

Syracuse University
School of Information Studies
IST 659 (*online*)
Summer 2017

School of Information Studies
Syracuse University

IST659 DATA ADMINISTRATION CONCEPTS AND DATABASE MANAGEMENT Final Project

Instructor: Chad Harper

Student: David Forteguerre
Email: dfortegu@syr.edu

I. PROJECT OVERVIEW

1. Project summary

For this final project, I decided to create and implement a database that will help me manage my students at Syracuse University.

As a graduate French instructor, I teach a daily lower-division 4-credit French course. My responsibilities include lesson planning, instruction, assessment (test preparation and grading), management of the textbook online component, and student advising. I also teach a weekly 1-credit conversation French course every other semester — depending on need. My responsibilities include syllabus design, lesson planning, instruction, and assessment (grading). Given that I teach a different course with twenty to thirty new students every semester and considering that I am a full-time graduate student at the same time, this project comes at the right time for me as I need to find a better, simpler, and faster way to keep track of important information about my students, their academic performance, and the courses I am teaching. Having to create several new Excel spreadsheets every semester, going back and forth between different files, and inserting/updating/deleting data in all my files whenever there is a need to do so can lead to many data inconsistencies, anomalies, and can also be very time-consuming. That is why I believe this student-management database will help me in many ways, as I finally will be able to stay organized and have consistent data records. As a result, this project turns out to be both interesting and practical in my case.

Most of the data that will eventually be inserted into the database (i.e. about my students, their performance, and about my courses) come from MySlice (i.e. class rosters and course catalogs) and Blackboard. Some other data (e.g. a student's country of origin, native language, foreign language skills, etc.) come from my personal records. At the beginning of every semester, I hand out a personal information form to my students in order to get to know them better. This helps me tailor the course content and organization to try and meet every student's individual needs (e.g. students whose native language is *not* a Romance or Germanic language may struggle more with grammar and vocabulary learning).

Considering that this project is for my personal use, I only intend to keep track of data that are relevant to *me* in the database. As a result, there will be no instructor entity, as I will be the only instructor teaching all the courses contained in the database. Likewise, even though there will be some information about textbooks and program coordinators, no separate entities will be created for such data types.

This project will remain private and will not be used for professional or other academic reasons as it presently is. *If I ever decide to use this database project again for different purposes in the future, all the sample data will be removed in order to preserve student confidentiality.*

The first part of this project will contain a broad overview of the database, its conceptual model (ERD), and its logical model. The second part will consist of the various steps needed for implementing it (from the SQL code needed to create the database; insert, update, and delete data; to the creation of views, functions, stored procedures, and forms.)

2. Tools and resources

1. Draw.io was used for conceptual modeling.
2. Vertabelo (free student version) was used for logical modeling. <https://www.vertabelo.com/>
3. SQL (Structured Query Language), an international standard for database manipulation, will be used in the second part of this project.
4. Microsoft SQL Server will be used as Database Management System (DBMS).
5. Several Syracuse University websites (notably MySlice and Blackboard) as well as my personal student records (mostly spreadsheets) will eventually be used for data retrieval and insertion into the database.
6. Microsoft SQL Server 2014 Management Studio.
7. Microsoft Access 2016.

3. Conventions

Here's a list of the conventions used in this project when creating diagrams:

- **Primary Keys** will be underlined. PKs are always **required** attributes, so underlying them will suffice.
- **Required attributes** will be marked as r.
- {Multivalued attributes} will be marked as m.
- Composite attributes (x_1, x_2, x_n) will be marked as c.
- [Derived attributes] will be marked as d.

II. CONCEPTUAL MODEL

1. Business rules, stakeholder(s), and data questions

1. Here is a list of the major business rules of this student-management database project:

Database-level rules:

- Only data about courses, students, and their performance in a given course will be kept in this database.
- A student *may* be enrolled in one or more courses.
- A course *may* have one or more students.
- A student *may* speak one or more languages, and one language *may* be spoken by one or more students.
- Every student enrolled in a course *must* have a performance and information record (i.e. grades, lateness, tardiness)

Entity-level rules:

- A student must have an SUID, first and last name, must be in a certain year of study (freshman, sophomore, junior, senior), must have at least one major, must let us know

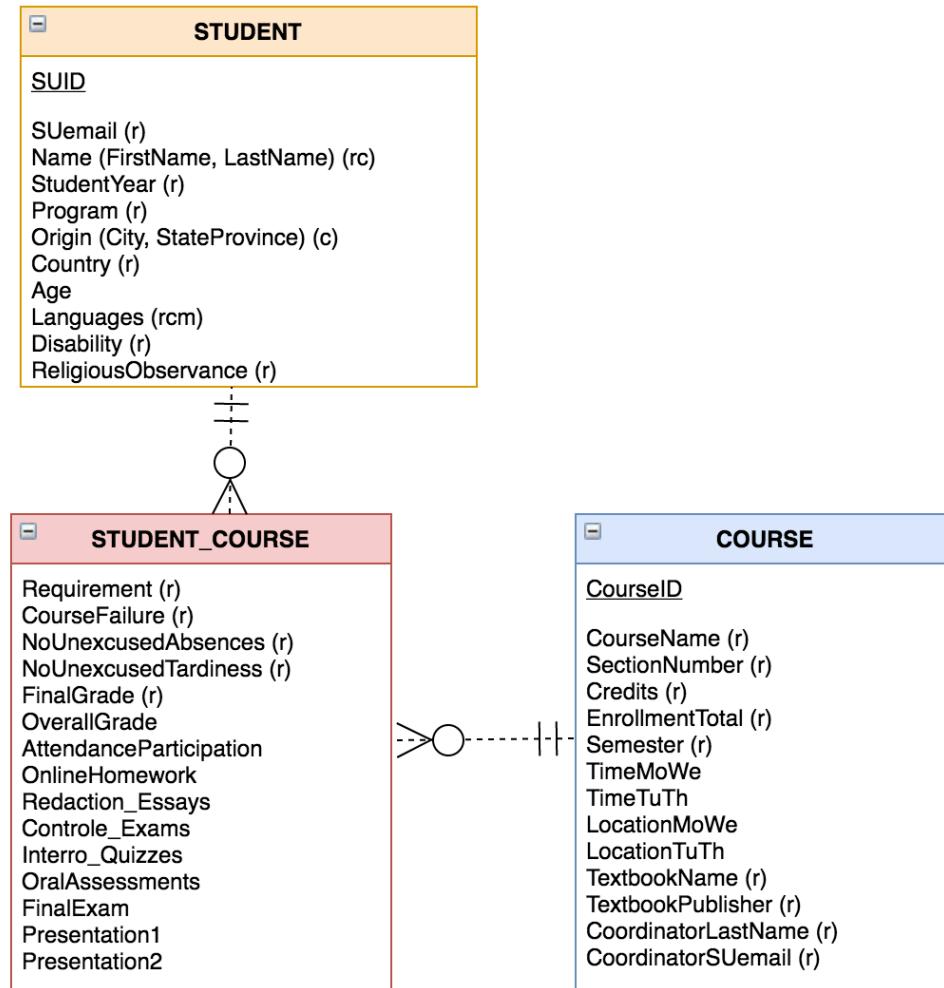
their country of origin and native language, and whether they have a religious observance and/or a disability. A student may speak one or more foreign languages.

