Syracuse University iSchool

Course Catalog Database Report

IST-659

Yibo Feng

Contents

[Project Summary 3](#_Toc39308161)

[Entity and Attribute Table 4](#_Toc39308162)

[Entity Relationship Diagram 8](#_Toc39308163)

[Creation of Tables 9](#_Toc39308164)

[Major Data Questions 18](#_Toc39308165)

[Interfaces 22](#_Toc39308166)

[Report 22](#_Toc39308167)

# Project Summary

**This project focuses on designing a database for Syracuse University iSchool course catalog** and course registration system. The iSchool is offering a course catalog which entails available classes of different levels pertinent to various information areas. Currently, t**he course catalog can be accessed through 2 separate routes which simply display a list of available courses and their basic information (course code, instructor, overview, and pre-requisite course).** Apart from the course catalog, the course registration system is integrated with the class searching system called ‘Myslice’ where users can check their current credit records and view available courses. In conclusion, the current course catalog and course registration system are separate, and they are not capable of providing one single platform for comprehensive data sources, resulting in user inconveniences.

Therefore, **the purpose of this system development is to establish an integrated iSchool course catalog database and link it to the course registration system for enhancing user conveniences.** For instance, students are expected to **shorten their time for browsing** and selecting courses they are interested in or required to take and make better course registration quality-wise. Also, administrators can analyze such data more conveniently and make use of the analysis for improving iSchool academic course design. In the long run, it will contribute to making iSchool courses more attractive as well as elevating students’ competences.

For this, we need to build an integrated single database which involves holistic data about all iSchool courses. The database will be complemented with detailed course information both in academic and professional aspects such as technical tools, prerequisite courses, related academic field, career field and so on. Based on the database, functions will be diversified targeting different user groups. Students will be able to view their profile as well as credit status and directly register courses which align with their academic and career goals. The system will also help instructors notify course updates and classroom information in real time. Companies could make use of this system to find potential employees by querying the database and administrators could be able to control the whole system more efficiently.

This summary states overall system design, specific purposes and functions, entities and attributes, dependencies and relationships between the various entities and attributes (ERD), business rules, and major data questions related to major issues in the system management.

For this project, we will use following tools: Access, SQL Server, Visio.

# 

# Entity and Attribute Table

1. Class: This entity captures information about the classes that ischool offered and all the information toward each specific class

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Attr type | NOT NULL/NULL | PK | Other constraints | Description |
| ClassID | VARCHAR(20) | NOT NULL | PK |  | Primary key identifier of the class, can be defined as courseID |
| ClassRoomID | VARCHAR(20) | NOT NULL | FK |  | Identify the location of the class |
| InstID | VARCHAR(40) | NOT NULL | FK |  | Identify the Instructor |
| Credits | VARCHAR(10) | NOT NULL |  |  | Credits for the class |
| Semester | VARCHAR(20) | NOT NULL |  |  | Define which semester it is |
| StartDate | DATETIME | NOT NULL |  |  | The class start date |
| EndDate | DATETIME | NOT NULL |  |  | The class end date |
| Duration | VARCHAR(20) | NOT NULL |  |  | Define the duration of class |
| ClassSize | VARCHAR(20) | NOT NULL |  |  | The size of the class |
| WKSch | VARCHAR(20) | NOT NULL |  | (MTWTHFSAS) | Class days of each week (MTWTHF) |
| CourseID | VARCHAR(20) | NOT NULL | FK |  | Define the course type that the class belongs to |
| Enrollment | INTEGER |  |  |  | The actual number of students |

1. Course: This entity captures information about the courses that ischool offered, the table covered the details about each course and it’s descriptions and requirements

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Attr type | NOT NULL/NULL | PK | Other constraints | Description |
| CourseID | VARCHAR(20) | NOT NULL | PK |  | Primary key identifier for the specific subject |
| SubjectCode | VARCHAR(20) | NOT NULL |  | GET/IDS/IST/MBC | Global Enterprise Technology(GET)/Information Technology, Design and Startup(IDS)/Information Studies(IST)/(MBC) |
| TTID | INTEGER |  | FK |  | Define which technical tool is going to be used in this subject |
| CourseName | VARCHAR(50) | NOT NULL |  |  | Name of the subject |
| PreRequestCourse1 | VARCHAR(20) |  |  |  | The combination of the Pre required courses. |
| PreRequestCourse2 | VARCHAR(20) |  |  |  | Text description of the pre request courses. |

1. ClassRoom: This entity captures information about the class room which contain the geographic information for each class room

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Attr type | NOT NULL/NULL | PK | Other constraints | Description |
| ClassRoomID | VARCHAR(50) | NOT NULL | PK |  | Define the place where to take class( including TBA and Online) |
| Capacity | VARCHAR(20) | NOT NULL |  |  | The classroom capacity |
| ClassRoomNumber | VARCHAR(20) | NOT NULL |  |  | The classroom number associated with the building |
| ClassBuilding | VARCHAR(80) | NOT NULL |  |  | The building of classroom |

1. TechTool: This entity captures information about the technical tools that ischool’s course covered

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Attr type | NOT NULL/NULL | PK | Other constraints | Description |
| TTID PK | INTEGER | NOT NULL | PK |  | Primary key for the technical tool which is going to be used |
| TTName | VARCHAR(20) | NOT NULL |  |  | The name of that technical tool |
| TTDescribe | VARCHAR(50) | NOT NULL |  |  | The description of that technical tool |

1. Instructor: This entity captures information about the instructor who teach ischool courses and its details

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Atrribute name | Atrr type | NOT NULL/NULL | PK | Other constraints | Description |
| InstID | VARCHAR(40) | NOT NULL | PK |  | Primary identifier for the instructor |
| InstFName | VARCHAR(40) | NOT NULL |  |  | First name of instructor |
| InstLName | VARCHAR(40) | NOT NULL |  |  | Last name of instructor |
| InstEmail | VARCHAR(50) | NOT NULL |  | Unique | Instructor’s email |
| InstGender | VARCHAR(20) | NOT NULL |  | Male orFemale | The gender of the instructor |
| InstTitle | VARCHAR(20) | NOT NULL |  |  | The position title of the instructor |

1. Student: This entity captures information about the students and the targeting job position associated with it

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Atrribute name | Atrr type | NOT NULL/NULL | PK | Other constraints | Description |
| SUID PK | VARCHAR(20) | NOT NULL | PK |  | Primary identifier for students |
| StudFName | VARCHAR(20) | NOT NULL |  |  | First name of students |
| StudLName | VARCHAR(20) | NOT NULL |  |  | Last name of students |
| StudEmail | VARCHAR(50) | NOT NULL |  | Unique | Student’s email |
| StudGender | VARCHAR(20) | NOT NULL |  |  | The gender of the student |
| StudPhoneNum | VARCHAR(20) | NOT NULL |  | Unique | Phone number of students |
| StudMajor | VARCHAR(20) | NOT NULL |  |  | The program that the student enrolls in |
| StudLevel | VARCHAR(20) | NOT NULL |  |  | Whether a student is graduate or undergraduate |
| TargetJob | VARCHAR(20) |  |  |  | The kind of job that the student is most like to pursue |

