CS362 - Structured Query Language for Data Management

Andy Logan

March 19, 2018

Individual Projects 1 – 5

Dr. Angela Griffin

# Table of Contents

[IP 1 3](#_Toc509251881)

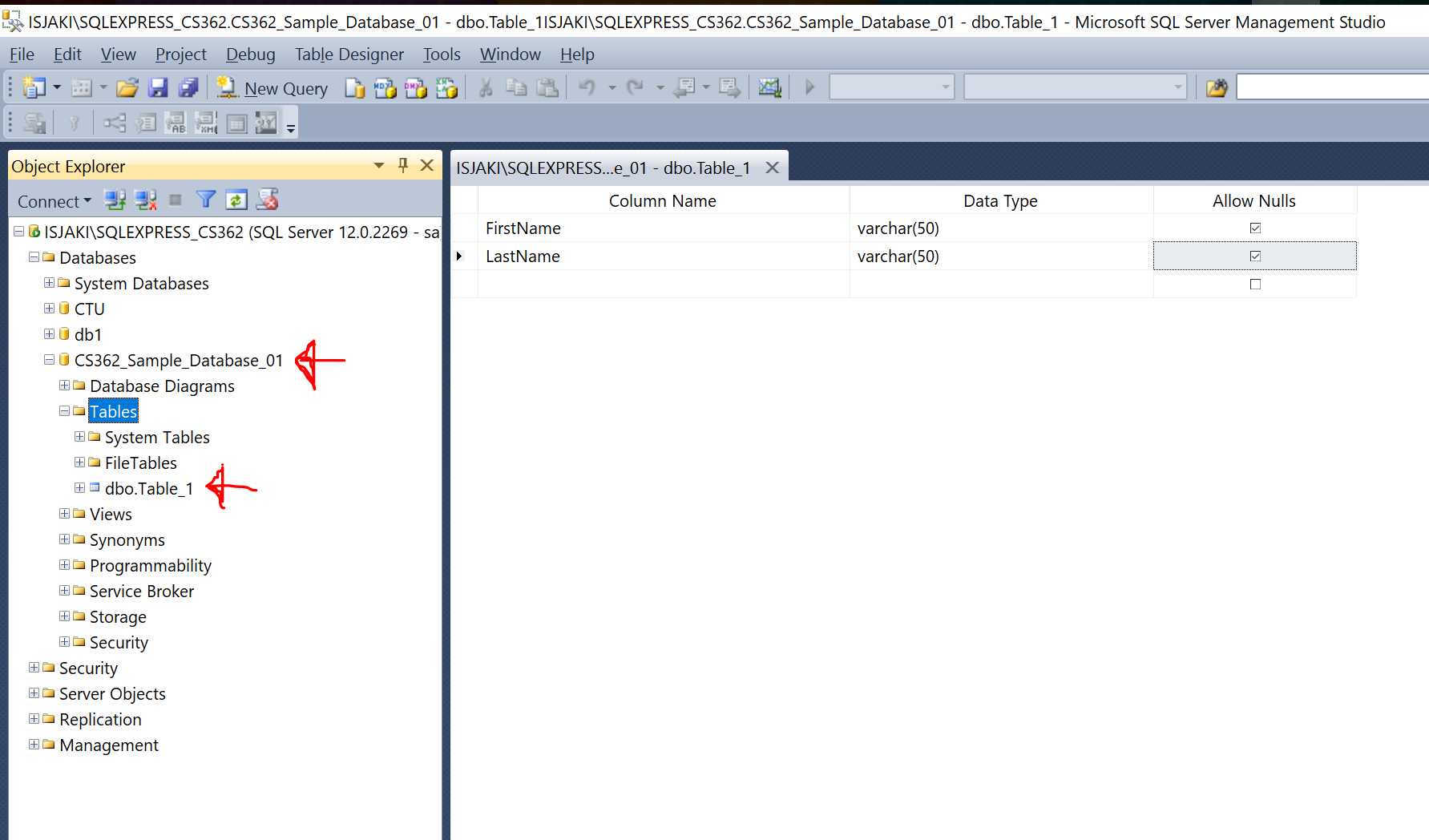
[IP 2 4](#_Toc509251882)

