSERT INTO Section (SeID, Semester, Year) VALUES (10, 'Summer', 2019);
INSERT INTO Section (SeID, Semester, Year) VALUES (11, 'Summer', 2017);
```

Student table:

```
INSERT INTO Student(STID, Credit, FirstName, LastName, GPA)VALUES(1, 90, 'Felix',
'Harris', 3.1);
INSERT INTO Student(STID, Credit, FirstName, LastName, GPA)VALUES(2, 90, 'James',
'Watson', 2);
INSERT INTO Student(STID, Credit, FirstName, LastName, GPA)VALUES(3, 90,
'Marcus', 'Morris', 2.7);
INSERT INTO Student(STID, Credit, FirstName, LastName, GPA)VALUES(4, 90,
'Johnny', 'Howard', 3.4);
INSERT INTO Student(STID, Credit, FirstName, LastName, GPA)VALUES(5, 90, 'John',
'Smith', 3.1);
INSERT INTO Student(STID, Credit, FirstName, LastName, GPA)VALUES(6, 44, 'Phil',
'Newton', 2.1);
INSERT INTO Student(STID, Credit, FirstName, LastName, GPA)VALUES(7, 44,
'Andrew', 'Morrison', 2);
INSERT INTO Student(STID, Credit, FirstName, LastName, GPA)VALUES(8, 44,
'Mackenzie', 'Johnson', 3.2);
INSERT INTO Student(STID, Credit, FirstName, LastName, GPA)VALUES(9, 44, 'Max',
'Phillips', 3);
INSERT INTO Student(STID, Credit, FirstName, LastName, GPA)VALUES(10, 44, 'Linus',
'Torwards', 4);
INSERT INTO Student(STID, Credit, FirstName, LastName, GPA)VALUES(11, 44,
'Khaled', 'Jababo', 4);
INSERT INTO Student(STID, Credit, FirstName, LastName, GPA)VALUES(12, 44,
'Aiman', 'Hanna', 4);
INSERT INTO Student(STID, Credit, FirstName, LastName, GPA)VALUES(13, 44,
'Messi', 'Lionel', 1);
INSERT INTO Student(STID, Credit, FirstName, LastName, GPA)VALUES(14, 44, 'Luis',
'Loni', 3);
INSERT INTO Student(STID, Credit, FirstName, LastName, GPA)VALUES(15, 44,
'Goun', 'Lio', 2);
```

```

INSERT INTO Student(STID, Credit, FirstName, LastName, GPA)VALUES(16, 44,
'Harves', 'Ri', 2);
INSERT INTO Student(STID, Credit, FirstName, LastName, GPA)VALUES(17, 44,
'Jonathan', 'Izzmifirtre', 3.9);
INSERT INTO Student(STID, Credit, FirstName, LastName, GPA)VALUES(18, 44,
'Nicolas', 'Tabourette', 3.8);
INSERT INTO Student(STID, Credit, FirstName, LastName, GPA)VALUES(19, 44,
'Nathan', 'Mackinnon', 3.8);
INSERT INTO Student(STID, Credit, FirstName, LastName, GPA)VALUES(20, 44,
'Sydney', 'Crosby', 4);
INSERT INTO Student(STID, Credit, FirstName, LastName, GPA)VALUES(21, 44,
'David', 'Izzmifartre', 3.9);
INSERT INTO Student(STID, Credit, FirstName, LastName, GPA)VALUES(22, 44,
'Poche', 'Jigyuan', 4);
INSERT INTO Student(STID, Credit, FirstName, LastName, GPA)VALUES(23, 44, 'Larry',
'Simpson', 2.7);
INSERT INTO Student(STID, Credit, FirstName, LastName, GPA)VALUES(24, 44, 'Foki',
'Ruki', 2);
INSERT INTO Student(STID, Credit, FirstName, LastName, GPA)VALUES(25, 44,
'Smity', 'Sam', 4);
INSERT INTO Student(STID, Credit, FirstName, LastName, GPA)VALUES(26, 44, 'Rawl',
'Sol', 3);
INSERT INTO Student(STID, Credit, FirstName, LastName, GPA)VALUES(27, 44, 'Sara',
'Flore', 4);
INSERT INTO Student(STID, Credit, FirstName, LastName, GPA)VALUES(28, 90, 'Joey',
'Looker', 3.2);
INSERT INTO Student(STID, Credit, FirstName, LastName, GPA)VALUES(29, 90,
'Lucienne', 'Bouchard', 1.5);
INSERT INTO Student(STID, Credit, FirstName, LastName, GPA)VALUES(30, 90,
'Esmeralda', 'Bobo', 2.7);
INSERT INTO Student(STID, Credit, FirstName, LastName, GPA)VALUES(31, 90,
'Laurent', 'Voyer', 2.8);
INSERT INTO Student(STID, Credit, FirstName, LastName, GPA)VALUES(32, 90,
'Eddey', 'Boucher', 3);

```

Supervises table:

```

INSERT INTO Supervises (SupID, STID) VALUES (1, 6);
INSERT INTO Supervises (SupID, STID) VALUES (1, 7);
INSERT INTO Supervises (SupID, STID) VALUES (1, 8);
INSERT INTO Supervises (SupID, STID) VALUES (1, 9);
INSERT INTO Supervises (SupID, STID) VALUES (1, 10);
INSERT INTO Supervises (SupID, STID) VALUES (1, 11);
INSERT INTO Supervises (SupID, STID) VALUES (1, 12);
INSERT INTO Supervises (SupID, STID) VALUES (1, 17);
INSERT INTO Supervises (SupID, STID) VALUES (1, 18);
INSERT INTO Supervises (SupID, STID) VALUES (1, 19);

```

```
INSERT INTO Supervises (SupID, STID) VALUES (1, 20);
INSERT INTO Supervises (SupID, STID) VALUES (4, 21);
INSERT INTO Supervises (SupID, STID) VALUES (1, 22);
INSERT INTO Supervises (SupID, STID) VALUES (1, 13);
INSERT INTO Supervises (SupID, STID) VALUES (1, 14);
INSERT INTO Supervises (SupID, STID) VALUES (1, 15);
INSERT INTO Supervises (SupID, STID) VALUES (1, 16);
INSERT INTO Supervises (SupID, STID) VALUES (1, 23);
INSERT INTO Supervises (SupID, STID) VALUES (1, 24);
INSERT INTO Supervises (SupID, STID) VALUES (1, 25);
INSERT INTO Supervises (SupID, STID) VALUES (1, 26);
INSERT INTO Supervises (SupID, STID) VALUES (1, 27);
```

Supervisor table:

```
INSERT INTO Supervisor (SupID, IID) VALUES (1, 1);
INSERT INTO Supervisor (SupID, IID) VALUES (4, 2);
INSERT INTO Supervisor (SupID, IID) VALUES (2, 3);
INSERT INTO Supervisor (SupID, IID) VALUES (3, 4);
INSERT INTO Supervisor (SupID, IID) VALUES (5, 5);
INSERT INTO Supervisor (SupID, IID) VALUES (6, 6);
INSERT INTO Supervisor (SupID, IID) VALUES (7, 7);
INSERT INTO Supervisor (SupID, IID) VALUES (8, 8);
INSERT INTO Supervisor (SupID, IID) VALUES (9, 9);
INSERT INTO Supervisor (SupID, IID) VALUES (10, 10);
```

Teach table:

```
INSERT INTO Teach (SeID, IID) VALUES (1, 10);
INSERT INTO Teach (SeID, IID) VALUES (2, 9);
INSERT INTO Teach (SeID, IID) VALUES (3, 8);
INSERT INTO Teach (SeID, IID) VALUES (4, 7);
INSERT INTO Teach (SeID, IID) VALUES (5, 6);
INSERT INTO Teach (SeID, IID) VALUES (6, 5);
INSERT INTO Teach (SeID, IID) VALUES (7, 4);
INSERT INTO Teach (SeID, IID) VALUES (8, 3);
INSERT INTO Teach (SeID, IID) VALUES (9, 2);
INSERT INTO Teach (SeID, IID) VALUES (10, 1);
INSERT INTO Teach (SeID, IID) VALUES (11, 2);
INSERT INTO Teach (SeID, IID) VALUES (1, 2);
```

TeachingAssistant table:

```
INSERT INTO TeachingAssistant (TAID, TotalHours, AssignmentMarking,
LabInstructor, NumCourses, TutorialSession) VALUES (1, 3, 'y', 'n', 1, 'y');
INSERT INTO TeachingAssistant (TAID, TotalHours, AssignmentMarking,
LabInstructor, NumCourses, TutorialSession) VALUES (2, 6, 'y', 'y', 2, 'n');
```

```

INSERT INTO TeachingAssistant (TAID, TotalHours, AssignmentMarking,
LabInstructor, NumCourses, TutorialSession) VALUES (3, 7, 'n', 'n', 3, 'n');
INSERT INTO TeachingAssistant (TAID, TotalHours, AssignmentMarking,
LabInstructor, NumCourses, TutorialSession) VALUES (4, 10, 'y', 'n', 2, 'n');
INSERT INTO TeachingAssistant (TAID, TotalHours, AssignmentMarking,
LabInstructor, NumCourses, TutorialSession) VALUES (5, 3, 'y', 'y', 3, '');
INSERT INTO TeachingAssistant (TAID, TotalHours, AssignmentMarking,
LabInstructor, NumCourses, TutorialSession) VALUES (6, 5, 'y', 'y', 2, 'y');
INSERT INTO TeachingAssistant (TAID, TotalHours, AssignmentMarking,
LabInstructor, NumCourses, TutorialSession) VALUES (7, 7, 'n', 'n', 2, 'y');
INSERT INTO TeachingAssistant (TAID, TotalHours, AssignmentMarking,
LabInstructor, NumCourses, TutorialSession) VALUES (8, 3, 'y', 'n', 1, 'n');
INSERT INTO TeachingAssistant (TAID, TotalHours, AssignmentMarking,
LabInstructor, NumCourses, TutorialSession) VALUES (9, 8, 'n', 'n', 1, 'n');
INSERT INTO TeachingAssistant (TAID, TotalHours, AssignmentMarking,
LabInstructor, NumCourses, TutorialSession) VALUES (10, 4, 'y', 'y', 3, 'n');

```

Under table:

```

INSERT INTO Under (DName, PName) VALUES ('Anthropology', 'Arts');
INSERT INTO Under (DName, PName) VALUES ('Engineering', 'Computer
Architecture');
INSERT INTO Under (DName, PName) VALUES ('Engineering', 'Computer Games');
INSERT INTO Under (DName, PName) VALUES ('Engineering', 'Computer Hardware');
INSERT INTO Under (DName, PName) VALUES ('Computer Science', 'Computer
Science');
INSERT INTO Under (DName, PName) VALUES ('Jewish Studies', 'Business');
INSERT INTO Under (DName, PName) VALUES ('Jewish Studies', 'Economics');
INSERT INTO Under (DName, PName) VALUES ('Jewish Studies', 'English');
INSERT INTO Under (DName, PName) VALUES ('Jewish Studies', 'Law');
INSERT INTO Under (DName, PName) VALUES ('Mathematics', 'Mathematics');

```

Undergraduate table:

```

INSERT INTO Undergraduate(STID)VALUES(1);
INSERT INTO Undergraduate(STID)VALUES(2);
INSERT INTO Undergraduate(STID)VALUES(3);
INSERT INTO Undergraduate(STID)VALUES(4);
INSERT INTO Undergraduate(STID)VALUES(5);
INSERT INTO Undergraduate(STID)VALUES(28);
INSERT INTO Undergraduate(STID)VALUES(29);
INSERT INTO Undergraduate(STID)VALUES(30);
INSERT INTO Undergraduate(STID)VALUES(31);
INSERT INTO Undergraduate(STID)VALUES(32);

```

Within table:

```
INSERT INTO Within (CID, DName) VALUES (1, 'Computer Science');
INSERT INTO Within (CID, DName) VALUES (2, 'Physics');
INSERT INTO Within (CID, DName) VALUES (3, 'French');
INSERT INTO Within (CID, DName) VALUES (4, 'Gender Studies');
INSERT INTO Within (CID, DName) VALUES (5, 'Biology');
INSERT INTO Within (CID, DName) VALUES (6, 'Mathematics');
INSERT INTO Within (CID, DName) VALUES (7, 'History');
INSERT INTO Within (CID, DName) VALUES (8, 'Engineering');
INSERT INTO Within (CID, DName) VALUES (9, 'Anthropology');
INSERT INTO Within (CID, DName) VALUES (10, 'Jewish Studies');
```

Work table:

```
INSERT INTO Work (DName, IID) VALUES ('Physics', 1);
INSERT INTO Work (DName, IID) VALUES ('Mathematics', 2);
INSERT INTO Work (DName, IID) VALUES ('Biology', 3);
INSERT INTO Work (DName, IID) VALUES ('Engineering', 4);
INSERT INTO Work (DName, IID) VALUES ('Gender Studies', 5);
INSERT INTO Work (DName, IID) VALUES ('Anthropology', 6);
INSERT INTO Work (DName, IID) VALUES ('Jewish Studies', 8);
INSERT INTO Work (DName, IID) VALUES ('History', 9);
INSERT INTO Work (DName, IID) VALUES ('Computer Science', 10);
INSERT INTO Work (DName, IID) VALUES ('French', 7);
INSERT INTO Work (DName, IID) VALUES ('Computer Science', 11);
```

5) Query **SELECT COUNT(*) FROM R**

Requirement: For each relation **R** created in your database, report the result of the following SQL statement:

The following **SELECT COUNT(*) FROM R**

SELECT COUNT(*) FROM AssignTo:

COUNT(*)
10

SELECT COUNT(*) FROM Belong:

COUNT(*)
33

SELECT COUNT(*) FROM Class:

COUNT(*)
10

SELECT COUNT(*) FROM ClassTimeslot:

COUNT(*)
10

SELECT COUNT(*) FROM Course:

COUNT(*)
10

SELECT COUNT(*) FROM Department:

COUNT(*)
10

SELECT COUNT(*) FROM EnrolledIn:

COUNT(*)
32

SELECT COUNT(*) FROM Fund:

COUNT(*)
22

SELECT COUNT(*) FROM Graduate:

COUNT(*)
22

SELECT COUNT(*) FROM Has:

COUNT(*)
11

SELECT COUNT(*) FROM Instructor:

COUNT(*)
11

SELECT COUNT(*) FROM Program:

COUNT(*)
10

SELECT COUNT(*) FROM ResearchFunding:

COUNT(*)
10

SELECT COUNT(*) FROM Section:

COUNT(*)
11

SELECT COUNT(*) FROM Student:

COUNT(*)
32

SELECT COUNT(*) FROM Supervises:

COUNT(*)
22

SELECT COUNT(*) FROM Supervisor:

COUNT(*)
10

SELECT COUNT(*) FROM Teach:

COUNT(*)
12

SELECT COUNT(*) FROM TeachingAssistant:

COUNT(*)
10

SELECT COUNT(*) FROM Under:

COUNT(*)
10

SELECT COUNT(*) FROM Undergraduate:

COUNT(*)
10

SELECT COUNT(*) FROM Within:

COUNT(*)
10

SELECT COUNT(*) FROM Work:

COUNT(*)
11