1. Registration: This entity captures information about the registration that students who signed up for the classes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Attr type | NOT NULL/NULL | PK | Other constraints | Description |
| RegstID | VARCHAR(20) | NOT NULL | PK |  | Primary identifier for the registration |
| RegstDate | DATETIME | NOT NULL |  |  | The date on which the student enrolls the course |
| SUID | VARCHAR(20) | NOT NULL | FK |  | Define the student who take the classClasPK |
| ClassID | VARCHAR(20) | NOT NULL | FK |  | Define the class which the student takes |

1. Company: This entity captures information about the companies who are seeking for employees searching

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Attr type | NOT NULL/NULL | PK | Other constraints | Description |
| CompID | INTEGER | NOT NULL | PK |  | Primary identifier for the company |
| CompState | VARCHAR(20) | NOT NULL |  |  | The state of the company |
| CompCity | VARCHAR(20) | NOT NULL |  |  | The city of the company |
| CompField | TEXT |  |  |  | The field of the company |
| CompValue | TEXT |  |  |  | The description of the value of the company |
| CompName | VARCHAR(20) | NOT NULL |  |  | The Name of Company |

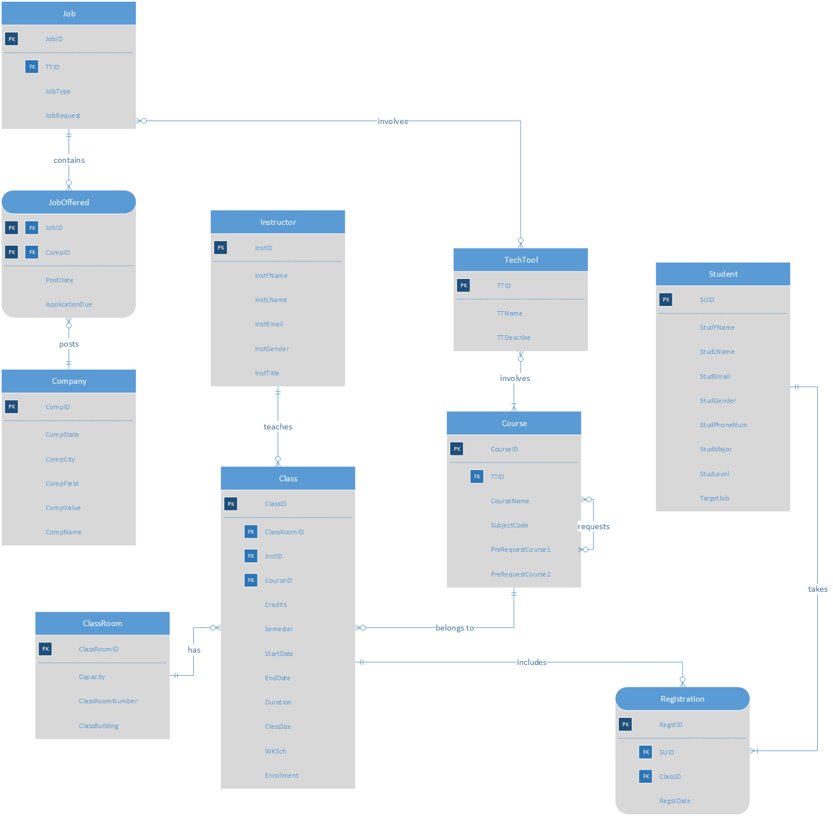
1. Job: This entity captures information about the jobs that ischool students might applying

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Attr type | NOT NULL/NULL | PK | Other constraints | Description |
| JobID | INTEGER | NOT NULL | PK |  | Primary identifier for the job |
| TTID | INTEGER | NOT NULL | FK |  | Define the tech tool that the job needs |
| JobType | VARCHAR(50) | NOT NULL |  |  | The type of job such as data, consulting. |
| JobRequest | TEXT | NOTNULL |  |  | Whether the job is an internship or a part-time job or a full-time job. |

1. JobOffered: This entity captures information about the job details that hiring companies offered

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attr type | NOT NULL/NULL | PK | Other constraints | Description |
| INTEGER | NOT NULL | PK |  | Define the Company that posts the job |
| INTEGER | NOT NULL | PK |  | Define the job that the company posts |
| DATETIME | NOTNULL |  |  | The date of the job posted |
| DATETIME |  |  |  | The due of the job application |

# Entity Relationship Diagram



# Creation of Tables

--- Create ClassRoom ---

CREATE TABLE ClassRoom(

ClassRoomID VARCHAR(50) NOT NULL PRIMARY KEY,

Capacity VARCHAR(20) NOT NULL,

ClassRoomNumber VARCHAR(20) NOT NULL,

ClassBuilding VARCHAR(80) NOT NULL,

);

--- Create TechTool ---

CREATE TABLE TechTool(

TTID INTEGER NOT NULL PRIMARY KEY,

TTName VARCHAR(20) NOT NULL,

TTDescribe VARCHAR(50) NOT NULL,

);

--- Create Instructor ---

CREATE TABLE Instructor(

InstID VARCHAR(40) NOT NULL PRIMARY KEY,

InstFName VARCHAR(40) NOT NULL,

InstLName VARCHAR(40) NOT NULL,

InstEmail VARCHAR(50) NOT NULL UNIQUE,

InstGender VARCHAR(20) NOT NULL,

InstTitle VARCHAR(40) NOT NULL,

CONSTRAINT chk\_InstGendr CHECK (InstGender = 'Male' OR InstGender = 'Female'),

);

--- Create Student ---

CREATE TABLE Student(

SUID VARCHAR(20) NOT NULL PRIMARY KEY,

StudFName VARCHAR(20) NOT NULL,

StudLName VARCHAR(20) NOT NULL,

StudEmail VARCHAR(50) NOT NULL UNIQUE,

StudGender VARCHAR(20) NOT NULL,

StudPhoneNum VARCHAR(20) NOT NULL UNIQUE,

StudMajor VARCHAR(20) NOT NULL,

StudLevel VARCHAR(20) NOT NULL,

TargetJob VARCHAR(20),

CONSTRAINT chk\_StudLevel CHECK (StudLevel = 'Undergraduate' OR StudLevel = 'Graduate'),

);

--- Create Course ---

CREATE TABLE Course(

CourseID VARCHAR(20) NOT NULL PRIMARY KEY,

SubjectCode VARCHAR(20) NOT NULL,

TTID INTEGER FOREIGN KEY REFERENCES TechTool(TTID),

CourseName VARCHAR(50) NOT NULL,

PreRequestCourse1 VARCHAR(20),

PreRequestCourse2 VARCHAR(20),

);