[IP 3 6](#_Toc509251883)

[IP 4 9](#_Toc509251884)

[IP 5 12](#_Toc509251885)

# IP 1



As I had Microsoft SQL Server 2014 installed on my computer from the last time I attempted to take this course, I had no issues with the software. After joining Live Chat 1, the instructions were clear and the creation of the new database and table were quite easy.

# IP 2

create database University;

create table students(

studentid int IDENTITY(1,1) primary key not null,

firstname varchar(50) not null,

lastname varchar(50) not null,

birthdate date not null,

gender char(1) not null,

startdate date not null,

GPA decimal (3,2) not null,

isactive text not null,

bio text not null,

advisorid int not null);

create table advisors(

advisorid int IDENTITY(1,1) primary key not null,

firstname varchar(50) not null,

lastname varchar(50) not null,

email varchar(100) not null);

create table classes(

classid int IDENTITY(1,1) primary key not null,

classcode varchar(50) not null,

classname varchar(100) not null,

classdescription varchar(1000) not null);

create table students\_classes(

studentclassid int IDENTITY(1,1) primary key not null,

studentid int not null,

classid int not null,

startdate date not null,

assignment1 decimal(5,2),

assignment2 decimal(5,2),

assignment3 decimal(5,2),

assignment4 decimal(5,2),

classgpa decimal(3,2) not null);

alter table students with check add constraint fk\_students\_advisors

foreign key(advisorid) references advisors (advisorid);

alter table students check constraint fk\_students\_advisors;

alter table students\_classes with check add constraint fk\_students\_classes\_studentid

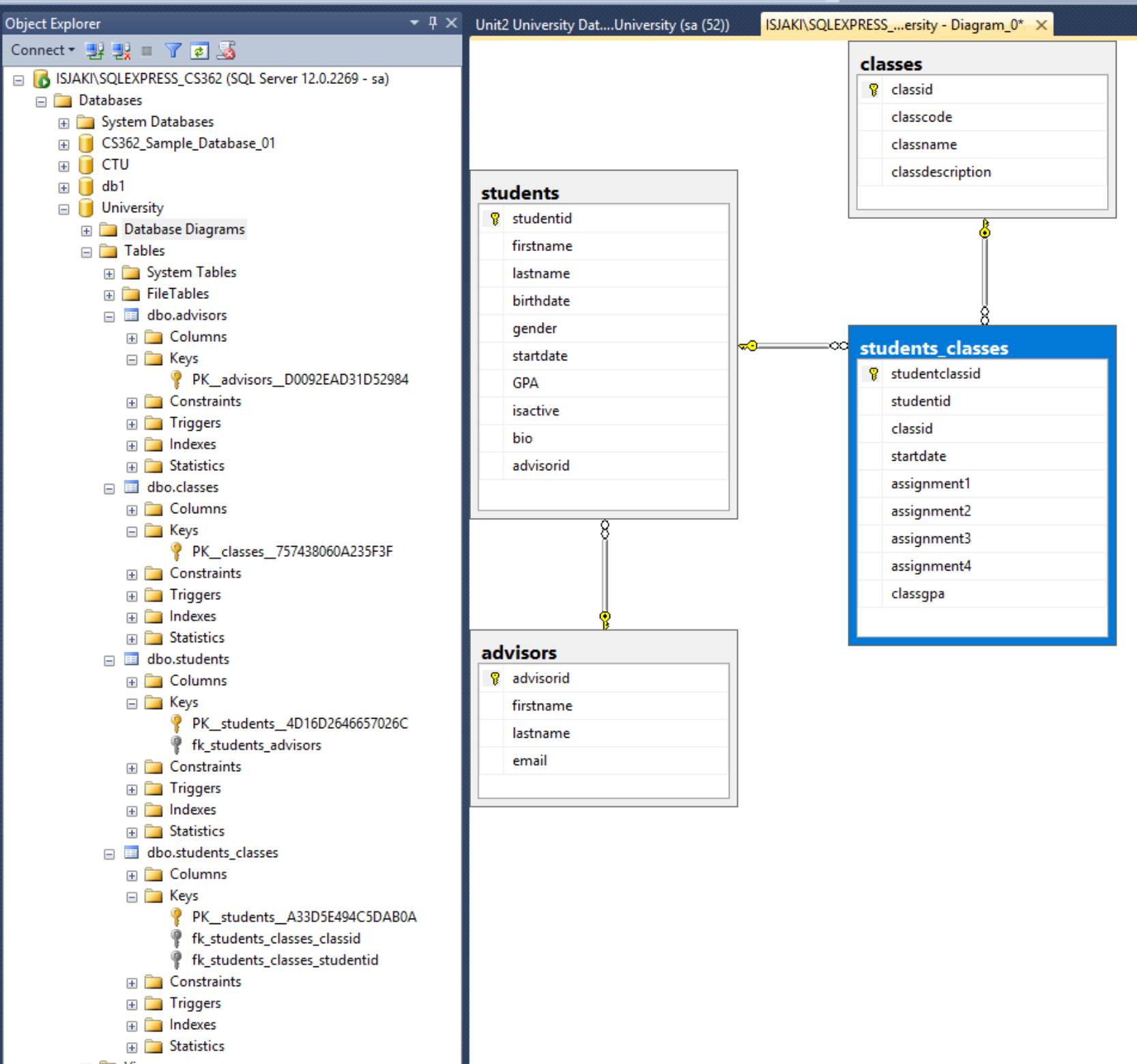
foreign key(studentid) references students(studentid);

