Project Documentation -Fall 2022

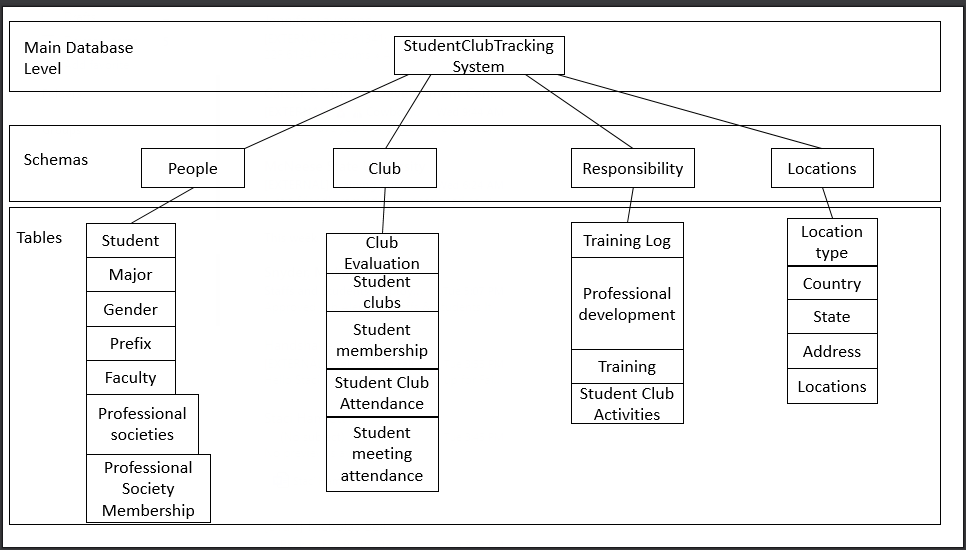
For this document you will be deleting this line, and anything between < >. For each line between < >, you need to write/include the required information in your own words. Add tables, headers, etc as specified. You will be adding to this document as the semester progresses, you do not need to complete all of it at the start.

# Case Study Overview

A nearby school has faced a very strange dilemma. Essentially their professional society called the steam academy has sky rocketed in numbers and attendance. Therefore, gaining funding, resources, and recognition has been increasingly difficult from other competing societies. For the campus to know which clubs and societies require the most resources, they need structured databases to track contributions made by each student in order to add to an annual report to provide to the department supervisors. This leads to more complications- societies and clubs are required to track each students attendance to club events, as well as to needing to coordinate these events in general. They also have many requirements and constraints to be considered. For instance, each student has to be a part of a national professional society and engage in various professional development activities. At these activities, students are required to have active attendance; at least 75% of the events a year. Thus, each of the club members will have earned their keep within the professional society. Successful creation of this tracking system will allow each student society to produce an annual, yet professional, report of all aforementioned contributions, participations, and evaluations of their members.

In order to execute the creation of the tracking system, we have to break down the information we need into their own individual classes. All the details of the clubs should be grouped logically and strategically so that it’s easy to give its yearly evaluation. Therefore, “clubs” could be separated on its own. It could also be reasonably inferred that the system contains people, and all the information that would be associated with them. Because of this inference, “People” could be classified into their own uniquely distinctive groups. Both clubs and professional societies have common requirements/responsibilities that need to be met in order for students to attend training like keeping track of a training log, club activities, etc.. All these obligations are consequently going to be put into a different grouping. As well as needing the information of the people, clubs, and their responsibilities, we’re also going to need to know the location specifics of the events themselves. With these groupings, all of the information would be efficiently ordered in a way so that it can be translated into a database that the clients can easily have access to and thoroughly understand. Not only that, but our tasks as developers will also include for us to formulate relationships between each of the previously established classifications. With the interrelated analogies there will be seamless efficacies for the user to work on their annual reports in its simplest form. With the creation of this database, the lives of the universities will be made easier at keeping track of the student club records.

# Database Flowchart with Schemas and Tables



# ER Diagram



# Tables Flowchart

Diagram

Description automatically generated

# Relationship Descriptions

* The Student Club Membership table will use both the student table and the club table to see whether a student has a membership with a certain club.
* Student Club Activity Attendance table is also a bridge table that uses the tables Student, Clubs, and Activities to identify if a student participated in a specific activity with a specific club.
* The Professional Society Membership uses the Student and Society Table to see if a student is a part of a society or not.
* The Club Evaluations table uses the Activities table and Club table to identify what club will be under evaluations and what activities the club has done.
* The Training Log Table uses the Student and Training table to identify whether a student has taken a training session.

# Table Design

## Student

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraint** |
| StudentId | INTEGER | PRIMARY KEY/IDENTITY |
| FirstName | VARCHAR(50) | REQUIRED |
| MiddleName | VARCHAR(50) | Optional |
| LastName | VARCHAR(50) | REQUIRED |
| BirthDate | DATE | REQUIRED |
| Age | INTEGER | CHECK/ REQUIRED |
| Gender | VARCHAR(30) | ForeignKey |
| EmailAddress | VARCHAR(150) | REQUIRED |
| PhoneNumberMain | VARCHAR(12) | REQUIRED |
| PhoneNumberSecondary | VARCHAR(12) | Optional |
| MailingAddress | VARCHAR(150) | REQUIRED |
| MajorId | INTEGER | Foreign |
| GPA | DECIMAL(3,2) | Check/ REQUIRED |
| GraduationDate | DATE | Optional |
| ActiveMembership | BOOLEAN | REQUIRED |
| TrainingCompleted | BOOLEAN | REQUIRED |

## Major

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraint** |
| MajorId | INTEGER | PRIMARY KEY /IDENTITY |
| MajorName | VARCHAR(50) | Required |

## Student Club Activities

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraint** |
| ActivityId | INTEGER | PRIMARY KEY/ IDENTITY |
| ActivityName | VARHCAR(50) | Unique |
| LocationId | INTEGER | Required |
| ActivityDescription | VARCHAR(250) | Required |

## Prefix

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraint** |
| PrefixId | INTEGER | PRIMARY KEY/ IDENTITY |
| PrefixName | CHAR(3) | REQUIRED |

## Location Type

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraint** |
| LocationTypeId | INTEGER | PRIMARY KEY/ IDENTITY |
| LocationTypeName | VARCHAR(25) | REQUIRED |

## STATE

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraint** |
| StateID | INTEGER | PRIMARY KEY, IDENTITY |
| StateAbbreviation | CHAR(2) | REQUIRED |
| StateName | VARCHAR(50) | REQUIRED |

## STUDENT CLUBS

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraint** |
| ClubId | INTEGER | PRIMARY KEY, IDENTITY |
| ClubName | VARCHAR(50) | UNIQUE, REQUIRED |
| ClubStartDate | DATE | REQUIRED |
| FacId | INTEGER | REQUIRED |
| ClubDesc | VARCHAR(500) | REQUIRED |

## STUDENT CLUB MEMBER

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraint** |
| MemberId | INTEGER | PRIMARY KEY, IDENTITY |
| StudentId | INTEGER | FOREIGN KEY, REQUIRED |
| ClubId | INTEGER | FOREIGN KEY, REQUIRED |
| MemberStart | DATE | REQUIRED |
| MemberEnd | DATE | OPTIONAL |
| MemberActive | BOOLEAN | REQUIRED |
| DutyDescription | VARCHAR(500) | REQUIRED |