--- Create Class ---

CREATE TABLE Class(

ClassID VARCHAR(20) NOT NULL PRIMARY KEY,

ClassRoomID VARCHAR(50) NOT NULL FOREIGN KEY REFERENCES ClassRoom(ClassRoomID),

InstID VARCHAR(40) NOT NULL FOREIGN KEY REFERENCES Instructor(InstID),

Credits VARCHAR(10) NOT NULL,

Semester VARCHAR(20) NOT NULL,

StartDate DATETIME NOT NULL,

EndDate DATETIME NOT NULL,

Duration VARCHAR(20) NOT NULL,

ClassSize VARCHAR(20) NOT NULL,

WkSch VARCHAR(20) NOT NULL,

CourseID VARCHAR(20) NOT NULL FOREIGN KEY REFERENCES Course(CourseID),

CONSTRAINT chk\_WkSch CHECK (WkSch='M' OR WkSch='T' OR WkSch='W' OR WkSch='TH' OR WkSch='F' OR WkSch ='SA' OR WkSch='S'),

Enrollment INTEGER,

);

--- Create Registration ---

CREATE TABLE Registration(

RegstID VARCHAR(20) NOT NULL PRIMARY KEY,

RegstDate DATETIME NOT NULL,

SUID VARCHAR(20) FOREIGN KEY REFERENCES Student(SUID),

ClassID VARCHAR(20) FOREIGN KEY REFERENCES Class(ClassID),

);

--- Create Company ---

CREATE TABLE Company(

CompanyId INTEGER NOT NULL PRIMARY KEY,

CompState VARCHAR(20) NOT NULL,

CompCity VARCHAR(20) NOT NULL,

CompField TEXT,

CompValue TEXT,

CompName VARCHAR(20) NOT NULL,

);

--- Create Job ---

CREATE TABLE Job(

JobID INTEGER NOT NULL PRIMARY KEY,

TTID INTEGER NOT NULL FOREIGN KEY REFERENCES TechTool(TTID),

JobType VARCHAR(50) NOT NULL,

JobRequest TEXT NOT NULL,

);

--- Create JobOffered ---

CREATE TABLE JobOffered(

CompID INTEGER NOT NULL REFERENCES Company(CompanyID),

JobID INTEGER NOT NULL REFERENCES Job(JobID),

PostDate DATETIME NOT NULL,

ApplicationDue DATETIME,

PRIMARY KEY(CompID,JobID)

);

Insertion of Tables

---Insert data - ClassRoom table---

INSERT INTO ClassRoom (ClassRoomID,Capacity,ClassRoomNumber,ClassBuilding)

VALUES('H002','40','2','Hinds Hall')

INSERT INTO ClassRoom (ClassRoomID,Capacity,ClassRoomNumber,ClassBuilding)

VALUES('H008','40','8','Hinds Hall')

INSERT INTO ClassRoom (ClassRoomID,Capacity,ClassRoomNumber,ClassBuilding)

VALUES('H010','40','10','Hinds Hall')

INSERT INTO ClassRoom (ClassRoomID,Capacity,ClassRoomNumber,ClassBuilding)

VALUES('H033','40','33','Hinds Hall')

INSERT INTO ClassRoom (ClassRoomID,Capacity,ClassRoomNumber,ClassBuilding)

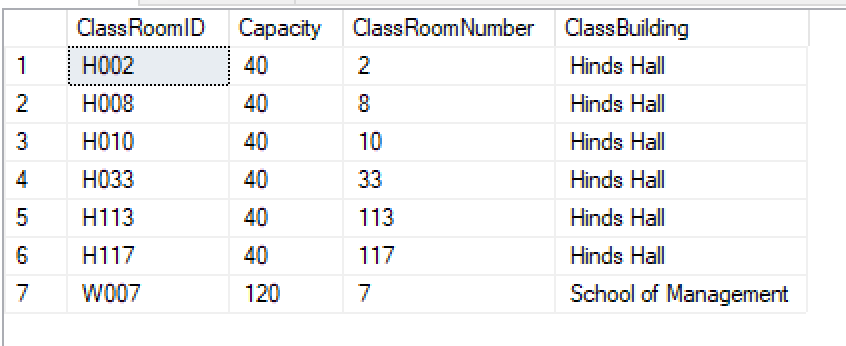
VALUES('H113','40','113','Hinds Hall')

INSERT INTO ClassRoom (ClassRoomID,Capacity,ClassRoomNumber,ClassBuilding)

VALUES('H117','40','117','Hinds Hall')

INSERT INTO ClassRoom (ClassRoomID,Capacity,ClassRoomNumber,ClassBuilding)

VALUES('W007','120','7','School of Management');



---Insert data - TechTool table---

INSERT INTO TechTool(TTID,TTName,TTDescribe)

VALUES(1,'Excel','Spreadsheet developed by Microsoft')

INSERT INTO TechTool(TTID,TTName,TTDescribe)

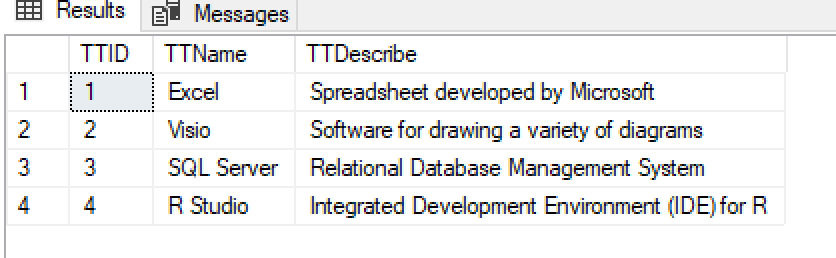
VALUES(2,'Visio','Software for drawing a variety of diagrams')

INSERT INTO TechTool(TTID,TTName,TTDescribe)

VALUES(3,'SQL Server','Relational Database Management System')

INSERT INTO TechTool(TTID,TTName,TTDescribe)

VALUES(4,'R Studio','Integrated Development Environment (IDE) for R');



---Insert data - Instructor table---

INSERT INTO Instructor(InstID,InstFName,InstLName,InstEmail,InstGender,InstTitle)

[VALUES('1','Hernando','Hoyos','hhoyos@syr.edu','Male','Professor](mailto:VALUES('1','Hernando','Hoyos','hhoyos@syr.edu','Male','Professor)')

INSERT INTO Instructor(InstID,InstFName,InstLName,InstEmail,InstGender,InstTitle)

[VALUES('2','Jilian','Lando','jlando@syr.edu','Female','Professor](mailto:VALUES('2','Jilian','Lando','jlando@syr.edu','Female','Professor)')

INSERT INTO Instructor(InstID,InstFName,InstLName,InstEmail,InstGender,InstTitle)

[VALUES('3','Anna','Chernobai','achernobai@syr.edu','Female','Professor](mailto:VALUES('3','Anna','Chernobai','achernobai@syr.edu','Female','Professor)')

INSERT INTO Instructor(InstID,InstFName,InstLName,InstEmail,InstGender,InstTitle)