alter table students\_classes check constraint fk\_students\_classes\_studentid;

alter table students\_classes with check add constraint fk\_students\_classes\_classid

foreign key(classid) references classes (classid);

alter table students\_classes check constraint fk\_students\_classes\_classid;



# IP 3

Introduction: The following SQL statements are important to database administration because they allow the creation, retrieval, updating, and destruction of table row elements (the data). These capabilities will be especially important when a user, such as a class scheduler, is tasked with working on these tables, as they will need various permissions to perform similar actions.

1. Insert the following classes' records

use University;

insert into classes (classcode, classname, classdescription)

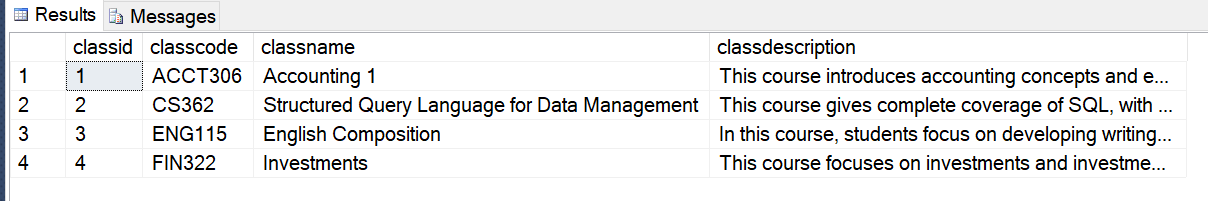
values ('ACCT306', 'Accounting 1', 'This course introduces accounting concepts and explores the accounting environment. It covers the basic structure of accounting, how to maintain accounts, use account balances to prepare financial statements, and complete the accounting cycle. It also introduces the concept of internal control and how to account for assets.'),

('CS362','Structured Query Language for Data Management ','This course gives complete coverage of SQL, with an emphasis on storage, retrieval, and manipulation of data.'),