- A course must have a Course ID, a name, a section number, a specific number of credits, a total number of enrolled students, a textbook name, and a language program coordinator.
 - A student's performance and information record must contain their number of absences and days they were late, the final letter grade they earned in the course, and must specify whether the course in question is a requirement for their degree or not, and whether they passed or failed it.
2. Considering that this database is for personal use only, the stakeholder for this project is myself. I will be the only person using and maintaining the database as well as the data.
3. Below is a list of five examples of data questions that this database will help answer:
1. **What is the personal information of a student enrolled in a given course?** (→ What is their first name, last name, SUID, and email address? Where does the student come from? What's their native language? Are they taking French to fulfill a degree requirement or is it for personal interest?)
 2. **What official grade did a specific student earn in a given course?** (→ What overall letter grade did they earn? What grade did they earn on their “Rédaction” essays specifically? Did they pass or fail the course?)
 3. **When did a student take a given course?** (→ What semester was it? Which year? What was the course level?)
 4. **How was a given student's attendance?** (→ How many times was the student in question absent and late to class in a given course? Did they pass or fail the course?)
 5. **How many students were there in a given course?** (→ What was the course level? When was it? Where was it?)

In the future, this database will also help me answer statistical questions thanks to the R programming language (e.g. What was the average age of students in a given course? What was the average overall grade of students in a given course? etc.)

2. Entity-Relationship Diagram (ERD)

Below is the ERD of my student-management database project. It shows the relationships of entity sets in the database.



3. ERD glossary & notes

1. Further details about entities are given right next to each entity name down below.
2. Further details about attributes will be given only when needed and not obvious from the name.
3. The attributes in STUDENT_COURSE that are in green constitute the breakdown of a student's overall grade as outlined in the course syllabus. However, I did not make them required in my database since they are just extra information that I do not necessarily need. What really matters to me is keeping the student's final (letter) grade in the database; the details about every component of the grade matter less but may still come in handy. Additionally, considering that Blackboard already automatically weighs each component of a student's grade as it should (e.g. the final exam is worth 20% of the overall grade) and gives us every student's final grade, I do not wish to include those calculations again in the database. However, I am aware that this must still be feasible by coding the rules and grade breakdown with SQL in the database, in order to factor in every component of the overall grade and weigh them properly. The result—the student's final grade—would appear in a derived attribute.

4. This ERD has two one-to-many relationships, as the STUDENT_COURSE entity is associative. This will be discussed again later on in the project.

5. In the STUDENT table, languages is a multivalued attribute because a student can speak one or several languages; it is also composite because a given language can be either a native language or a foreign language for a student.

STUDENT (*Collects data about students*)

SUID: unique 9-digit number identifying a student at SU.

SUemail (r): unique email address assigned to a student at SU.

Name (FirstName, LastName) (rc): (please note that FirstName will also include a student's middle name in this database, if any.)

StudentYear (r): options include *freshman*, *sophomore*, *junior*, *senior*, and *grad*. This data will be in the exact same format as the one given in the class roster on MySlice.

Program (r): the student's program of study. This data will be in the exact same format as the one given in the class roster on MySlice.

Origin (City, StateOrProvince) (c)

Country (r): *USA* will be used for the United States.

Age

Languages (rcm): the language(s) spoken by the student. Note that *French* will never be inserted as a language spoken by students; as the assumption is that they are taking a French course to learn the language.

Disability (r): whether the student has a disability that will require accommodations. Binary data type needed. Default value will be 0 (i.e. no).

ReligiousObservance (r): whether the student will have to miss class because of their faith. This information is available on MySlice and students must enter their religious observances in MySlice at the beginning of the course in order to make it official and be excused. Binary data type needed. Default value will be 0 (i.e. no).

STUDENT_COURSE (*Collects data about a student's info and performance in a given course*)

Requirement (r): whether the course is a requirement for the student. Binary data type needed.

CourseFailure (r): whether the student has failed the course (i.e. if they have an overall grade of below 60). Binary data type needed. Default value will be 0 (i.e. no).

NoUnexcusedAbsences (r): number of unexcused absences in a given course. Unexcused absences are defined in the course syllabus. For instance, in FRE201 (Fall16), students were allowed to have two undocumented absences with no impact on their grade; any other undocumented absence(s) would be marked as unexcused and would lower their participation grade.

NoUnexcusedTardiness (r): number of days a student was late to class. Repeated tardiness typically also affects a student's participation grade.

FinalGrade (r): the student's final letter grade; the official one that was posted to MySlice. Options include A, A-, B+, B, B-, C+, C, etc.

OverallGrade: this attribute corresponds to the "weighted total" column for a given student in the Blackboard grade center.

AttendanceParticipation
OnlineHomework
Redaction_Essays
Controle_Exams
Interro_Qizzes
OralAssessments
FinalExam
Presentation1
Presentation2

COURSE (*Collects data about a course. Most of the data will come from MySlice and Blackboard*)

CourseID: MySlice course ID.

CourseName (r): MySlice course name. Options include *FRE101, FRE102, FRE201, FRE202, FRE210*, which are the usual lower-division French courses taught by graduate instructors.

SectionNumber (r): MySlice section number. Options include *M001, M002, M003*, etc.