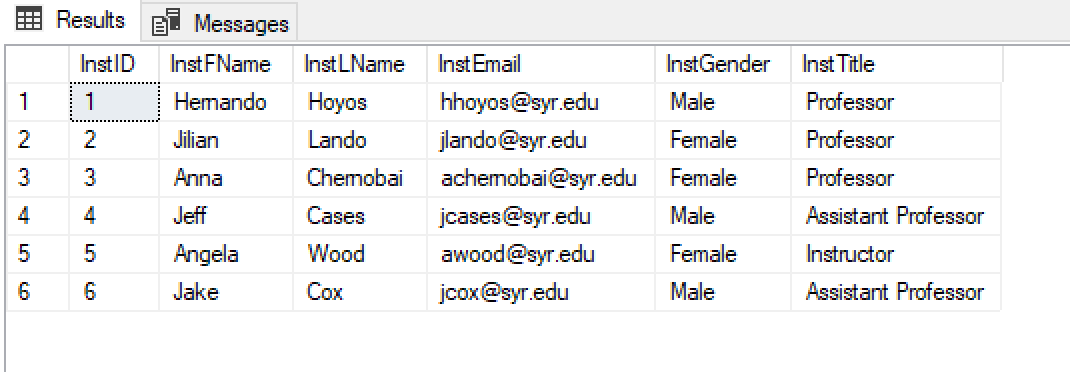
[VALUES('4','Jeff','Cases','jcases@syr.edu','Male','Assistant](mailto:VALUES('4','Jeff','Cases','jcases@syr.edu','Male','Assistant) Professor')

INSERT INTO Instructor(InstID,InstFName,InstLName,InstEmail,InstGender,InstTitle)

[VALUES('5','Angela','Wood','awood@syr.edu','Female','Instructor](mailto:VALUES('5','Angela','Wood','awood@syr.edu','Female','Instructor)')

INSERT INTO Instructor(InstID,InstFName,InstLName,InstEmail,InstGender,InstTitle)

[VALUES('6','Jake','Cox','jcox@syr.edu','Male','Assistant](mailto:VALUES('6','Jake','Cox','jcox@syr.edu','Male','Assistant) Professor');



---Insert data - Student table---

INSERT INTO Student(SUID,StudFName,StudLName,StudEmail, StudGender,StudPhoneNum,StudMajor,StudLevel,TargetJob)

[VALUES('S01','James','Smith','jsmith@syr.edu','Male','315-772-1221','IM','Undergraduate','Data](mailto:VALUES('S01','James','Smith','jsmith@syr.edu','Male','315-772-1221','IM','Undergraduate','Data)')

INSERT INTO Student(SUID,StudFName,StudLName,StudEmail, StudGender,StudPhoneNum,StudMajor,StudLevel,TargetJob)

[VALUES('S02','John','Williams','jwilliams@syr.edu','Male','253-432-1290','IM','Graduate','Data](mailto:VALUES('S02','John','Williams','jwilliams@syr.edu','Male','253-432-1290','IM','Graduate','Data)')

INSERT INTO Student(SUID,StudFName,StudLName,StudEmail, StudGender,StudPhoneNum,StudMajor,StudLevel,TargetJob)

[VALUES('S03','Robert','Johnson','rjohnson@syr.edu','Male','431-333-7129','ADS','Graduate','Consulting](mailto:VALUES('S03','Robert','Johnson','rjohnson@syr.edu','Male','431-333-7129','ADS','Graduate','Consulting)')

INSERT INTO Student(SUID,StudFName,StudLName,StudEmail, StudGender,StudPhoneNum,StudMajor,StudLevel,TargetJob)

[VALUES('S04','Mary','Davis','mdavis@syr.edu','Female','576-101-1119','EDS','Graduate','Consulting](mailto:VALUES('S04','Mary','Davis','mdavis@syr.edu','Female','576-101-1119','EDS','Graduate','Consulting)')

INSERT INTO Student(SUID,StudFName,StudLName,StudEmail, StudGender,StudPhoneNum,StudMajor,StudLevel,TargetJob)

VALUES('S05','Jennifer','Cuomo','jcuomo@syr.edu','Female','482-201-2973','LIS','Graduate',NULL); ---Insert data - Course table---

INSERT INTO Course(CourseID,SubjectCode,TTID,CourseName,PreRequestCourse1,PreRequestCourse2)

VALUES('IST618','IST',NULL,'Information Policy',NULL,NULL)

INSERT INTO Course(CourseID,SubjectCode,TTID,CourseName,PreRequestCourse1,PreRequestCourse2)

VALUES('IST659','IST','3','Database Management',NULL,NULL)

INSERT INTO Course(CourseID,SubjectCode,TTID,CourseName,PreRequestCourse1,PreRequestCourse2)

VALUES('IST687','IST','4','Data Science',NULL,NULL)

INSERT INTO Course(CourseID,SubjectCode,TTID,CourseName,PreRequestCourse1,PreRequestCourse2)

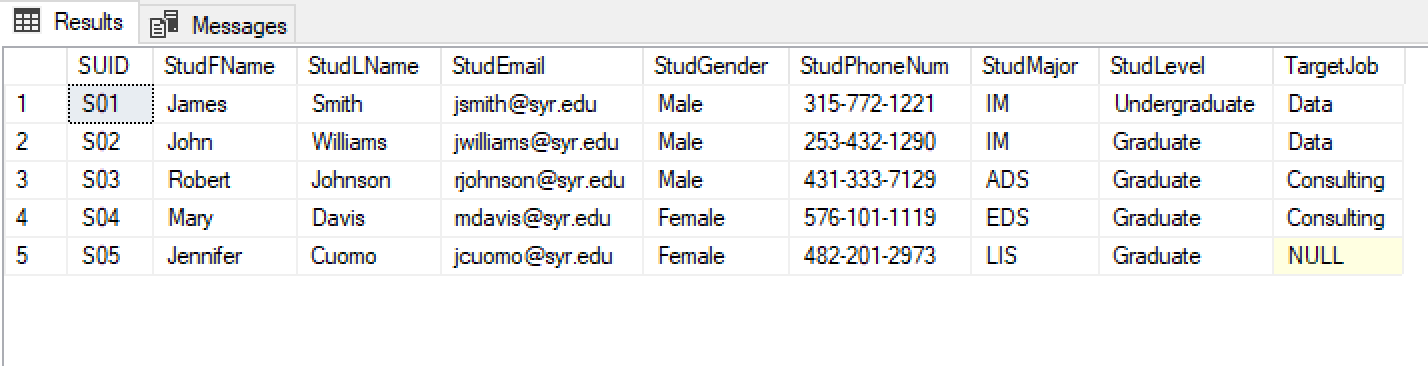
VALUES('IST707','IST','4','Data Analytics','IST687',NULL)

INSERT INTO Course(CourseID,SubjectCode,TTID,CourseName,PreRequestCourse1,PreRequestCourse2)

VALUES('IST722','IST','3','Data Warehouse','IST659',NULL)

INSERT INTO Course(CourseID,SubjectCode,TTID,CourseName,PreRequestCourse1,PreRequestCourse2)

VALUES('MBC638','MBC','1','Data Analysis and Decision Making',NULL,NULL);



---Insert data - Course table---

INSERT INTO Course(CourseID,SubjectCode,TTID,CourseName,PreRequestCourse1,PreRequestCourse2)

VALUES('IST618','IST',NULL,'Information Policy',NULL,NULL)

INSERT INTO Course(CourseID,SubjectCode,TTID,CourseName,PreRequestCourse1,PreRequestCourse2)

VALUES('IST659','IST','3','Database Management',NULL,NULL)