('ENG115','English Composition','In this course, students focus on developing writing skills through practice and revision. Students will examine expository, critical, and persuasive essay techniques.'),

('FIN322','Investments','This course focuses on investments and investment strategies. Various investment vehicles such as stocks, bonds, and commodities are examined. Students will explore the principles of security analysis and valuation.');

select \* from classes;



1. Insert the following advisors' records

use University;

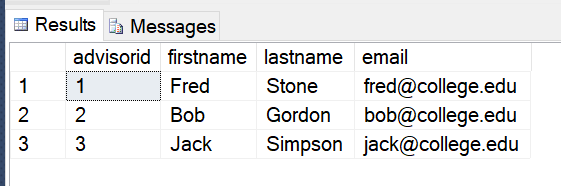
insert into advisors (firstname, lastname, email)

values ('Fred','Stone','fred@college.edu'),

('Bob','Gordon','bob@college.edu'),

('Jack','Simpson','jack@college.edu');

select \* from advisors;



1. Insert the following students' records

use University;

insert into students (firstname, lastname, birthdate, gender, startdate, GPA, isactive, bio, advisorid)

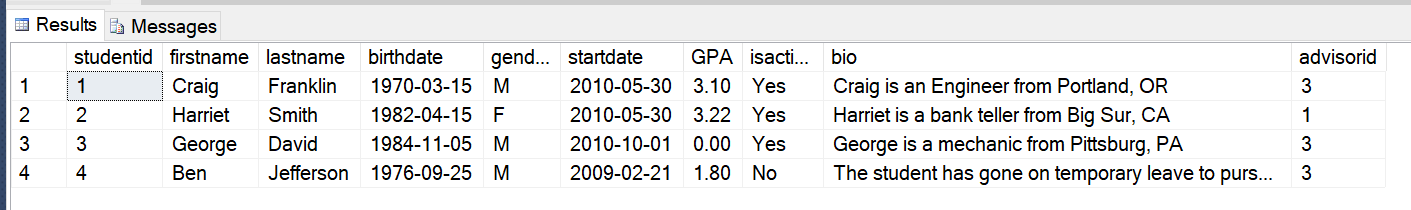
values ('Craig','Franklin','1970-03-15','M','2010-05-30',3.10,'Yes','Craig is an Engineer from Portland, OR',3),

('Harriet','Smith','1982-04-15','F','2010-05-30',3.22,'Yes','Harriet is a bank teller from Big Sur, CA',1),

('George','David','1984-11-05','M','2010-10-01',0.00,'Yes','George is a mechanic from Pittsburg, PA',3),

('Ben','Jefferson','1976-09-25','M','2009-02-21',1.80,'No','The student has gone on temporary leave to pursue other opportunities but plans on returning in 1 year.',3);

select \* from students;

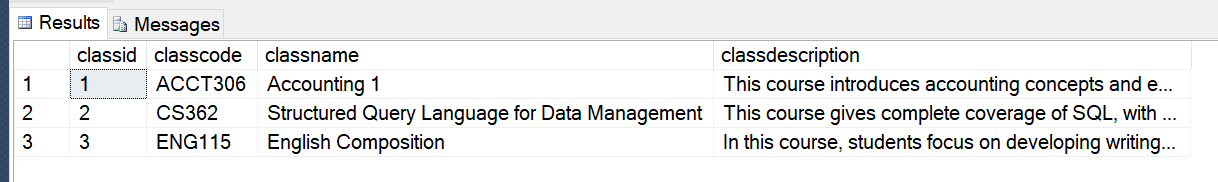


1. Delete the course named Investments from the system

use University;

delete from classes where classid = 4;

select \* from classes;