Credits (r): depends on the course type. Options include *4 or 1*.

EnrollmentTotal (r): number of students enrolled on the day following the withdrawal deadline. No data on students who dropped or withdrew from the class (and won't receive a grade) will be kept.

Semester (r): options include *Fall16, Spring17, Fall17, Spring18*; which are the only semesters I have been—and will be—an instructor at SU.

TimeMoWe: Monday and Wednesday class time.

TimeTuTh: Tuesday and Thursday class time.

LocationMoWe: Monday and Wednesday class location.

LocationTuTh: Tuesday and Thursday class location.

TextbookName (r): name of the textbook used with students. For FRE210 only, enter "No textbook" since no textbook is required for the conversation class.

TextbookPublisher (r): name of the company that publishes the textbook. For FRE210 only, enter "No textbook" since no textbook is required for the conversation class.

CoordinatorLastName (r): French program coordinator's last name.

CoordinatorSUemail (r): French program coordinator's SU email address.

4. Sample representative data (*unrefined*)

Below is a table containing sample data from SU students I have had in the past.

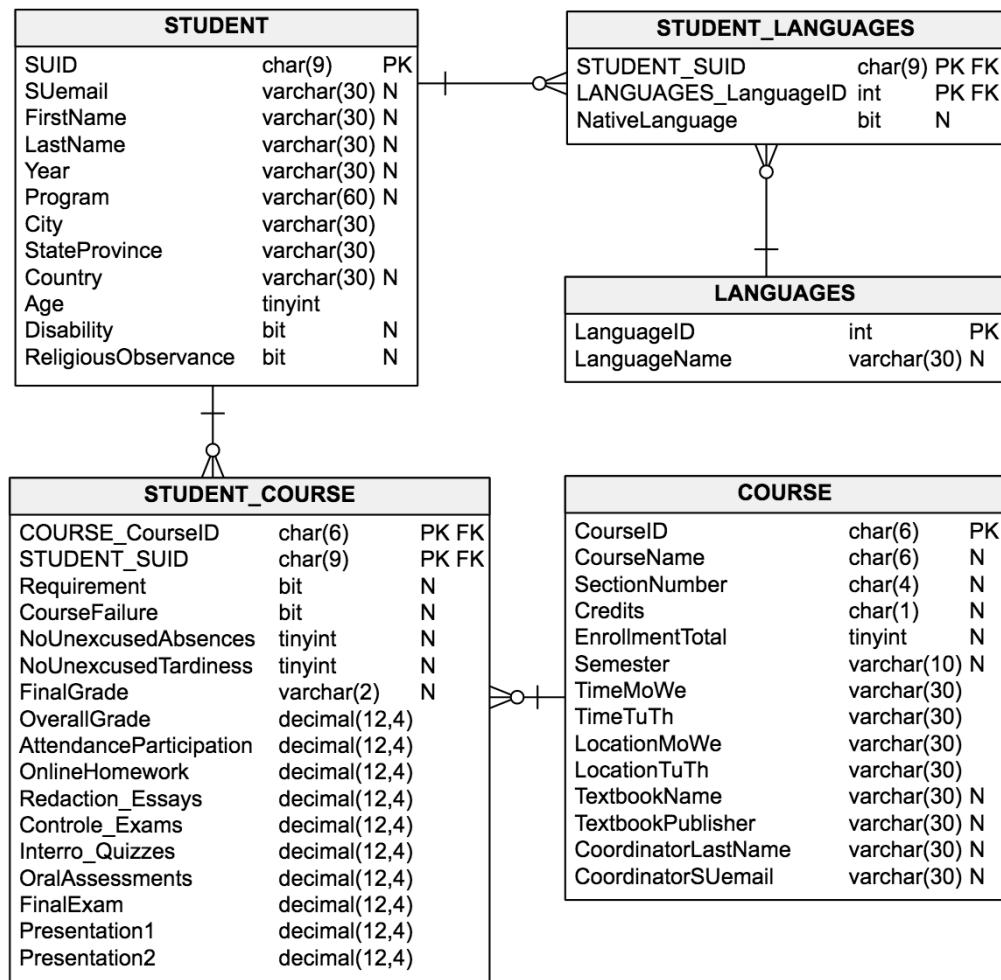
	EXAMPLE #1	EXAMPLE #2
STUDENT		
<u>SUID</u>	XXXXXXXXXX	XXXXXXXXXX
SUemail (r)	YYYYY@syr.edu	YYYYY@syr.edu
Name (FirstName, LastName) (rc)	YYYYYY, YYYYYY	YYYYYY, YYYYYY
StudentYear (r)	Freshman	Senior
Program (r)	Arts and Sciences - Arts and Sciences (Undeclared)	Public Communications - Television, Radio and Film
Origin (City, StateProvince) (c)	Trenton, NJ	Pittsburgh, PA
Country (r)	USA	USA
Age	21	21
Languages (rcm)	English, Russian, Italian***	English, Italian***
Disability (r)	0	0
ReligiousObservance (r)	0	0
STUDENT_COURSE		
Requirement (r)	1	0
CourseFailure (r)	0	0
NoUnexcusedAbsences (r)	1	0
NoUnexcusedTardiness (r)	0	0
FinalGrade (r)	B+	A
OverallGrade	87.77	95.26
AttendanceParticipation	89.23	95.71
OnlineHomework	89.90	
Redaction_Essays	89.00	
Controle_Exams	86.83	
Interro_Qizzes	84.50	
OralAssessments	94.33	
FinalExam	82.8	
Presentation1		90.00
Presentation2		98.00
COURSE		
<u>CourseID</u>	005687	005689
CourseName (r)	FRE201	FRE210
SectionNumber (r)	M005	M005
Credits (r)	4	1
EnrollmentTotal (r)	16	7
Semester (r)	Fall16	Spring17
TimeMoWe	10:35AM - 11:30AM	
TimeTuTh	11:00AM - 12:20PM	11:00AM - 12:20PM
LocationMoWe	Maxwell Hall 108	
LocationTuTh	CH001	HB CROUSE 200 located on quad
TextbookName (r)	Imaginez	No textbook
TextbookPublisher (r)	VHL	No textbook
Coordinator_LastName (r)	YYYYYY	YYYYYY
Coordinator_SUemail (r)	YYYYY@syr.edu	YYYYY@syr.edu

***This shows the need to create a separate table for languages as well as an associative entity linking students with languages.