INSERT INTO Course(CourseID,SubjectCode,TTID,CourseName,PreRequestCourse1,PreRequestCourse2)

VALUES('IST687','IST','4','Data Science',NULL,NULL)

INSERT INTO Course(CourseID,SubjectCode,TTID,CourseName,PreRequestCourse1,PreRequestCourse2)

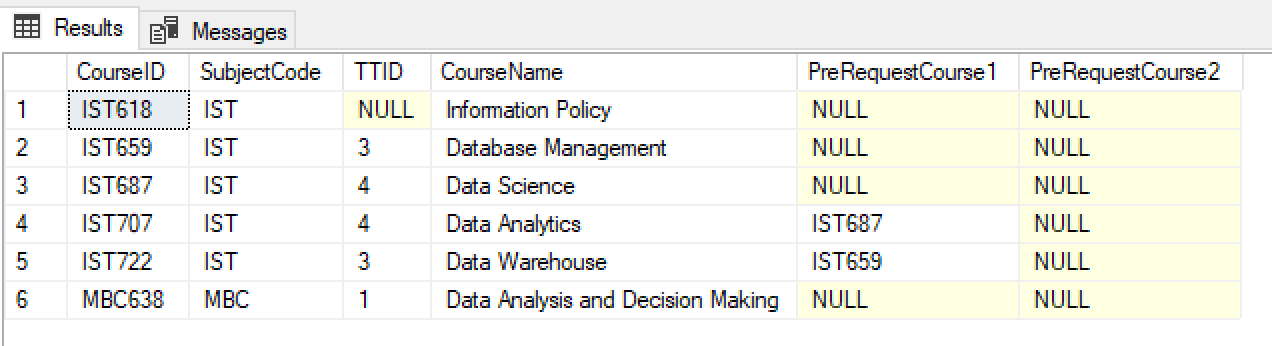
VALUES('IST707','IST','4','Data Analytics','IST687',NULL)

INSERT INTO Course(CourseID,SubjectCode,TTID,CourseName,PreRequestCourse1,PreRequestCourse2)

VALUES('IST722','IST','3','Data Warehouse','IST659',NULL)

INSERT INTO Course(CourseID,SubjectCode,TTID,CourseName,PreRequestCourse1,PreRequestCourse2)

VALUES('MBC638','MBC','1','Data Analysis and Decision Making',NULL,NULL);



---Insert data - Class table---

INSERT INTO Class(ClassID,ClassRoomID,CourseID,InstID,Credits,Semester,StartDate,EndDate,Duration,ClassSize,WKSch,Enrollment)

VALUES('618-M001','H002','IST618','5','3','Summer2019','2019-5-1 00:00:00.000', '2019-6-18 00:00:00.000','15:00-18:00','27','F',1)

INSERT INTO Class(ClassID,ClassRoomID,CourseID,InstID,Credits,Semester,StartDate,EndDate,Duration,ClassSize,WKSch,Enrollment)

VALUES('638-M001','W007','MBC638','3','3','Spring2019',2019-1-21, 2019-5-4,'17:00-18:20','80','TH',1)

INSERT INTO Class(ClassID,ClassRoomID,CourseID,InstID,Credits,Semester,StartDate,EndDate,Duration,ClassSize,WKSch,Enrollment)

VALUES('659-M001','H117','IST659','2','3','Fall2019',2019-8-26, 2019-12-12,'9:30-12:30','28','W',0)

INSERT INTO Class(ClassID,ClassRoomID,CourseID,InstID,Credits,Semester,StartDate,EndDate,Duration,ClassSize,WKSch,Enrollment)

VALUES('659-M002','H113','IST659','1','3','Fall2019',2019-8-26, 2019-12-12,'11:00-14:00','28','T',1)

INSERT INTO Class(ClassID,ClassRoomID,CourseID,InstID,Credits,Semester,StartDate,EndDate,Duration,ClassSize,WKSch,Enrollment)

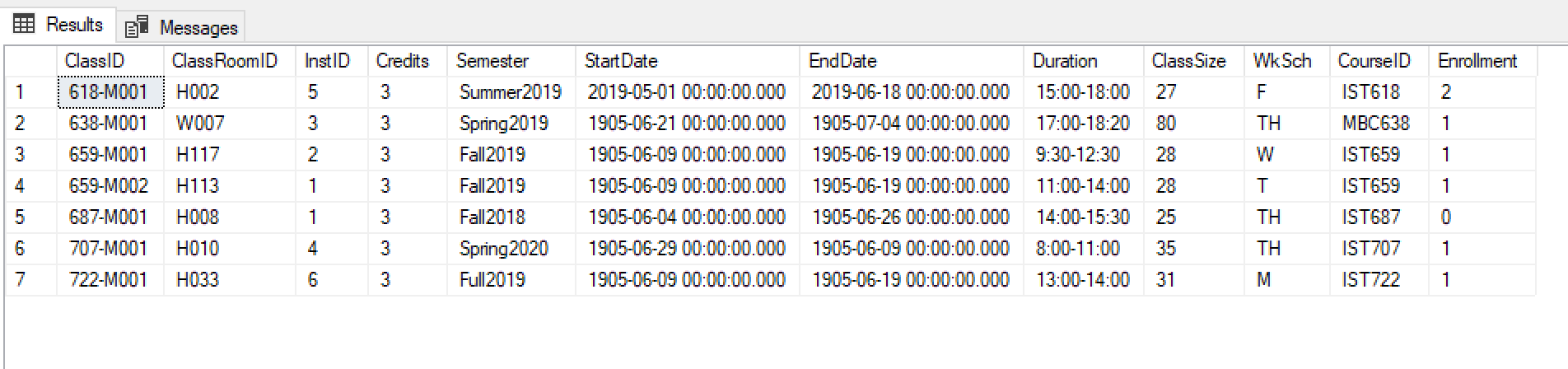
VALUES('687-M001','H008','IST687','1','3','Fall2018',2019-8-31, 2019-12-5,'14:00-15:30','25','TH',0)

INSERT INTO Class(ClassID,ClassRoomID,CourseID,InstID,Credits,Semester,StartDate,EndDate,Duration,ClassSize,WKSch,Enrollment)

VALUES('707-M001','H010','IST707','4','3','Spring2020',2019-1-13, 2019-4-30,'8:00-11:00','35','TH',1)

INSERT INTO Class(ClassID,ClassRoomID,CourseID,InstID,Credits,Semester,StartDate,EndDate,Duration,ClassSize,WKSch,Enrollment)

VALUES('722-M001','H033','IST722','6','3','Full2019',2019-8-26, 2019-12-12,'13:00-14:00','31','M',1);



---Insert data - Registration table---

INSERT INTO Registration(RegstID,RegstDate,SUID,ClassID)

VALUES('R01','8/22/2018','S01','659-M002')

INSERT INTO Registration(RegstID,RegstDate,SUID,ClassID)

VALUES('R02','1/7/2019','S03','638-M001')

INSERT INTO Registration(RegstID,RegstDate,SUID,ClassID)