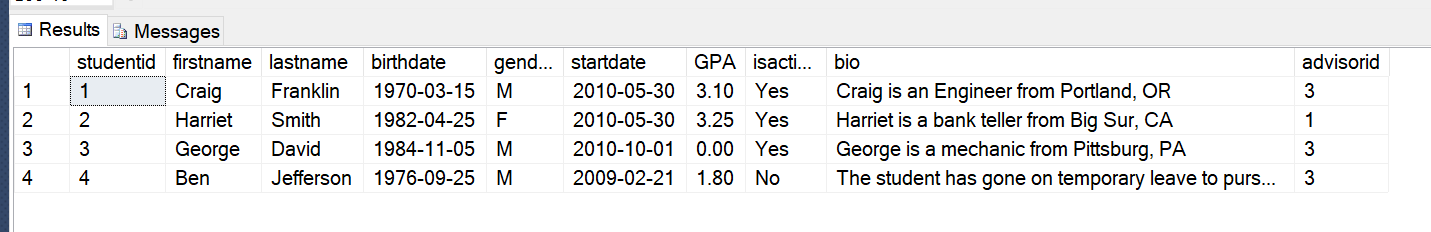


1. Change Harriet Smith’s birthdate to April 25, 1982 and her GPA to 3.25

use University;

update students set birthdate = '1982-04-25', GPA = 3.25 where studentid = 2;

select \* from students;



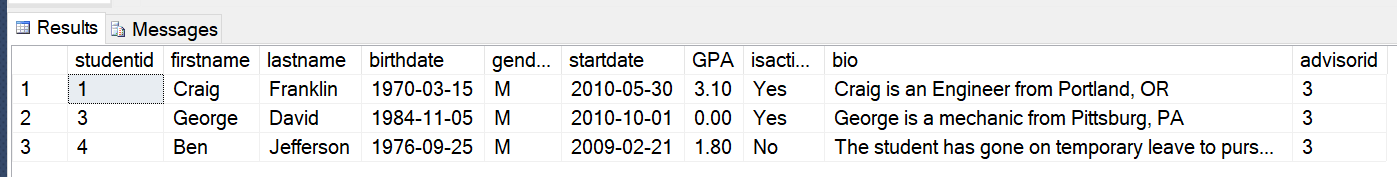
# IP 4

**List all active male students assigned to Advisors 1 or 3 (Fred Stone or Jack Simpson).**

select \* from students

where advisorid in (1,3)

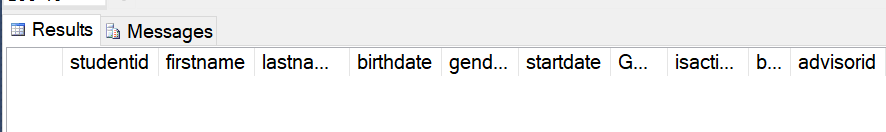
and gender = 'M';



**Provide a list of all students without a biography. *(all my students have biographies)***

select \* from students

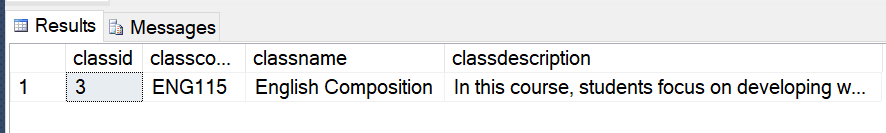
where bio like '';



**What classes are in the English department?**

select \* from classes

where classcode like 'ENG%';



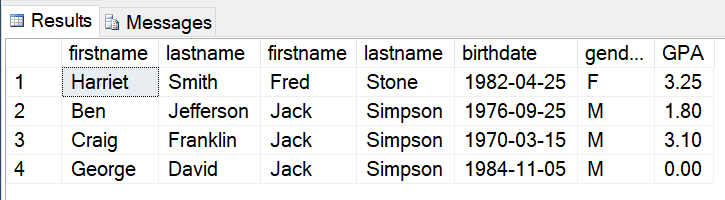
**Create a list of all students and their advisors. Sort by the advisor’s name and then the student’s name. Include the student’s birth date, gender, and GPA.**

select stu.firstname, stu.lastname, adv.firstname, adv.lastname, stu.birthdate, stu.gender, stu.GPA

from students stu inner join advisors adv

on stu.advisorid = adv.advisorid

order by adv.firstname, adv.lastname, stu.firstname, stu.lastname;