III. NORMALIZED LOGICAL MODEL

Below you will find the normalized logical model of my student-management database project. We are now a step further in the database design, given that this model is DBMS-specific and includes more technical references (i.e. data types, referential integrity constraints, etc.)

1. Normalized Logical Model



2. About the Logical Model

a) Diagram and relationships

- After adding our referential integrity constraints, we notice that the STUDENT_COURSE relation is both an associative and weak entity. It is associative because it has the many sides of two one-to-many relationships; and it is weak because it has required foreign keys in it.

2. Given that a relation *cannot* contain any multivalued attributes in a logical model, we created a separate table for languages called LANGUAGES as well as an associative entity called STUDENT_LANGUAGES to link LANGUAGES with STUDENT. This associative entity will allow us to add language(s) for each student specifying whether each new language entry is a native language or a foreign language for a given student thanks to a binary data type (i.e. 1 = native language; 0 = foreign language.)

b) Data types

1. Many data types in this logical model are very specific (e.g. char(9) for SUID, char(6) for CourseID, etc.) because those IDs already have fixed lengths determined by the university.
2. Integer data types (int) are only used for attributes that will eventually involve potential calculations and statistics. For instance, using (int) will help me see how many students I have had in total, or the average age of my students in a given course, or the number of absences in a given course, etc.
3. For percentages, decimal(12,4) was used. However, most percentages will have the following format: "100.00".

c) Normalization

A database design is said to be in first normal form (1NF) when all the multivalued attributes have been removed, in second normal form (2NF) when all the partial dependencies have been removed, and in third normal form (3NF) when all the transitive dependencies have been removed.

Not all the tables in this database project were converted to 2NF or 3NF by choice. For instance, the COURSE table as shown in the logical model above is still in 1NF; that is because most of the data about students and courses that will eventually be inserted into this database will be retrieved from a specific source (i.e. MySlice) and reported in the database *in its original format*. That also explains my data type choices; some of them could be better if the data did not eventually come from MySlice.

Besides, it is important to keep in mind that even though the level of details and accuracy of this database design could be improved in several ways (e.g. a separate relation could be created to keep records on textbook publishers; another to list the countries the students come from; another to list language program coordinators; another to list programs of study; etc.), this project should still remain moderate in size and scope, as again, my ultimate goal is to keep records of *students and courses*.

IV. PHYSICAL DATABASE DESIGN

In this section, I will provide my Data Definition Language (DDL) code to build my database tables and any objects such as stored procedures, views, and functions in Microsoft SQL Server 2014 Management Studio. Note that I will only create views and functions in this project; there will not be any stored procedures (approved by Prof. Harper.) I will also include the necessary code to remove those objects.

Below is the code.

a) TABLES Creation

Please see code in SQL file.

Screenshot:

The screenshot shows the Microsoft SQL Server Management Studio interface. The left pane displays the Object Explorer with a tree view of databases, tables, and other objects. The right pane contains two DDL scripts for creating tables. The first script creates the 'Student_Course' table with various columns like Course_CourseID, Student_SUID, and FinalGrade, along with constraints and a primary key. The second script creates the 'Student_Languages' table with columns for Student_SUID and Languages_LanguageID, also with a primary key constraint. Both scripts end with a semicolon. At the bottom of the right pane, a message box indicates that the command(s) completed successfully.

```
-- TABLE CREATION: STUDENT_COURSE
CREATE TABLE Student_Course (
    Course_CourseID char(6) FOREIGN KEY REFERENCES Course(CourseID),
    Student_SUID char(9) FOREIGN KEY REFERENCES Student(SUID),
    Requirement bit NOT NULL,
    CourseFailure bit DEFAULT '0' NOT NULL,
    NoUnexcusedAbsences tinyint NOT NULL,
    NoUnexcusedTardiness tinyint NOT NULL,
    FinalGrade varchar(2) NOT NULL,
    OverallGrade decimal(12,4),
    AttendanceParticipation decimal(12,4),
    OnlineHomework decimal(12,4),
    Redaction_Essays decimal(12,4),
    Controle_Exams decimal(12,4),
    Interro_Quizzes decimal(12,4),
    OralAssessments decimal(12,4),
    FinalExam decimal(12,4),
    Presentation1 decimal(12,4),
    Presentation2 decimal(12,4),
    CONSTRAINT PK_Student_Course PRIMARY KEY (Course_CourseID, Student_SUID)
);

-- TABLE CREATION: STUDENT_LANGUAGES
CREATE TABLE Student_Languages (
    Student_SUID char(9) FOREIGN KEY REFERENCES Student(SUID),
    Languages_LanguageID int FOREIGN KEY REFERENCES Languages(LanguageID),
    NativeLanguage bit NOT NULL,
    CONSTRAINT PK_Student_Languages PRIMARY KEY (Student_SUID, Languages_LanguageID)
);
```

NOTES:

I had to change a few attribute names while writing the code in this step, as some of them already existed in SQL Server as *key words*. The changes I made are also reflected in the first part of the project (even in the diagrams):

- In STUDENT, Year → **StudentYear**
- LANGUAGE (table name) → **LANGUAGES**
- STUDENT_LANGUAGE (table name) → **STUDENT_LANGUAGES**

- In STUDENT_LANGUAGES, Language_LanguageID → Languages_LanguageID

Besides, just for your information, I executed the **V. DATA CREATION** step before creating the views and functions below to be able to test them and make sure they returned the right data.

b) VIEWS Creation

In this section, I will create a few views that will come in handy in the future for me to look up data. Below is the code with explanations.

Please see code in SQL file.