## STUDENT CLUB ACTIVITY ATTENDANCE

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraint** |
| ClubEvaluationId | INTEGER | COMPOSITE, REQUIRED |
| StudentId | INTEGER | COMPOSITE FOREIGN KEY |
| ActivityId | INTEGER | COMPOSITE, REQUIRED |
| HoursAttended | INTEGER | REQUIRED |
| DateAttended | DATE | REQUIRED |
| CoordOrNo | BOOLEAN | REQUIRED |
| ContributionDesc | VARCHAR(50) | REQUIRED |

## Faculty

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraint** |
| FacultyId | INTEGER | PRIMARY KEY, IDENTITY |
| Prefix | INTEGER | FOREIGN KEY |
| FirstName | VARCHAR(50) | REQUIRED |
| MiddleName | VARCHAR(50) | OPTIONAL |
| LastName | VARCHAR (50) | REQUIRED |
| EmailAddress | VARCHAR(50) | REQUIRED |
| DepartmentName | VARCHAR (50) | REQUIRED |
| OfficeLocation id | VARHCAR(50) | REQUIRED |
| OfficeRoomNumber | INTEGER | REQUIRED, CHECK between 100 and 499 |
| PhoneNumber | CHAR(14) | REQUIRED |

## Professional Development

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraint** |
| ProfessionalDevelopment | INTEGER | PRIMARY KEY, IDENTITY |
| StudentId | INTEGER | FOREIGN KEY |
| ProfessionalDevelopmentType | VARCHAR(50) | REQUIRED |
| SchoolMandated | BOOL | REQUIRED |
| Description | VARCHAR (500) | REQUIRED |

## Addresses

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraint** |
| AddressId | INTEGER | PRIMARY KEY, IDENTITY |
| Country | INTEGER | FOREIGN KEY |
| State | INTEGER | FOREIGN KEY, CHECK |
| City | VARCHAR(50) | REQUIRED |
| Street | VARCHAR (50) | REQUIRED |
| StreetNumber | INTEGER | REQUIRED |
| ApartmentNumber | INTEGER | Optional |

## Student Meeting attendance

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraint** |
| StudentId | INTEGER | FOREIGN COMPOSITE KEY |
| ClubID | INTEGER | FOREIGN KEY COMPOSITE KEY |
| Date | DATETIME | NOT NULL |

## Professional Societies

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraint** |
| Society ID | INTEGER | PRIMARY KEY/IDENTITY |
| Society Name | VARCHAR (50) | REQUIRED |
| Contact Email | VARCHAR (50) | REQUIRED |
| Contact Phone | VARCHAR (12) | REQUIRED |
| Founding Date | DATE | REQUIRED |
| Brief Description | VARCHAR (500) | REQUIRED |
| Address ID | VARCHAR (50) | REQUIRED |

## Gender

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraint** |
| GenderID | INTEGER | PRIMARY KEY/IDENTITY |
| GenderName | VARCHAR (50) | REQUIRED |

## Locations

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraint** |
| LocationID | INTEGER | PRIMARY KEY/IDENTITY |
| LocationName | VARCHAR (50) | REQUIRED/CHECK if location is virtual |
| LocationAddressID | VARCHAR (50) | FOREIGN KEY |
| LocationMainContact | VARCHAR (50) | FOREIGN KEY |
| LocationType | INTEGER | FOREIGN KEY |
| LocationCapacity | INTEGER | OPTIONAL/CHECK that capacity is between 100 and 1000 |

## Professional Society Member

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraint** |
| StudentID | INTEGER | FOREIGN COMPOSITE KEY |
| SocietyID | VARCHAR (50) | FOREIGN COMPOSITE KEY |
| StartMembershipDate | DATE | REQUIRED |
| EndMembershipDate | DATE | OPTIONAL |

## Country

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraint** |
| CountryID | INTEGER | PRIMARY KEY, IDENTITY |
| CountryAbbreviation | CHAR (2) | REQUIRED |
| CountryName | VARCHAR (50) | REQUIRED |

## Club Evaluations

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraint** |
| ClubID | INTEGER | FOREIGN COMPOSITE KEY |
| ActivityID | INTEGER | FOREIGN COMPOSITE KEY |
| Semester | VARCHAR (50) | REQUIRED |
| Year | CHAR (4) | REQUIRED |
| President | INTEGER | FOREIGN KEY CHECK duplicate |
| Vice President | INTEGER | FOREIGN KEY CHECK duplicate |
| Secretary | INTEGER | FOREIGN KEY CHECK duplicate |
| Treasurer | INTEGER | FOREIGN KEY CHECK duplicate |
| InGoodStanding | BOOLEAN | REQUIRED |

## Training

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraint** |
| TrainingID | INTEGER | PRIMARY KEY, IDENTITY |
| TrainingName | VARCHAR (50) | REQUIRED |
| FacultyAssigner | INTEGER | FOREIGN KEY |
| SemesterOffered | VARCHAR (50) | REQUIRED |

## Training Log

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraint** |
| StudentID | INTEGER | FOREIGN COMPOSITE KEY |
| TrainingID | INTEGER | FOREIGN COMPOSITE KEY |
| TrainingStartDate | DATE | REQUIRED |
| TrainingEndDate | DATE | REQUIRED |

# Appendix A: Responsibility

## Group member 1: Nicholas Bell

|  |  |
| --- | --- |
| **Object worked on** | **Description of Job** |
| Flowchart with tables and schemas | Creating the flowchart hierarchy to show the placement of both schemas and tables |
| Created tables: Student, Major, Student Club Activities, Prefix, and Location Type | Created the tables for the schemas and identified what data types and constraints each one needed. |
| Inserts | Did the inserts for the tables assigned and helped others with their inserts and took screenshots |
| Database and Tables | Helped correct some of the keys and constraints if there was a mistake for the database |
| Flowchart with tables and schemas | After receiving the comments from our last and made the required changes for the next submission |
| Excell Sheet | Created the excel document to import the data into the database |
| Queries | Did part of the select statements of the main tables and worked on the 8 table queries |
| Join Statements | Wrote the code and took the screenshots for the join statements |
| ERD | Contributed to designing/creating the ER Diagram |

## Group Member 2: Nathan guidry

|  |  |
| --- | --- |
| **Object worked on** | **Description of Job** |
| Flowchart with tables and schemas | Organized and use logical thinking to convert it to powerpoint |
| The table relationship chart | Connecting all the table in the case study |
| Created tables: State, Student Clubs, Student club membership, and Student Club Activity Attendance | Created the tables for the schemas and identified what data types and constraints each one needed. |
| Inserts | Did the inserts for the tables assigned and helped others with their inserts and took screenshots |
| Database and Table | Helped correct some of the keys and constraints if there was a mistake for the database |
| Excell Sheet | Created the excel document to import the data into the database |
| Queries | Did part of the select statements of the main tables and worked on the 8 table queries |
| Join Statements | Wrote the code and took the screenshots for the join statements |
| ERD | Contributed to designing/creating the ER Diagram |

## Group Member 3: staci hamlett

|  |  |
| --- | --- |
| **Object worked on** | **Description of Job** |
| The table relationship chart | Connecting all the table in the case study |
| Flowchart with tables and schemas | Organized and use logical thinking to convert it to powerpoint |
| Created tables: Faculty, Addresses, Professional Development, and Student meeting attendance | Created the tables for the schemas and identified what data types and constraints each one needed. |
| Inserts | Did the inserts for the tables assigned and helped others with their inserts and took screenshots |
| Master doc editing | Took the comments from last submission and corrected them for the next one |
| Excell Sheet | Created the excel document to import the data into the database |
| Queries | Did part of the select statements of the main tables and worked on the 8 table queries |
| Join Statements | Wrote the code and took the screenshots for the join statements |