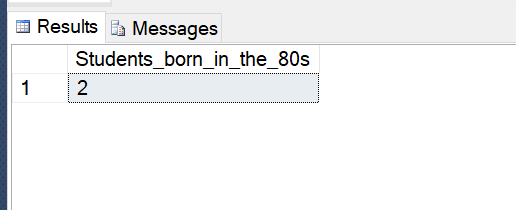


**How many students were born in the 1980s?**

select count(firstname) as Students\_born\_in\_the\_80s

from students

where birthdate like '198%';

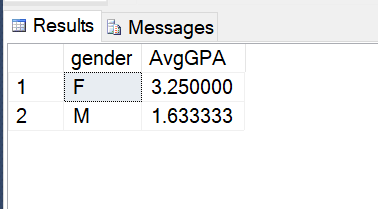


**Write a query to show the average GPA by gender.**

select gender, AVG(GPA) as AvgGPA

from students

group by gender;



**Provide a list of all advisors and the number of active students assigned to each.**

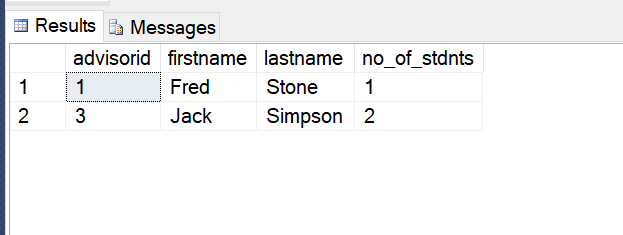
select s.advisorid, a.firstname, a.lastname, count(s.advisorid) as no\_of\_stdnts

from students s right join advisors a

on s.advisorid = a.advisorid

where s.isactive like 'Yes'

group by s.advisorid, a.firstname, a.lastname;



**Filter out any advisors with more than 1 student.**

select s.advisorid, a.firstname, a.lastname, count(s.advisorid) as no\_of\_stdnts

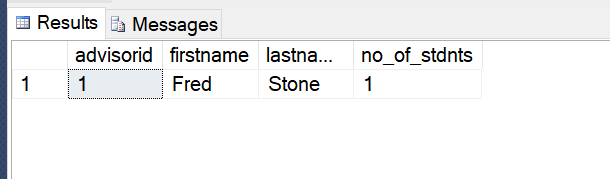
from students s right join advisors a

on s.advisorid = a.advisorid

where s.isactive like 'Yes'

group by s.advisorid, a.firstname, a.lastname

having count(s.advisorid) < 2;



# IP 5

Add tables “degree” and “degree\_classes”, add column “degreeid” to the “students” table, and set foreign keys between these two tables and the “students” and “classes” tables. Then tie degrees and classes together in the “degrees\_classes” table, as well as give some students some degrees.

1. choose database to work on

use University;

1. create the “degrees” table

create table degrees (

degreeid int identity(1,1) primary key not null,

degreename varchar(100) not null,

degreedesc varchar(1000) not null,

);

1. create the “degrees\_classes” table

create table degrees\_classes (

degreeclassid int primary key not null,

degreeid int not null,

classid int not null,

);

1. set foreign key between “degrees\_classes” table and “classes” table on column “classid”

alter table degrees\_classes with check add constraint

fk\_degrees\_classes\_classes foreign key (classid)

references classes (classid);

1. add column “degreeid” to “students” table

alter table students

add degreeid int;

1. set foreign key between “students” and “degrees” tables on column “degreeid”

alter table students with check add constraint

fk\_students\_degrees foreign key (degreeid)

references degrees (degreeid);