Screenshot:

```

-- VIEW Creation: Return the number of courses students took
IF EXISTS (SELECT * FROM INFORMATION_SCHEMA.VIEWS WHERE TABLE_NAME = 'StudentCoursesView')
BEGIN
    DROP VIEW StudentCoursesView
END
GO

CREATE VIEW StudentCoursesView AS

SELECT
    Student.SUID, Student.FirstName, Student.LastName, Student.Country,
    COUNT(Course.CourseID) AS "Number of courses taken"
FROM Student
JOIN Student_Course ON Student_Course.Student_SUID = Student.SUID
JOIN Course ON Course.CourseID = Student_Course.Course_CourseID
GROUP BY Student.SUID, Student.FirstName, Student.LastName, Student.Country
GO

-- Test:
SELECT * FROM StudentCoursesView
GO

```

Total Number of Courses Taught	Total Number of Students
3	42

SUID	FirstName	LastName	SUemail	StudentYear	Program	Country	Disability	ReligiousObservance
1	[REDACTED]	[REDACTED]	@syr.edu	Freshman	Arts and Sciences - Arts and Sciences (Undeclared)	USA	0	0

SUID	FirstName	LastName	Country	Number of languages spoken
1	Daniel	[REDACTED]	Panama	2
2	Nora	[REDACTED]	USA	3
3	Kyle	[REDACTED]	USA	1

SUID	FirstName	LastName	Country	Number of courses taken
1	[REDACTED]	Nora	[REDACTED]	1

Query executed successfully. ist-s-students.syr.edu (12.... | AD\dfortegu (67)

c) FUNCTIONS Creation

In this section, I will create a few functions that will come in handy in the future for me to look up data. Below is the code with explanations.

Please see code in SQL file.

Screenshot:

```
DROP FUNCTION dbo.CourseID ;
GO

CREATE FUNCTION dbo.CourseID (@CourseName char(6), @Semester varchar(10))
--Note that those two local variables are enough to return a specific course ID, since as stated in this project introduction (see first page), I
--... per semester. I cannot teach two sections of the same course each semester. Therefore, those two variables are enough to retrieve the info
RETURNS char(6)
AS
BEGIN
    RETURN (
        SELECT Course.CourseID
        FROM Course
        WHERE Course.CourseName = @CourseName AND Course.Semester = @Semester
    )
END
GO

-- Test (FRE101, Fall17, CourseID 005681)
SELECT dbo.CourseID('FRE101', 'Fall17') AS "Course ID"
GO
-- Test (FRE201, Fall16, CourseID 005687)
SELECT dbo.CourseID('FRE201', 'Fall16') AS "Course ID"
GO
```

Student Name
1 Adam
Student Name
1 Kyle
Student SUID
1 [REDACTED]
Student SUID
1 [REDACTED]
Course ID

Query executed successfully.

V. DATA ENTRY

In this section, I will provide a few representative examples of how my data will be created. Note that no safety controls (e.g. creating another user and granting/restricting permissions) will be added in this project, as I am the only person who will ever use this database (approved by Prof. Harper.)

Below is the code for data creation.

Please see code in SQL file.

Screenshot:

The screenshot shows the SQL Server Management Studio interface. On the left is the Object Explorer tree view, which lists various database objects like tables, views, and stored procedures. The main area contains two query windows. The top window is titled "SQLQuery2 Data Ins... (AD\dfortegu (71))" and displays a multi-line T-SQL script. The bottom window is titled "SQLQuery1 tables c... (AD\dfortegu (70))" and displays the results of a SELECT query. The results grid has columns for SUID, SUemail, FirstName, LastName, StudentYear, Program, City, StateProvince, Country, Age, Disability, and ReligiousObservance. The bottom status bar indicates the query was executed successfully.

SUID	SUemail	FirstName	LastName	StudentYear	Program	City	StateProvince	Country	Age	Disability	ReligiousObservance
1	[REDACTED]	Rachel	[REDACTED]	Freshman	Arts and Sciences - Psychology-U	NULL	NULL	USA	NULL	0	0
2	[REDACTED]	Adam	[REDACTED]	Sophomore	Arts and Sciences - Forensic Science-U	NULL	NULL	USA	NULL	0	0
3	[REDACTED]	Amie	[REDACTED]	Freshman	VPA-Music - Recording & Allied Ent. Indust	NULL	NULL	USA	NULL	0	0
4	[REDACTED]	Natalie	[REDACTED]	Senior	Public Communications - Television, Radio and Film	NULL	NULL	USA	NULL	0	0
5	[REDACTED]	Zoe	[REDACTED]	Junior	Public Communications - Newspaper & Online Journ...	NULL	NULL	Australia	NULL	0	0
6	[REDACTED]	Daniel	[REDACTED]	Sophomore	Public Communications - Television, Radio and Film	NULL	NULL	Panama	NULL	0	0
7	[REDACTED]	Nora	[REDACTED]	Freshman	Arts and Sciences - Arts and Sciences (Undeclared)	NULL	NULL	USA	NULL	0	0
8	[REDACTED]	Kyle	[REDACTED]	Sophomore	Public Communications - Advertising	NULL	NULL	USA	NULL	0	0

CourseID	CourseName	SectionNumber	Credits	EnrollmentTotal	Semester	TimeMoWe	TimeTuTh	LocationMoWe	LocationTuTh	TextbookName	TextbookPublisher	Co
1	005681	FRE101	M001	4	19	Fall17	12:45PM - 1:40PM	12:30PM - 1:25PM	Link Hall 200	Marshall Square Mall 205C	Portals	VHL
2	005687	FRE201	M005	4	16	Fall16	10:35AM - 11:30AM	11:00AM - 12:20PM	Maxwell Hall 108	CH001	Imaginez	VHL
3	005689	FRE210	M005	1	7	Spring17	NULL	11:00AM - 12:20PM	NULL	HB CROUSE 200 locate...	No textbook	No textbook

VI. DATA MANIPULATION

In this section, I will provide a few representative examples of how data are manipulated through updates and deletions. Again, note that no safety controls (e.g. creating another user and granting/restricting permissions) will be added in this project, as I am the only person who will ever use this database (approved by Prof. Harper.)

Please see code in SQL file.