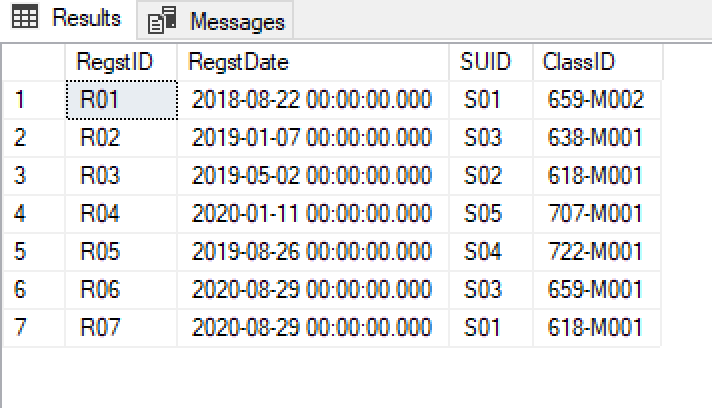
VALUES('R03','5/2/2019','S02','618-M001')

INSERT INTO Registration(RegstID,RegstDate,SUID,ClassID)

VALUES('R04','1/11/2020','S05','707-M001')

INSERT INTO Registration(RegstID,RegstDate,SUID,ClassID)

VALUES('R05','8/26/2019','S04','722-M001');



---Insert data - Company table---

INSERT INTO Company(CompanyId, CompName,CompState,CompCity,CompField,CompValue)

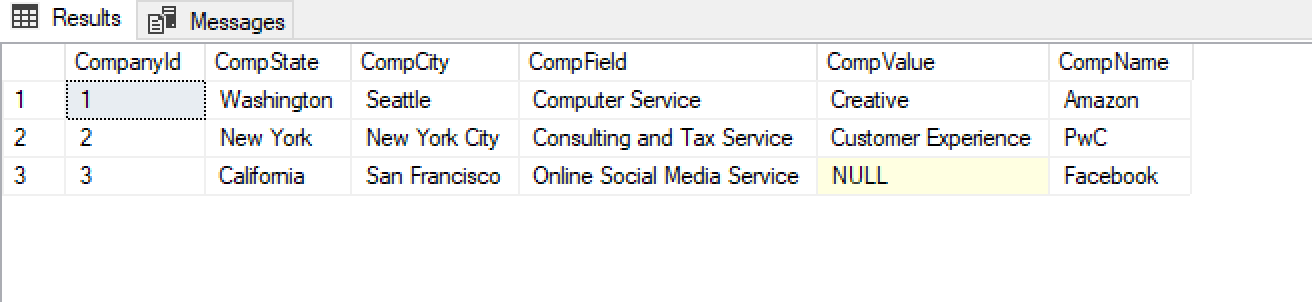
VALUES(1,'Amazon','Washington','Seattle','Computer Service','Creative')

INSERT INTO Company(CompanyId, CompName,CompState,CompCity,CompField,CompValue)

VALUES(2,'PwC','New York','New York City','Consulting and Tax Service','Customer Experience')

INSERT INTO Company(CompanyId, CompName,CompState,CompCity,CompField,CompValue)

VALUES(3,'Facebook','California','San Francisco','Online Social Media Service', NULL);



---Insert data - Job table---

INSERT INTO Job(JobID,TTID,JobType,JobRequest)

VALUES(1,1,'Consulting','Internship')

INSERT INTO Job(JobID,TTID,JobType,JobRequest)

VALUES(2,3,'Database','Full-time')

INSERT INTO Job(JobID,TTID,JobType,JobRequest)

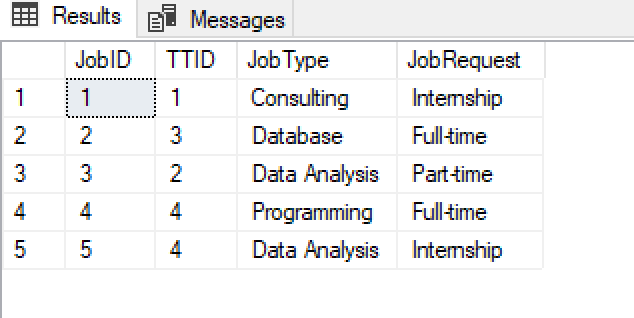
VALUES(3,2,'Data Analysis','Part-time')

INSERT INTO Job(JobID,TTID,JobType,JobRequest)

VALUES(4,4,'Programming','Full-time')

INSERT INTO Job(JobID,TTID,JobType,JobRequest)

VALUES(5,4,'Data Analysis','Internship');



---Insert data - JobOffered table---

INSERT INTO JobOffered(CompID,JobID,PostDate,ApplicationDue)

VALUES(1,1,'2020-5-1','2020-8-1')

INSERT INTO JobOffered(CompID,JobID,PostDate,ApplicationDue)

VALUES(1,2,'2020-5-1','2020-8-1')

INSERT INTO JobOffered(CompID,JobID,PostDate,ApplicationDue)

VALUES(1,3,'2020-5-1','2020-8-1')

INSERT INTO JobOffered(CompID,JobID,PostDate,ApplicationDue)

VALUES(2,1,'2020-5-1','2020-8-1')

INSERT INTO JobOffered(CompID,JobID,PostDate,ApplicationDue)

VALUES(2,2,'2020-5-1','2020-8-1')

INSERT INTO JobOffered(CompID,JobID,PostDate,ApplicationDue)

VALUES(3,1,'2020-5-1','2020-8-1')

INSERT INTO JobOffered(CompID,JobID,PostDate,ApplicationDue)

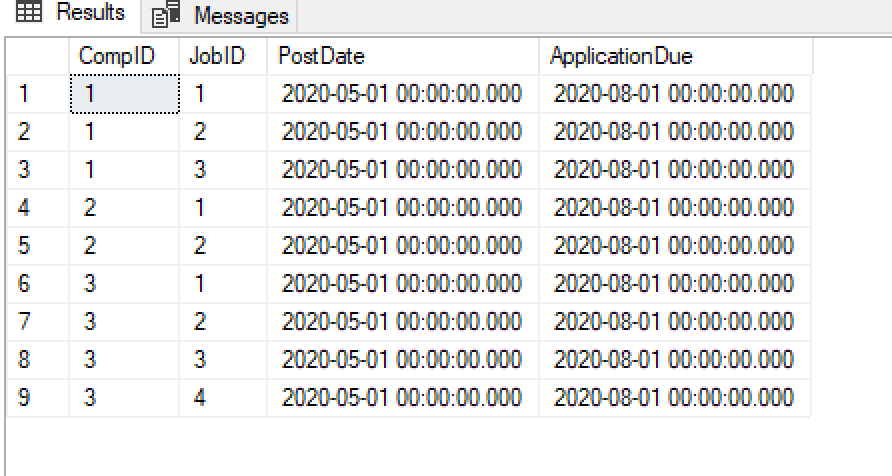
VALUES(3,2,'2020-5-1','2020-8-1')

INSERT INTO JobOffered(CompID,JobID,PostDate,ApplicationDue)

VALUES(3,3,'2020-5-1','2020-8-1')

INSERT INTO JobOffered(CompID,JobID,PostDate,ApplicationDue)

VALUES(3,4,'2020-5-1','2020-8-1');