## Group Member 4: Jayden lacombe

|  |  |
| --- | --- |
| **Object worked on** | **Description of Job** |
| Making the backup file accessible for everyone to use | Putting the database into GitHub, so the other members can access the database in an updated form |
| Flowchart with tables and schemas | Organized and use logical thinking to convert it to powerpoint |
| Created tables: Professional Societies, Gender, Locations, and Professional society Membership | Created the tables for the schemas and identified what data types and constraints each one needed. |
| Inserts | Coded the inserts for his part of the project |
| Excell Sheet | Created the excel document to import the data into the database |
| Queries | Did part of the select statements of the main tables and worked on the 8 table queries |
| Join Statements | Wrote the code and took the screenshots for the join statements |

## GROUP MEMBER 5: BRENDEN LEEDOM

|  |  |
| --- | --- |
| **Object worked on** | **Description of Job** |
| Flowchart with tables and schemas | Creating the flowchart hierarchy to show the placement of both schemas and tables |
| Created tables: Country, Club Evaluation, Training, and Training Log | Created the tables for the schemas and identified what data types and constraints each one needed. |
| Database and Tables | Helped correct some of the keys and constraints if there was a mistake for the database |
| Inserts | Did the inserts for the tables assigned and helped others with their inserts |
| Excell Sheet | Created the excel document to import the data into the database |
| Queries | Did part of the select statements of the main tables and worked on the 8 table queries |
| Join Statements | Wrote the code and took the screenshots for the join statements |
| ERD | Contributed to designing/creating the ER Diagram |

# Appendix B: Code Screenshots

## Database Creation Code

Graphical user interface, text, application, Word

Description automatically generated

## Database Schemas Creation Code

Graphical user interface, text, application

Description automatically generated

## iNSERT sCREENSHOTS

Locations

Graphical user interface, text, application, Word

Description automatically generated

Professional Societies Memborship

Text

Description automatically generated

Student Meeting Attendance:

Graphical user interface, text

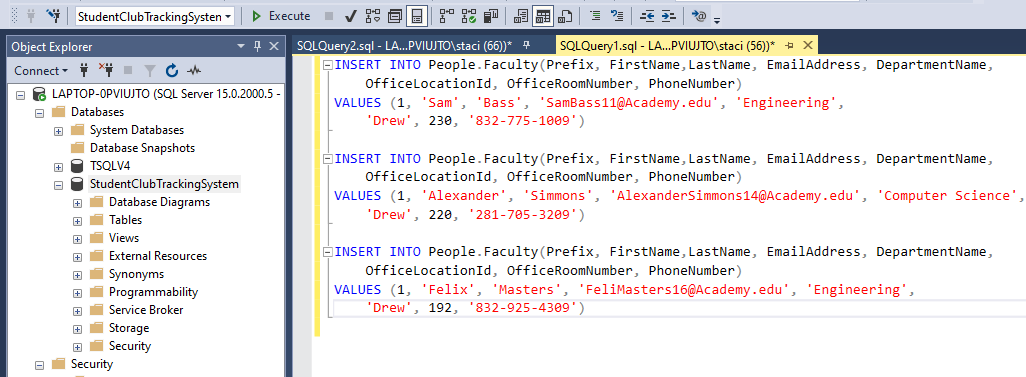
Description automatically generated

Professional Development:

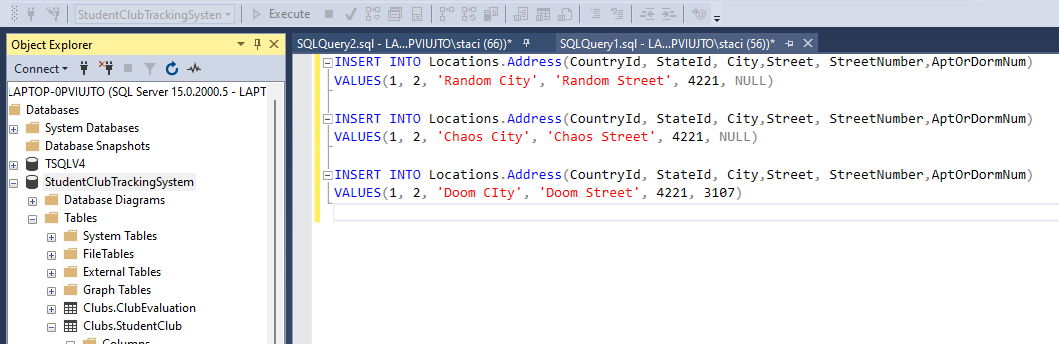
Graphical user interface, text, application

Description automatically generated

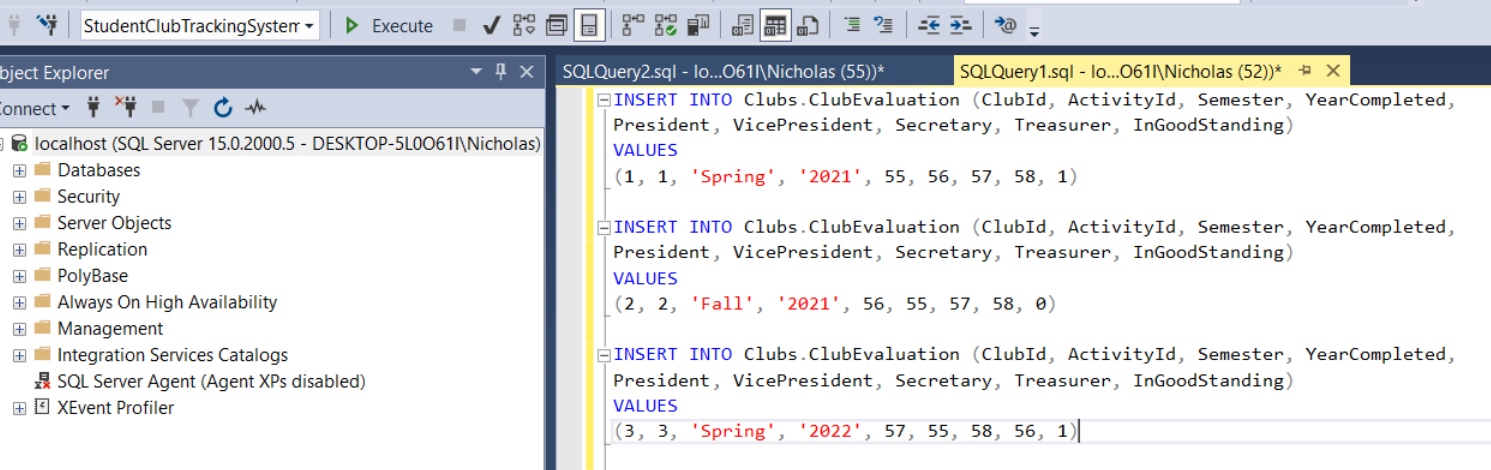
Faculty:



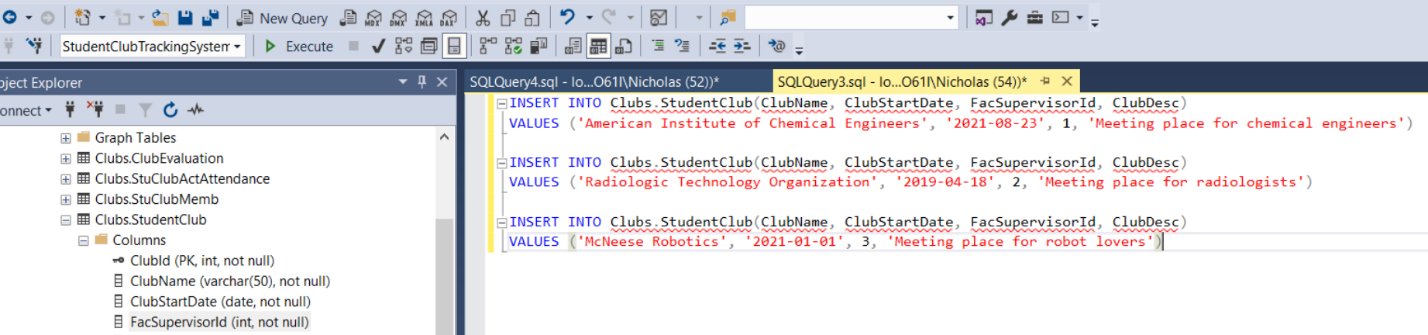
Address:



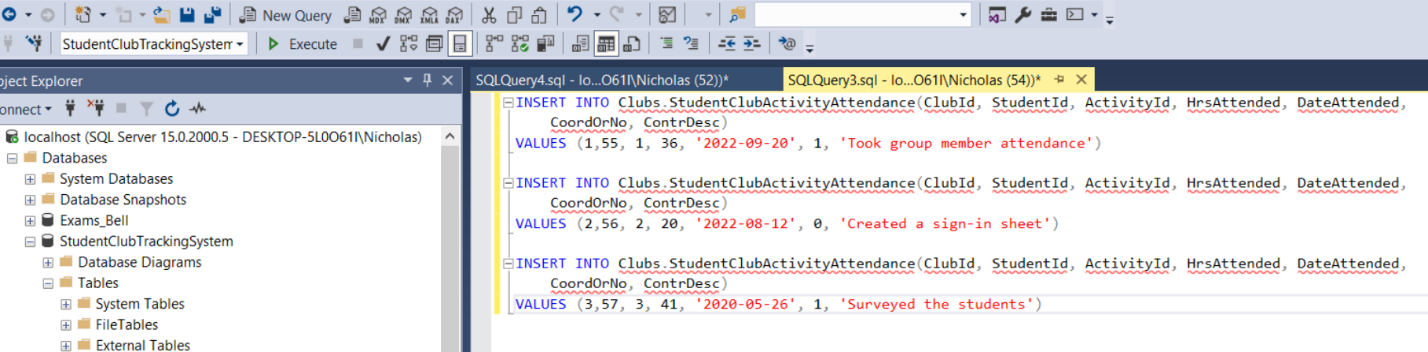
Club Evaluations:



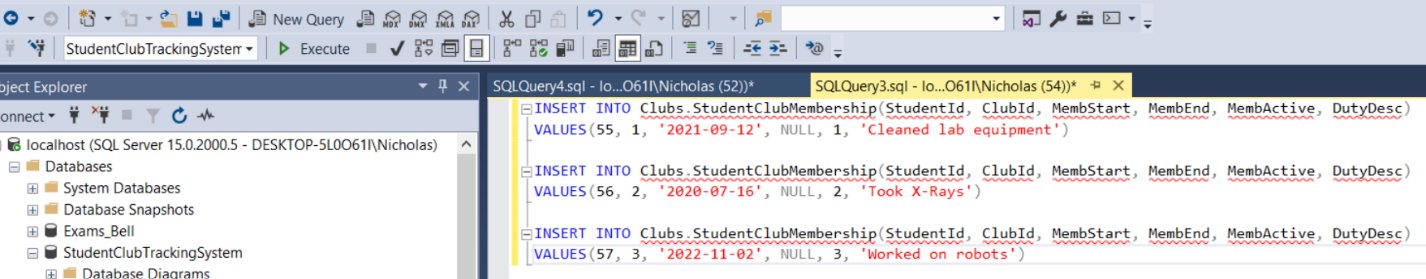
Student Clubs:



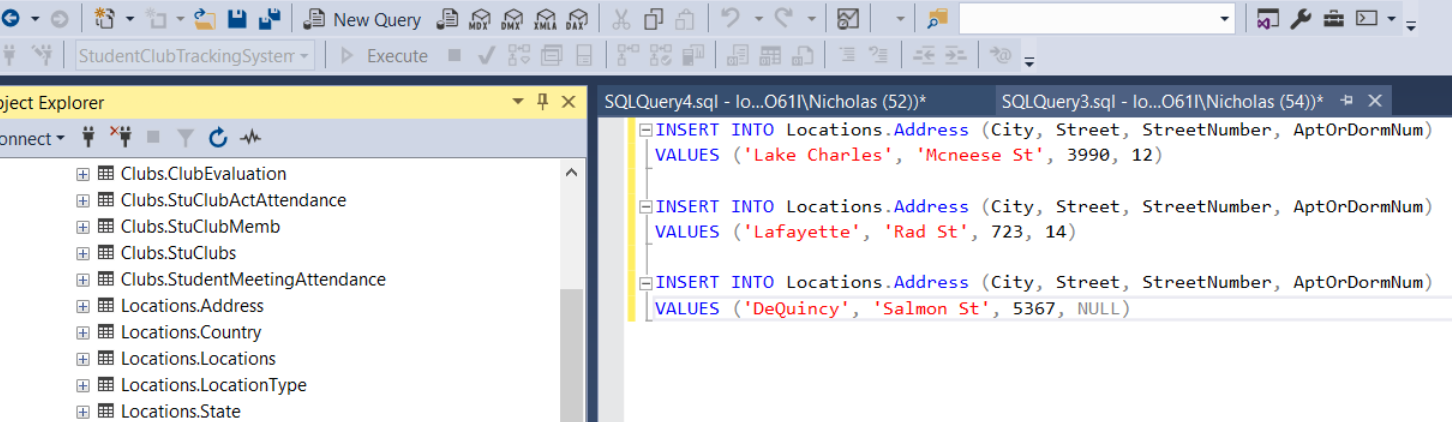
Student Club Activity:



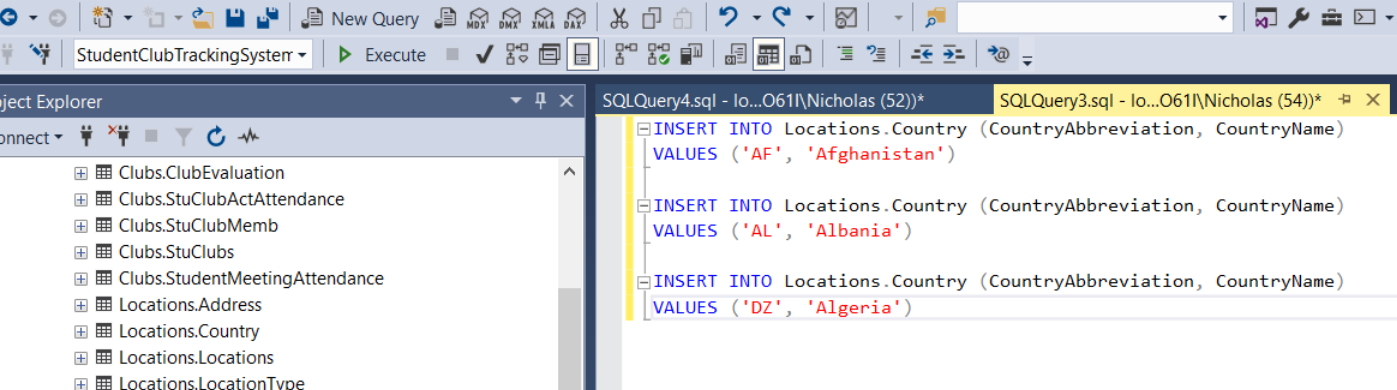
Student Club Membership:



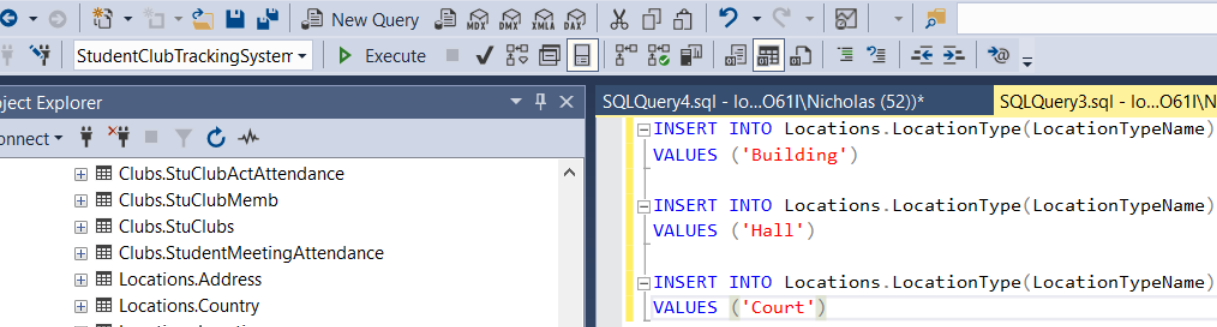
Address:



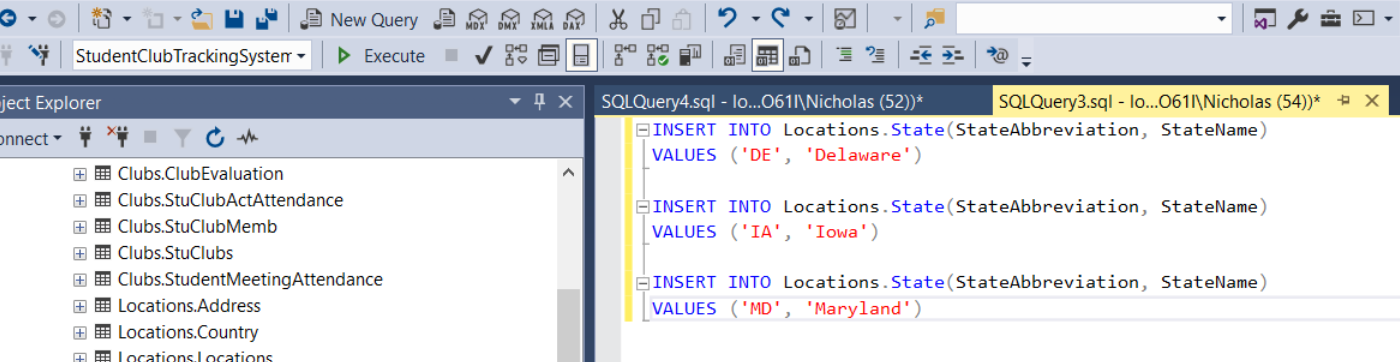
Country:



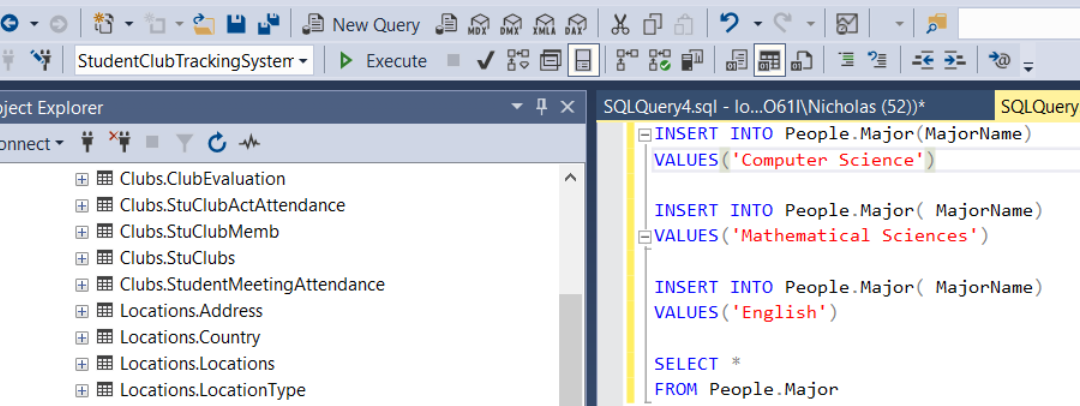
Location Type:



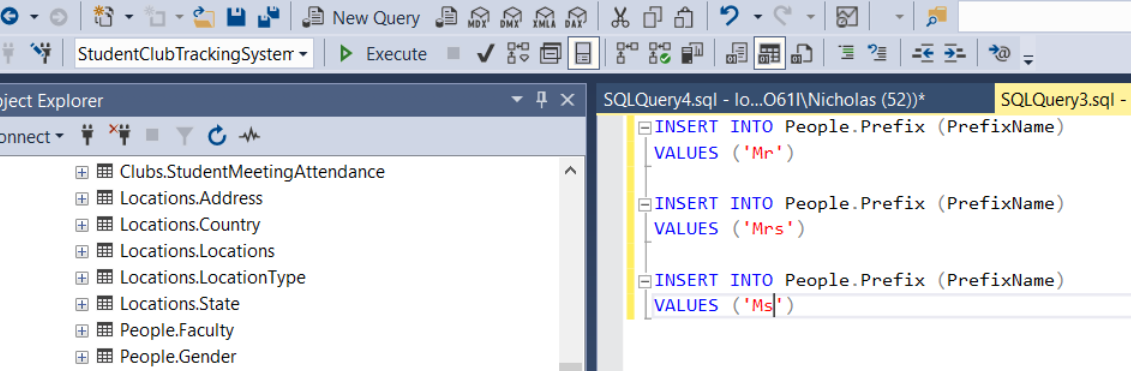
State:



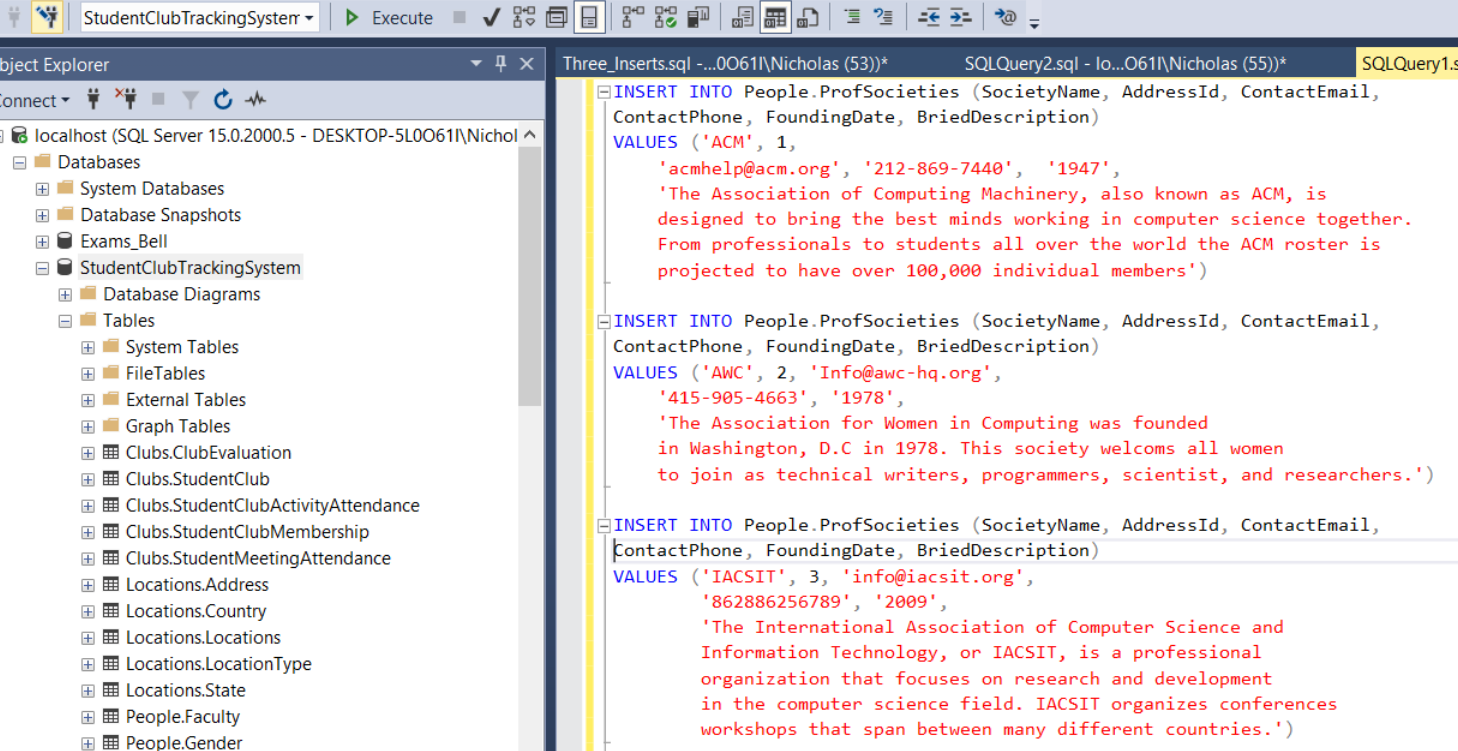
Major:



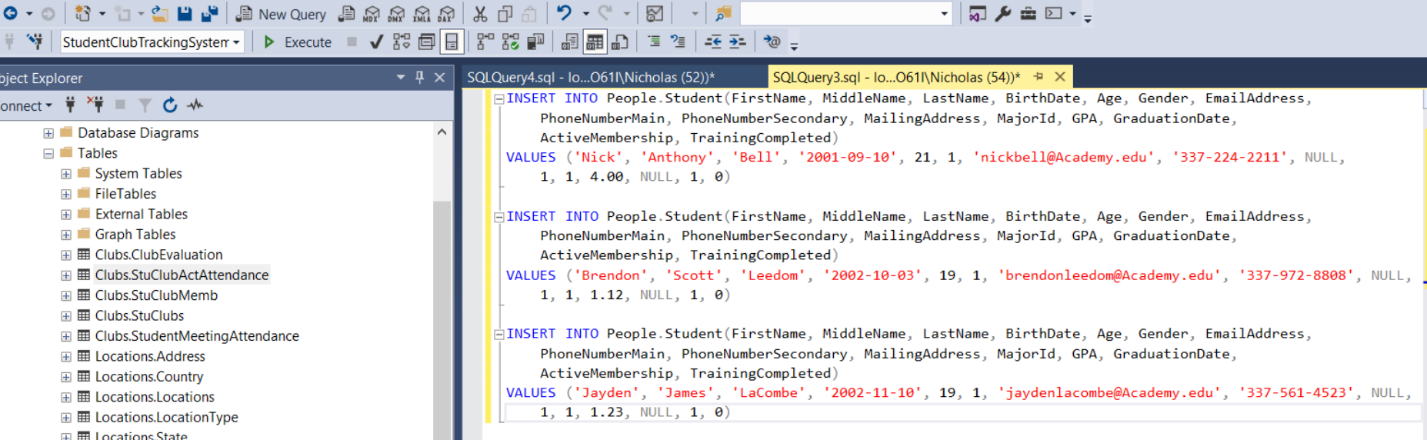
Prefix



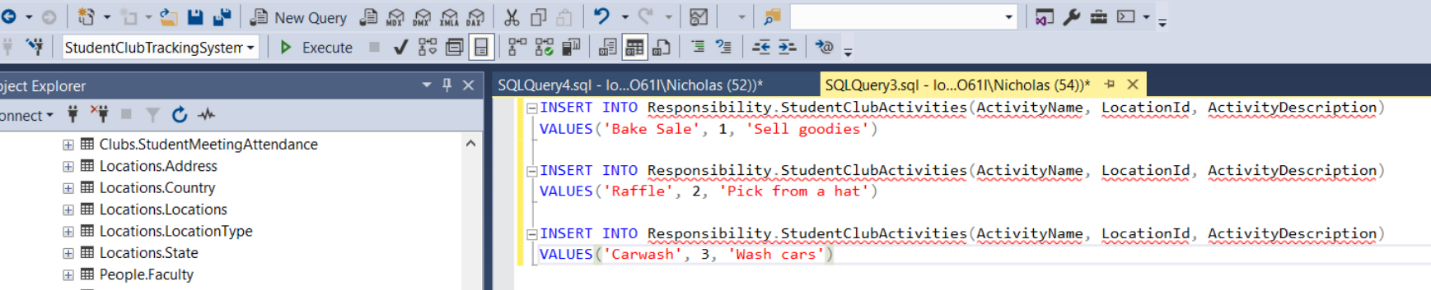
Professional Societies



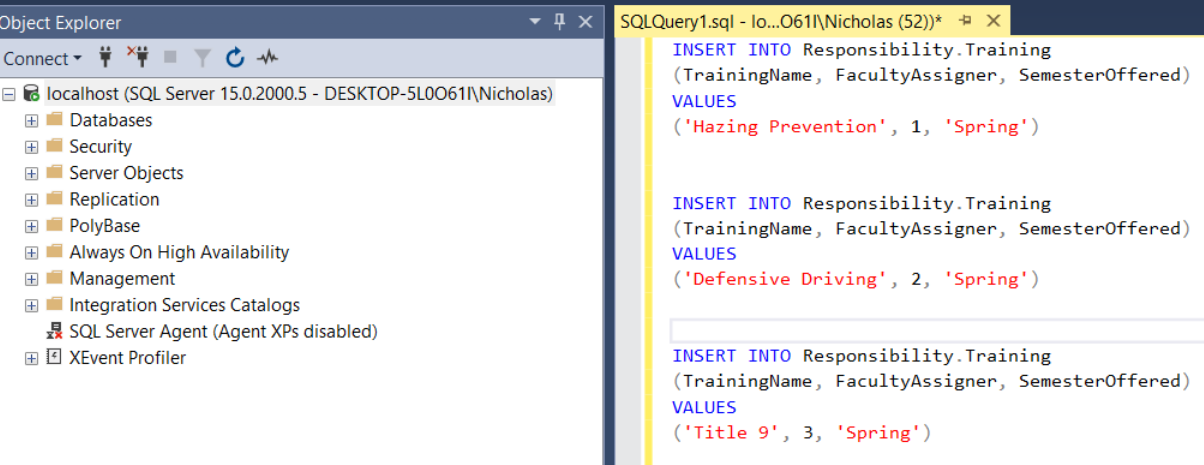
Student:



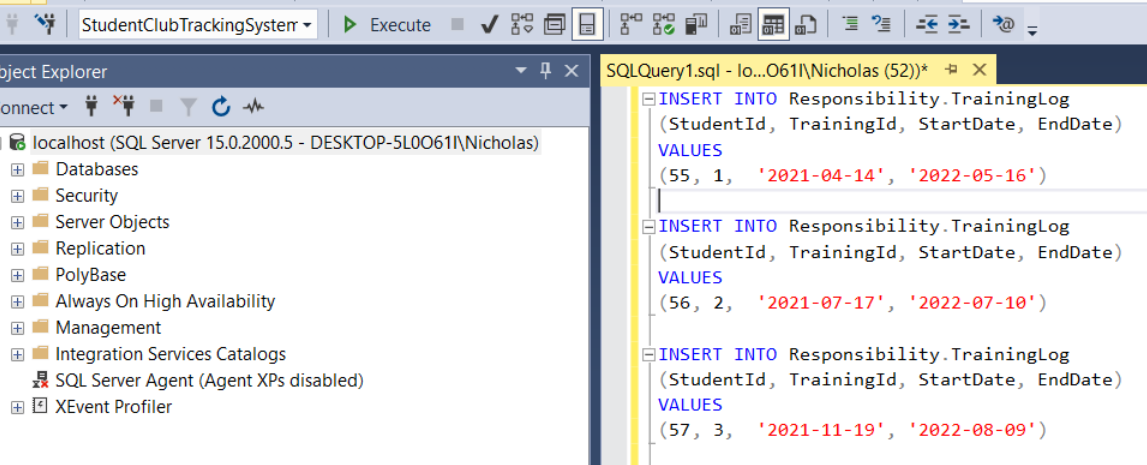
Student Club Activities:



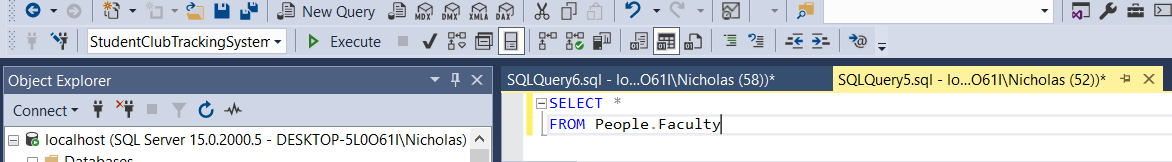
Training

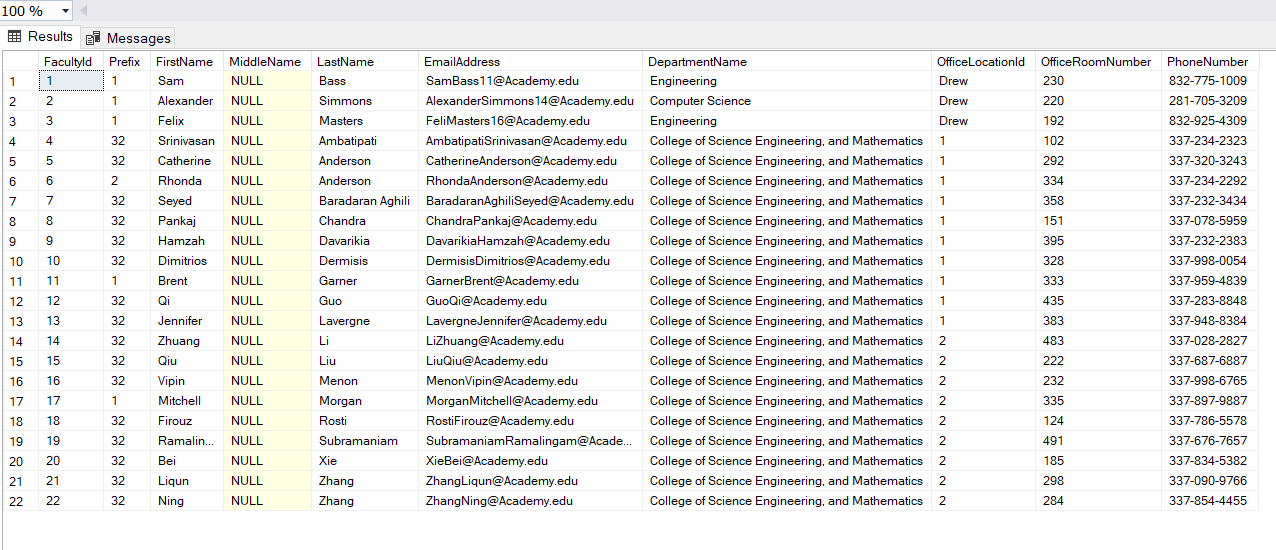


Training Log:



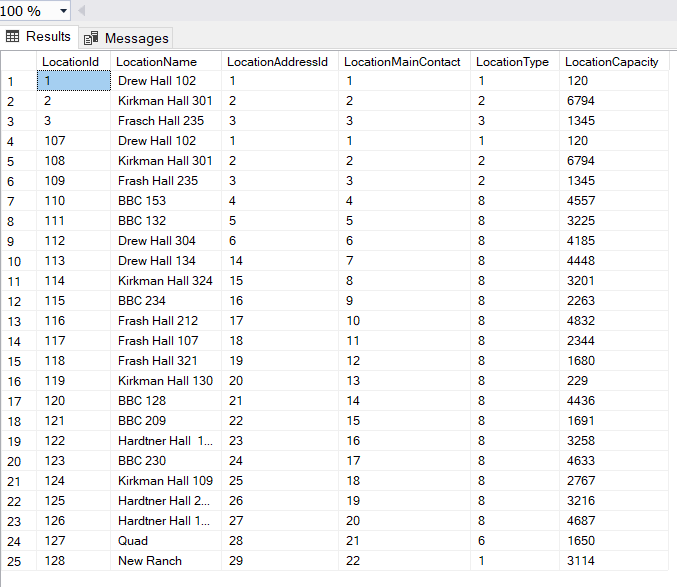
## Select Statements

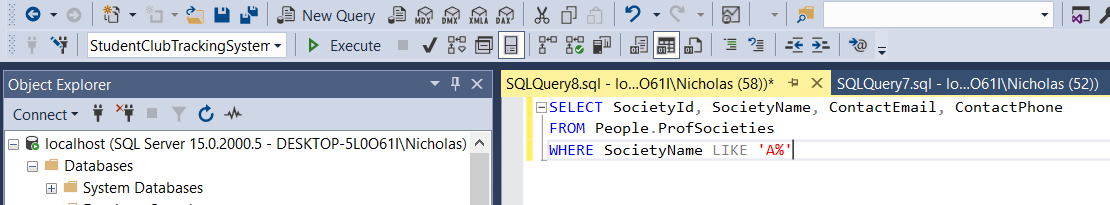
Returns Everything



Graphical user interface, text, application, Word

Description automatically generated



2 Tables that return 4 columns filtered on 1 criterion:



Graphical user interface, application

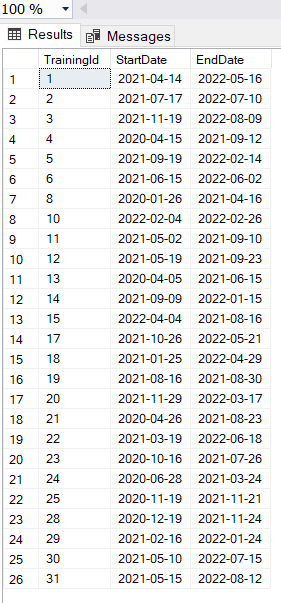
Description automatically generated



2 Tables that return 3 columns filtered on dates:

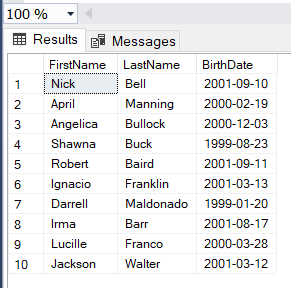
Graphical user interface, text, application

Description automatically generated



Graphical user interface, text, application, email

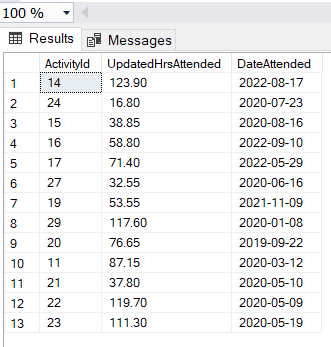
Description automatically generated



1 table that returns 3 columns filtered on 2 criterions:

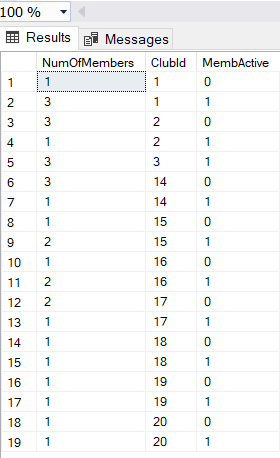
A picture containing diagram

Description automatically generated

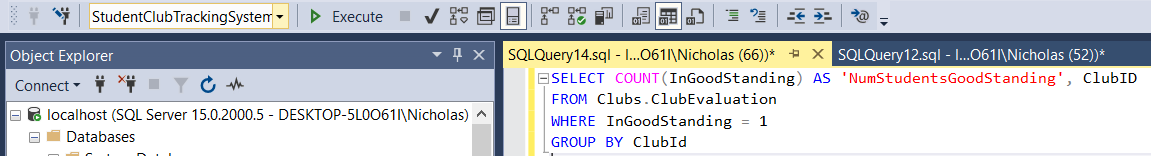


1 table that returns 3 columns and performs group by and order by: Graphical user interface, application

Description automatically generated



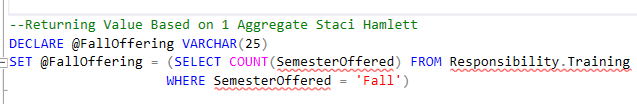
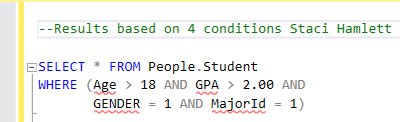
1 Table that returns 2 columns that computes an aggregate:

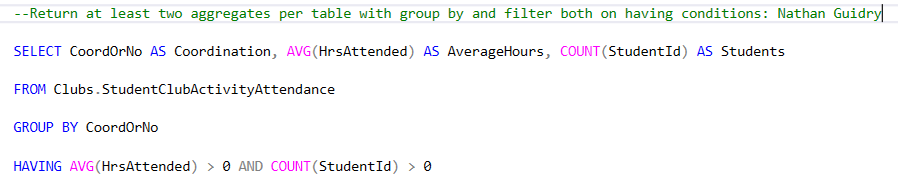
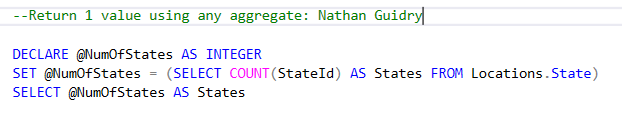


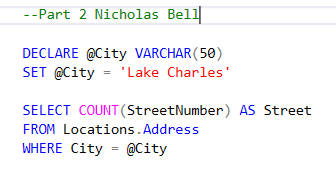
Table

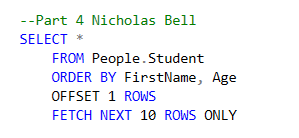
Description automatically generated

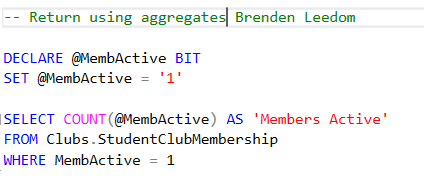
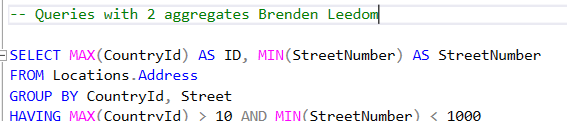
## SELECT QUERIES











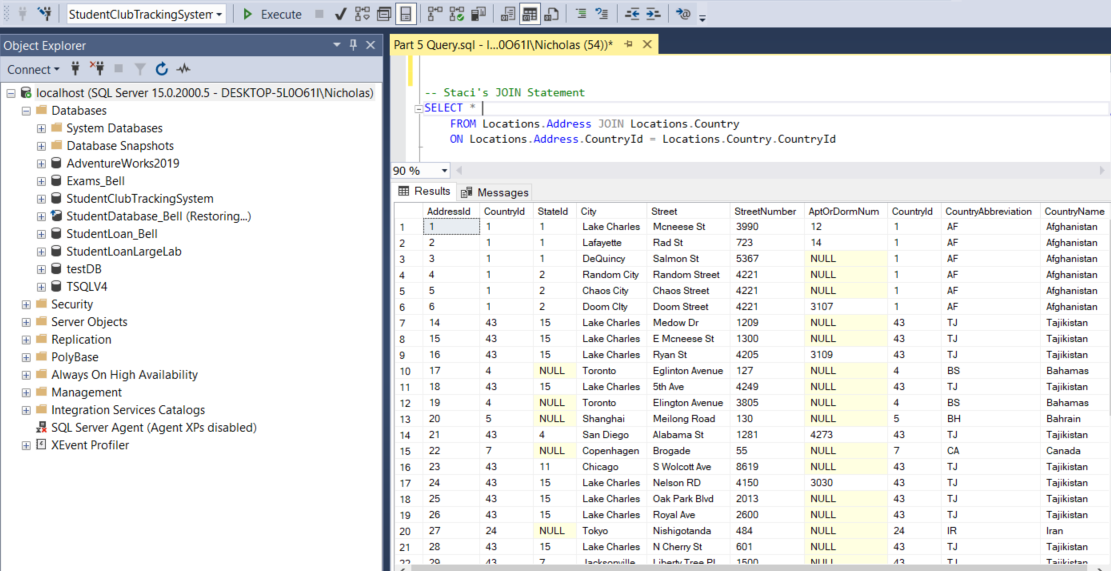
Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

# Join Statement



Jayden’s Join Statement

Word

Description automatically generated with low confidence

`Table

Description automatically generated