Screenshot:

The screenshot shows the Microsoft SQL Server Management Studio (SSMS) interface. On the left is the Object Explorer tree view, which lists various database objects like Database Diagrams, Tables, Views, Synonyms, Programmability, Functions, and System Functions. The central pane displays a SQL query window titled "SQLQuery2.sql - ist... (AD\dfortegu (65))". The query contains several UPDATE and DELETE statements with comments explaining the purpose of each operation. The bottom pane shows the "Messages" window with three rows of output indicating successful execution: "(1 row(s) affected)", "(1 row(s) affected)", and "(1 row(s) affected)".

```
-- We are now a few weeks into the semester, and students filled out their Personal Information Form.  
-- It is now time to update the STUDENT table. Please note that the data inserted below is fictitious.  
  
UPDATE Student SET ReligiousObservance = 1 WHERE Student.LastName = 'Black'  
UPDATE Student SET ReligiousObservance = 1 WHERE Student.LastName = 'White'  
UPDATE Student SET Country = 'Ecuador' WHERE Student.LastName = 'Black' --After realizing I didn't enter the right country for this student  
UPDATE Student SET FirstName = 'Amy' WHERE Student.LastName = 'Black' --After realizing I mispelt her first name in the database  
UPDATE Student SET City = 'Melbourne', StateProvince = 'Victoria', Age = '20' WHERE Student.LastName = 'Black' --After reading her personal info  
GO  
  
-- DELETIONS (examples)  
  
-- One of my students decided to withdraw from the course at the last minute. Let's delete her from the database.  
DELETE FROM Student WHERE Student.LastName = 'Black'  
-- One of the languages we recently added to the database isn't actually spoken by any of the students. Let's delete 'Bulgarian' from the database  
DELETE FROM Languages WHERE Languages.LanguageName = 'Bulgarian'  
GO
```

Messages

```
(1 row(s) affected)  
(1 row(s) affected)  
(1 row(s) affected)
```

VII. ANSWERING DATA QUESTIONS

In this part, I will code and demonstrate through screenshots the results of the SELECT statements that answer the data questions posed in the summary. I will give one sample answer per question. The questions were:

1. What is the personal information of a student enrolled in a given course?
2. What official grade did a specific student earn in a given course?
3. When did a student take a given course?
4. How was a given student's attendance?
5. How many students were there in a given course?

Below is the code. Note that each question requires different types of SELECT sand JOIN statements.

Please see code in SQL file.

Screenshot:

The screenshot shows the SQL Server Management Studio interface. On the left is the Object Explorer pane, which lists various database objects like tables, views, and stored procedures. The main area is a query window titled 'SQLQuery1 Answer... (AD\dfortegu (87))'. It contains five distinct SQL scripts, each starting with a comment block (e.g., '-- Let's retrieve Nora's attendance information.', '-- 5. How many students were there in a given course?'). The fifth script is a SELECT statement for 'EnrollmentTotal' from 'Course' where 'CourseName = 'FRE101'' and 'Semester = 'Fall17''. Below the query window is a 'Results' tab showing two result sets. The first result set is for 'FinalGrade' and 'Semester', showing 'B+' and 'Fall16' respectively. The second result set is for 'EnrollmentTotal', showing '19'. The bottom of the screenshot includes a caption '(Contains all 5 answers.)'.

```
-- Let's retrieve Nora's attendance information.
SELECT
    Student_Course.NoUnexcusedAbsences,
    Student_Course.NoUnexcusedTardiness
FROM Student_Course
JOIN Student ON Student_Course.Student_SUID = Student.SUID
JOIN Course ON Course.CourseID = Student_Course.Course_CourseID
WHERE Lastname = [REDACTED] AND Course.CourseName = 'FRE101'
GO

-- ANSWER: Nora had 1 unexcused absence, and she'd never been late to class.

-- 5. How many students were there in a given course?

-- Let's retrieve the number of students I have this semester (Fall17).
SELECT
    EnrollmentTotal
FROM Course
WHERE CourseName = 'FRE101' AND Semester = 'Fall17'
GO

-- ANSWER: I have 19 students in my FRE101 course this semester.
```

SUID	SUemail	FirstName	LastName	StudentYear	Program	City	StateProvince	Country	Age	Disability	ReligiousObservance
[REDACTED]	[REDACTED]@syry.edu	Nora	[REDACTED]	Freshman	Arts and Sciences - Arts and Sciences (Undeclared)	NULL	NULL	USA	NULL	0	1