# Major Data Questions

**For Students:**

**Q1: What are the prerequisite subjects each course requires?**

select C1.CourseID, C1.CourseName, C1.PreRequestCourse1, C2.CourseName as PreCourse1Name,C1.PreRequestCourse2,

C3.CourseName as PreCourse2Name

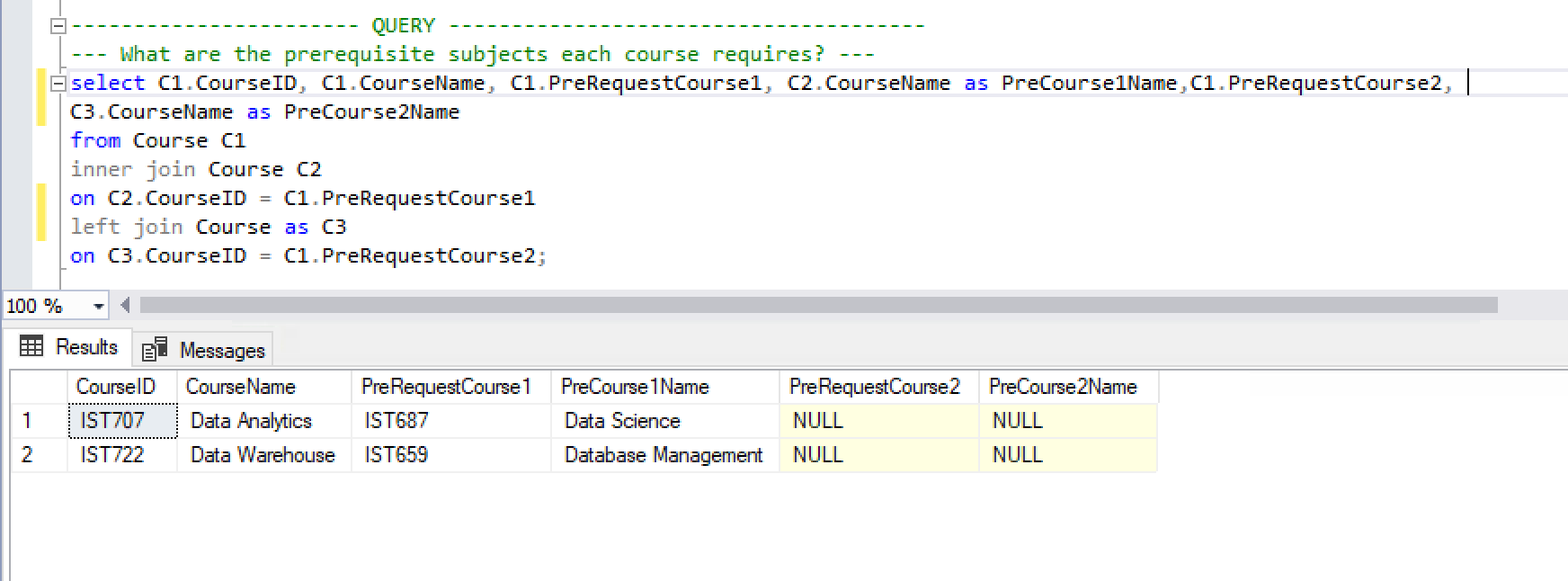
from Course C1

inner join Course C2

on C2.CourseID = C1.PreRequestCourse1

left join Course as C3

on C3.CourseID = C1.PreRequestCourse2;



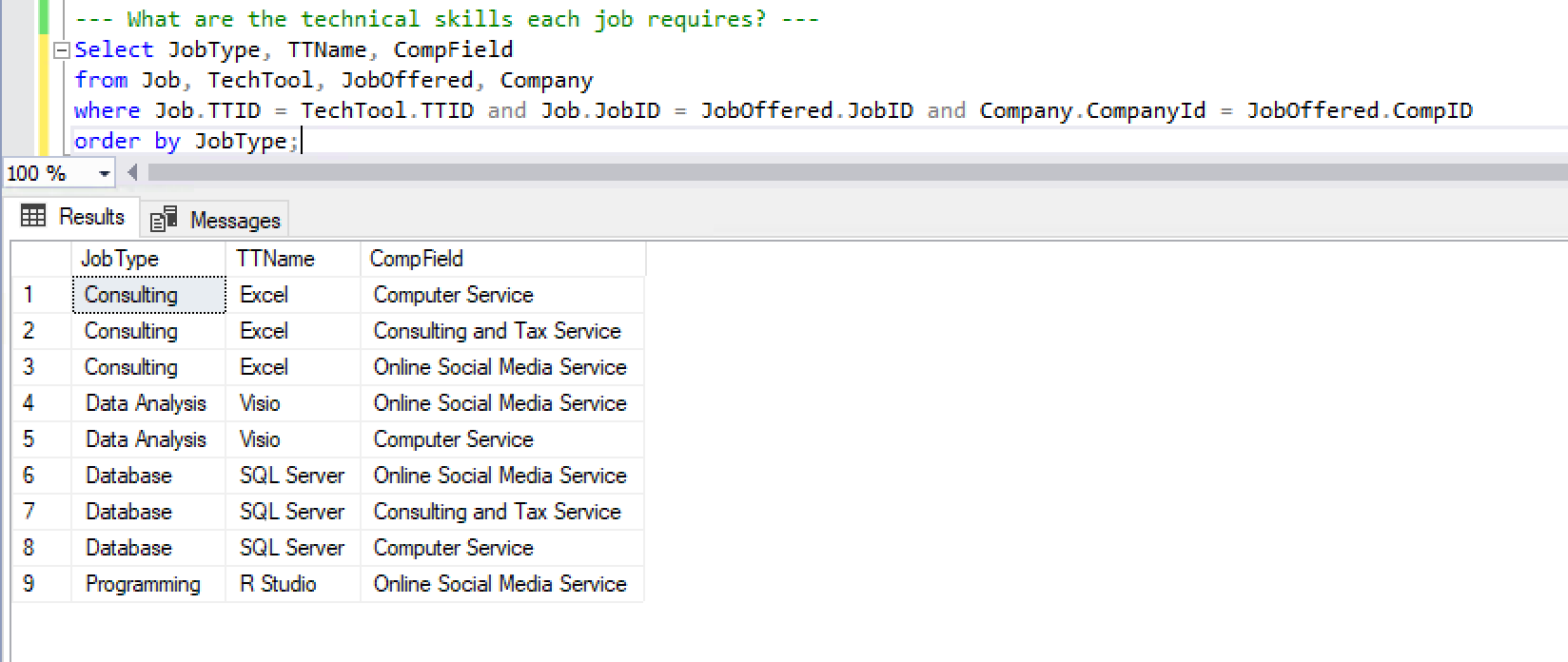
**Q2: What are the technical skills each job requires?**

Select JobType, TTName, CompField

from Job, TechTool, JobOffered, Company

where Job.TTID = TechTool.TTID and Job.JobID = JobOffered.JobID and Company.CompanyId = JobOffered.CompID

order by JobType;