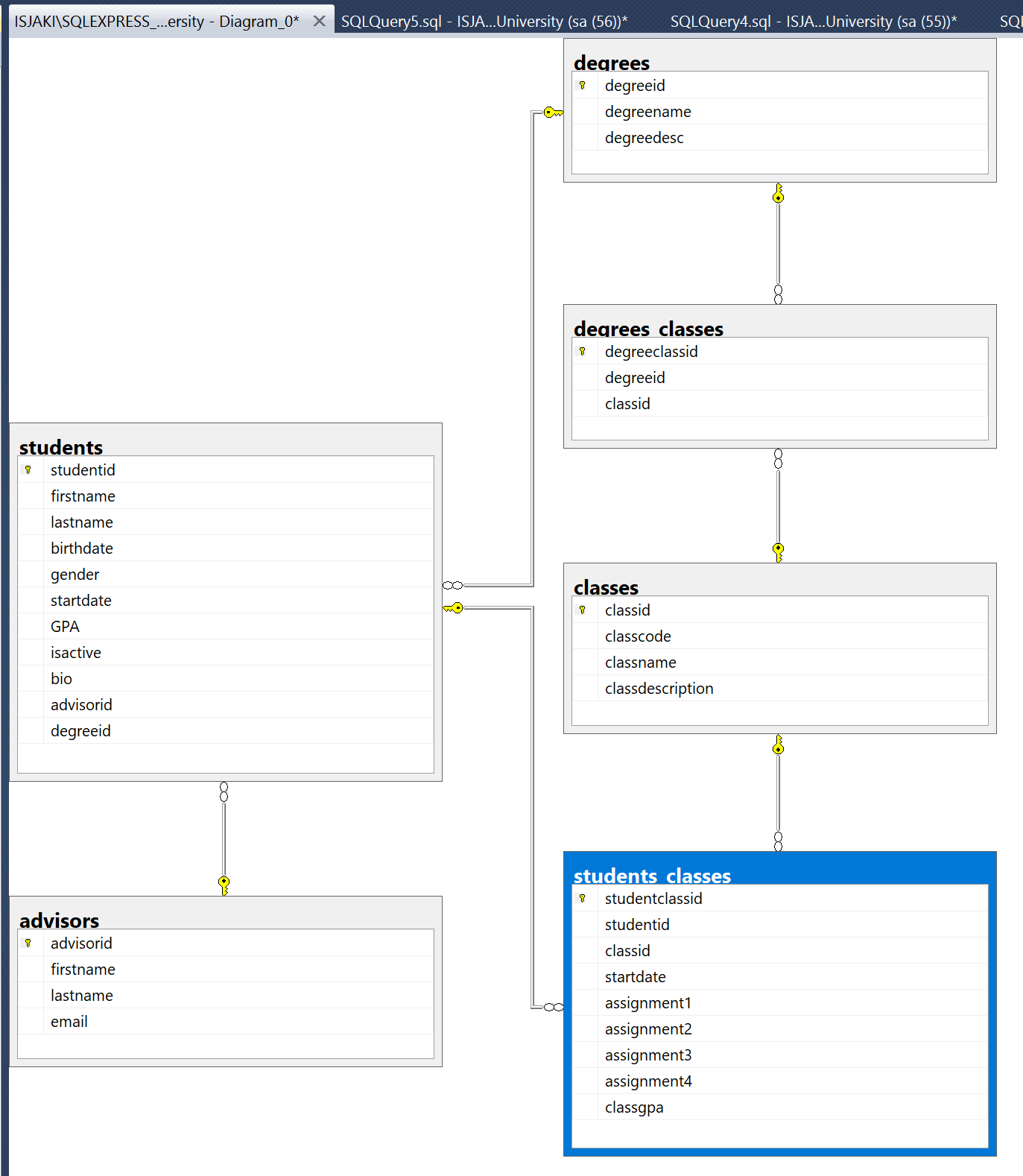
1. set foreign key between “degrees\_classes” and “degrees” tables on column “degreeid”

alter table degrees\_classes with check add constraint

fk\_degrees\_classes\_degrees foreign key (degreeid)

references degrees (degreeid);