FinalGrade
B+

Semester
Fall16

NoUnexcusedAbsences	NoUnexcusedTardiness
1	0

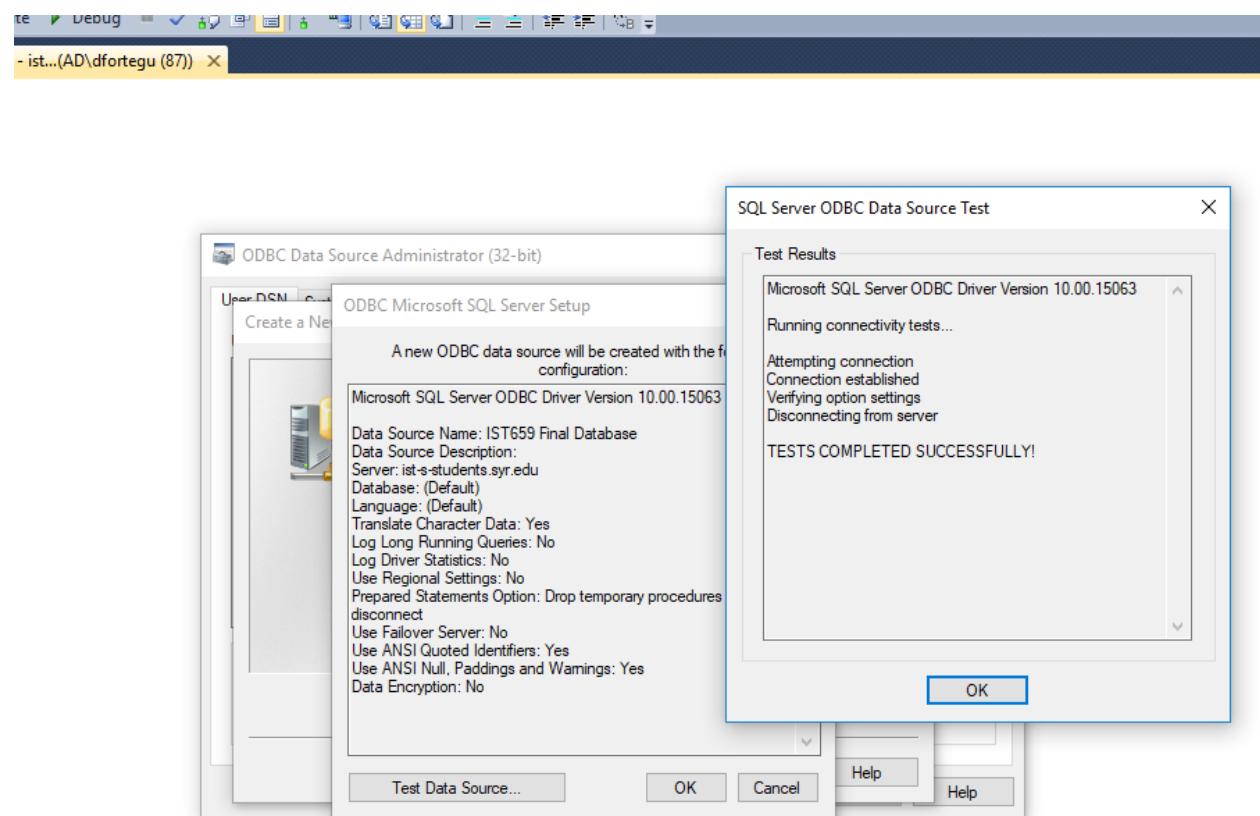
EnrollmentTotal
19

(Contains all 5 answers.)

VIII. IMPLEMENTATION (MICROSOFT ACCESS)

In this part, I will use Microsoft Access 2016 to build a basic front end that provides a user interface for maintaining and reporting on my data. I will provide data entry and maintenance screens, and report outputs.

1. Creation of the ODBC connection to the database

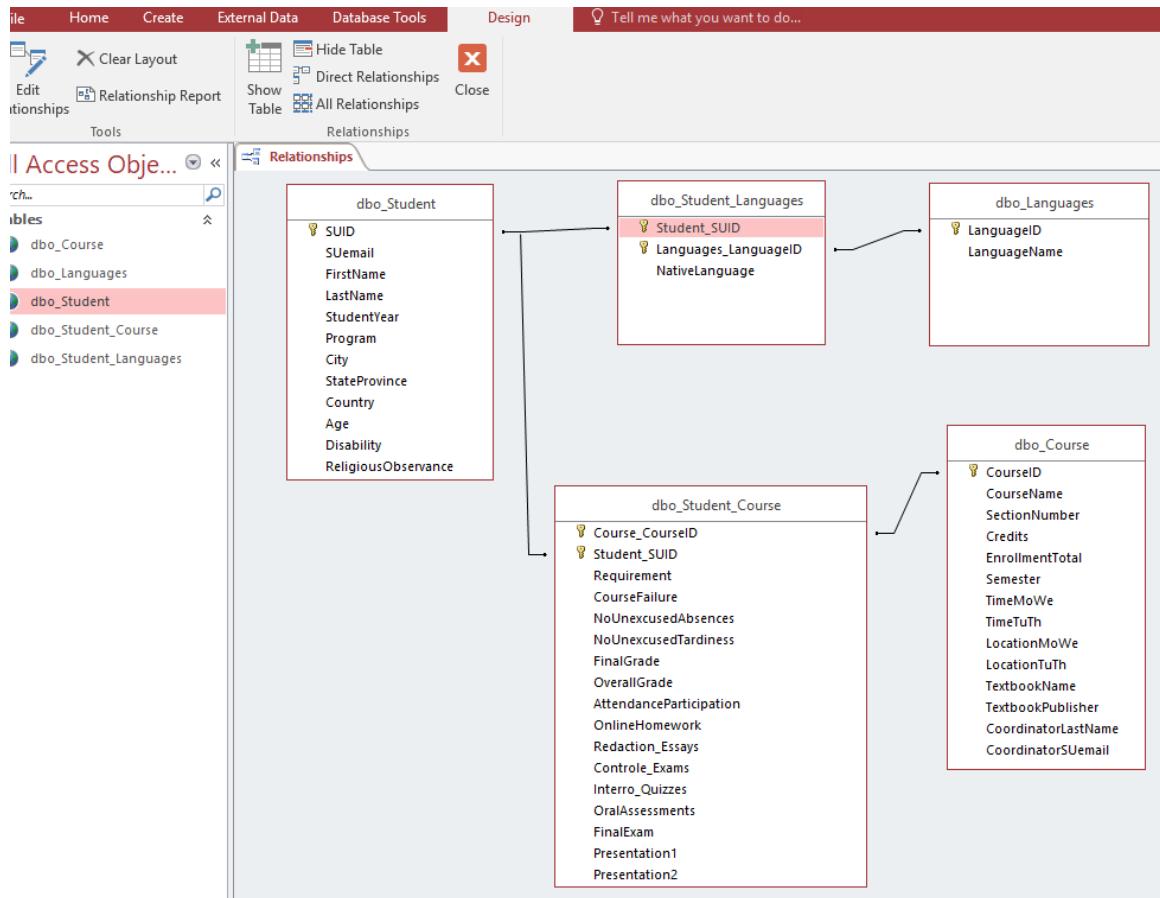


2. Microsoft Access database creation and link to SQL Server

The screenshot shows the Microsoft Access ribbon at the top with various tabs like Home, Create, External Data, Database Tools, Fields, Table, and Tell me what you want to do... Below the ribbon, there's a toolbar with icons for Text File, XML File, Database, More, Access, Saved, Excel, Text, XML, PDF, Email, Word Merge, and More. A sidebar on the left lists tables: dbo_Course, dbo_Languages, dbo_Student (which is selected and highlighted in red), dbo_Student_Course, and dbo_Student_Languages. The main area displays the 'dbo_Student' table with columns: SUID, SUemail, FirstName, LastName, StudentYear, Program, City, StateProvince, Country, Age, and Disability. There are 8 rows of data.

SUID	SUemail	FirstName	LastName	StudentYear	Program	City	StateProvince	Country	Age	Disability
[REDACTED]	[REDACTED]@syr.edu	Adam	[REDACTED]	Sophomore	Arts and Sciences			USA	0	
[REDACTED]	[REDACTED]@syr.edu	Amy	[REDACTED]	Freshman	VPA-Music - Recreational			USA	0	
[REDACTED]	[REDACTED]@syr.edu	Natalie	[REDACTED]	Senior	Public Communication			USA	0	
[REDACTED]	[REDACTED]@syr.edu	Zoe	[REDACTED]	Junior	Public Communication	Melbourne	Victoria	Australia	20	0
[REDACTED]	[REDACTED]@syr.edu	Daniel	[REDACTED]	Sophomore	Public Communication			Ecuador	0	
[REDACTED]	[REDACTED]@syr.edu	Nora	[REDACTED]	Freshman	Arts and Sciences			USA	0	
[REDACTED]	[REDACTED]@syr.edu	Kyle	[REDACTED]	Sophomore	Public Communication			USA	0	

3. Screenshot of completed Microsoft Access relationships window



4. Data entry and maintenance screens

SUID	SUemail	FirstName	LastName	StudentYear	Program	City	StateProvince	Country
	@syr.edu	Adam		Sophomore	Arts and Sciences		USA	
	@syr.edu	Amy		Freshman	VPA-Music - Rec.		USA	
	@syr.edu	Natalie		Senior	Public Commu		USA	
	@syr.edu	Zoe		Junior	Public Commu	Melbourne	Victoria	Australia
	@syr.edu	Daniel		Sophomore	Public Commu			Ecuador
	@syr.edu	Nora		Freshman	Arts and Sciences		USA	
	@syr.edu	Kyle		Sophomore	Public Commu		USA	

CourseID	CourseName	SectionNum	Credits	EnrollmentT	Semester	TimeMoWe	TimeTuTh	LocationMc
005681	FRE101	M001	4		19 Fall17	12:45PM - 1:40	12:30PM - 1:25	Link Hall 20
005687	FRE201	M005	4		16 Fall16	10:35AM - 11:30	11:00AM - 12:20	Maxwell Hall
005689	FRE210	M005	1		7 Spring17		11:00AM - 12:20	

SS Object Explorer

Clipboard

Sort & Filter

Records

Find

dbo_Student dbo_Course dbo_Languages dbo_Student_Course

LanguageID	LanguageName
9	English
10	Spanish
11	Italian
12	Russian
*	(New)

Access Object Explorer

Clipboard

Sort & Filter

Records

Find

Calibri

Text

dbo_Course dbo_Languages dbo_Student dbo_Student_Course

Course_Cou	Student_SUI	Requirement	CourseFailure	NoUnexcuse	NoUnexcuse	FinalGrade
005687	[REDACTED]	1	0	1	0	B+
*						

5. Forms creation

a) Screenshot: ADD STUDENT Form

ADD STUDENT FORM

SUID	[Redacted]
SUemail	[Redacted]@syr.edu
FirstName	Adam
LastName	[Redacted]
StudentYear	Sophomore
Program	Arts and Sciences - Forensic Science-U
Country	USA
Disability	False
ReligiousObservance	False
LANGUAGES Subform	
Language Name	Native Language?
English	True
Spanish	(Selected)
Italian	
Russian	

b) Screenshot: ADD COURSE Form

ADD COURSE FORM

CourseID	005681
CourseName	FRE101
SectionNumber	M001
Credits	4
EnrollmentTotal	19
Semester	Fall17
TimeMoWe	12:45PM - 1:40PM
TimeTuTh	12:30PM - 1:25PM
LocationMoWe	Link Hall 200
LocationTuTh	Marshall Square Mall 205C
TextbookName	Portails
TextbookPublisher	VHL
CoordinatorLastName	[Redacted]
CoordinatorSUemail	[Redacted]@syr.edu

c) Screenshot: ADD STUDENT PERFORMANCE Form

The screenshot shows the Microsoft Access 'ADD STUDENT PERFORMANCE' form. The form contains fields for Course_CourseID, Student_SUID, Requirement, CourseFailure, NoUnexcusedAbsences, NoUnexcusedTardiness, FinalGrade, OverallGrade, AttendanceParticipation, OnlineHomework, Redaction_Essays, Controle_Exams, Interro_Quizzes, OralAssessments, FinalExam, Presentation1, and Presentation2. The 'Requirement' field dropdown is currently open, displaying three options: '005681 FRE101', '005687 FRE201' (which is selected), and '005689 FRE210'. The 'CourseFailure' field has a dropdown menu showing 'False'. The 'NoUnexcusedAbsences' field contains the value '1'. The 'NoUnexcusedTardiness' field contains the value '0'. The 'FinalGrade' field contains 'B+'. The 'OverallGrade' field is empty. The other fields (AttendanceParticipation, OnlineHomework, Redaction_Essays, Controle_Exams, Interro_Quizzes, OralAssessments, FinalExam, Presentation1, Presentation2) are also empty.

Requirement	Value
005681	FRE101
005687	FRE201
005689	FRE210

6. Screenshot: STUDENTS RECORDS Report

The screenshot shows the Microsoft Access application interface. On the left, the navigation pane displays various database objects: Tables (dbo_Course, dbo_Languages, dbo_Student, dbo_Student_Course, dbo_Student_Languages), Queries (StudentList), Forms (ADD COURSE FORM, ADD STUDENT FORM, ADD STUDENT PERFORMANCE, LANGUAGES Subform), and Reports (STUDENTS RECORDS). The central area shows the "STUDENTS RECORDS" report preview. The report title is "STUDENTS RECORDS". The data is presented in a table with the following columns: SUID, SUemail, FirstName, LastName, Program, Country, Disability, and RO. The data rows are as follows:

SUID	SUemail	FirstName	LastName	Program	Country	Disability	RO
[REDACTED]	@syr.edu	Adam	[REDACTED]	Arts and Sciences - Forensic	USA	False	False
[REDACTED]	@syr.edu	Amy	[REDACTED]	VPA-Music - Recording & AI	USA	False	False
[REDACTED]	@syr.edu	Natalie	[REDACTED]	Public Communications - T	USA	False	False
[REDACTED]	@syr.edu	Zoe	[REDACTED]	Public Communications - N	Australia	False	False
[REDACTED]	@syr.edu	Daniel	[REDACTED]	Public Communications - T	Ecuador	False	False
[REDACTED]	@syr.edu	Nora	[REDACTED]	Arts and Sciences - Arts and	USA	False	True
[REDACTED]	@syr.edu	Kyle	[REDACTED]	Public Communications - A	USA	False	False

At the bottom of the report preview, there are navigation buttons for "Page: 1" and "No Filter". The status bar at the bottom of the screen shows "Page: 1" and "Num Lock".