1. Create a new database diagram



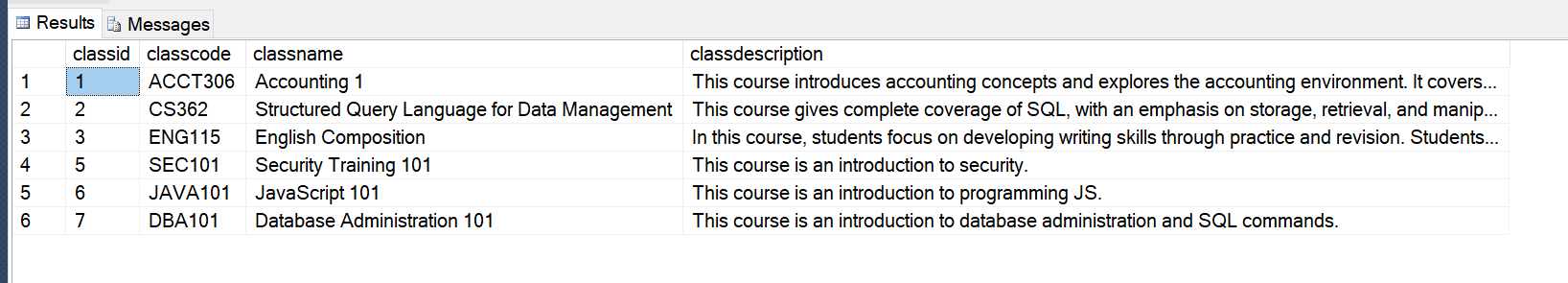
1. Add some more classes to the “classes” table

insert into classes (classcode, classname, classdescription)

values ('SEC101', 'Security Training 101', 'This course is an introduction to security.'),

('JAVA101', 'JavaScript 101', 'This course is an introduction to programming JS.'),

('DBA101', 'Database Administration 101', 'This course is an introduction to database administration and SQL commands.');



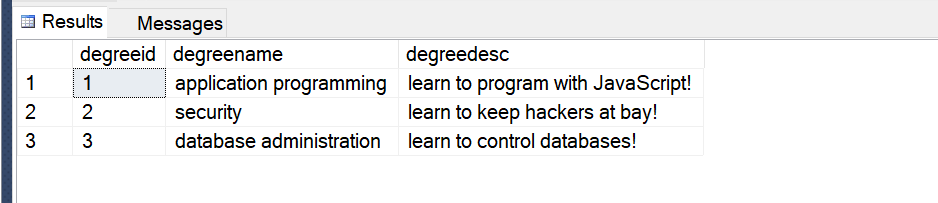
1. Add some degrees to the “degrees” table

insert into degrees (degreename, degreedesc)

values ('application programming', 'learn to program with JavaScript!'),

('security', 'learn to keep hackers at bay!'),

('database administration', 'learn to control databases!')



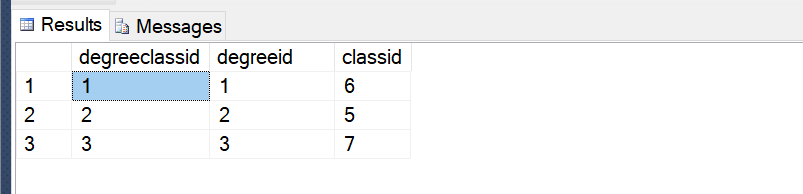
1. Tie some degrees to some classes with columns “degreeid” and “classid” in the “degrees\_classes” table

insert into degrees\_classes (degreeid, classid)

values (1, 6),

(2,5),

(3,7);



1. Give the students their degreeid’s

update students

set degreeid = 1

where studentid between 1 and 3;

update students

set degreeid = 3

where studentid = 4;