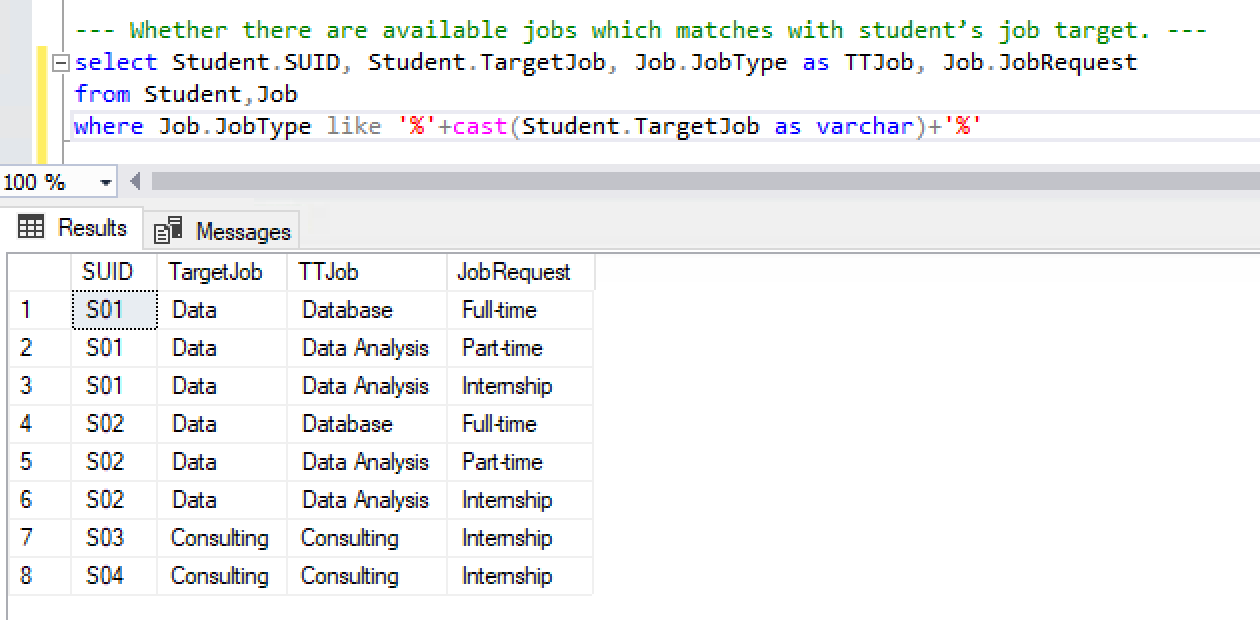


**Q3: Whether there are available jobs which matches with student’s job target.**

select Student.SUID, Student.TargetJob, Job.JobType as TTJob, Job.JobRequest

from Student,Job

where Job.JobType like '%'+cast(Student.TargetJob as varchar)+'%'



**For Instructors:**

**Q4: How many students enrolled in each class that specific instructor teaches?**

select ClassID, StudentNumb, InstName

from

(select Registration.ClassID as ClassID, count(Registration.SUID) as StudentNumb

from Registration

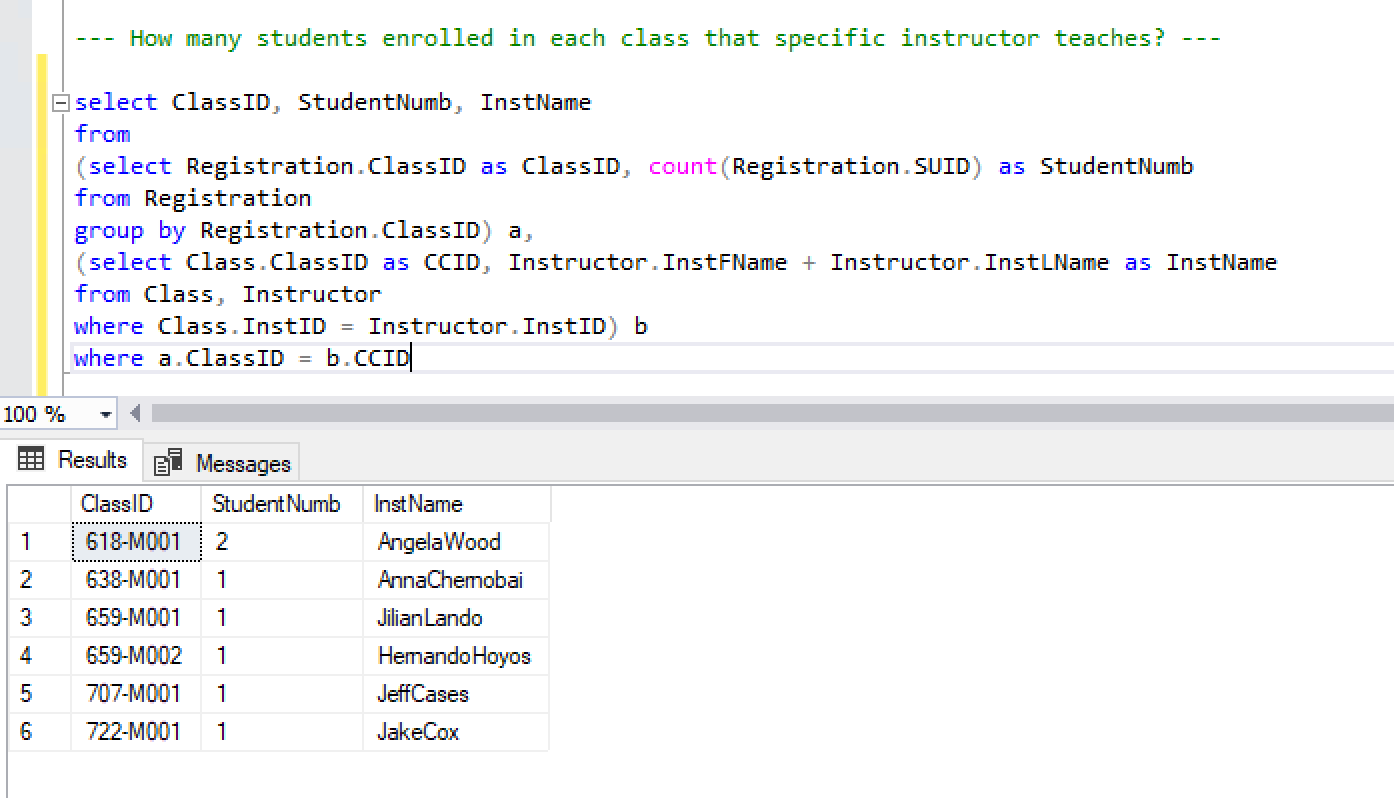
group by Registration.ClassID) a,

(select Class.ClassID as CCID, Instructor.InstFName + Instructor.InstLName as InstName

from Class, Instructor

where Class.InstID = Instructor.InstID) b

where a.ClassID = b.CCID



**Q5: Where is the classroom for each class that specific instructor teaches?**

select InstName, CCID, Room

from

(select ClassRoom.ClassRoomID + '-' + ClassRoom.ClassRoomNumber + ',' + ClassRoom.ClassBuilding as Room, Class.ClassID as CCID

from ClassRoom, Class

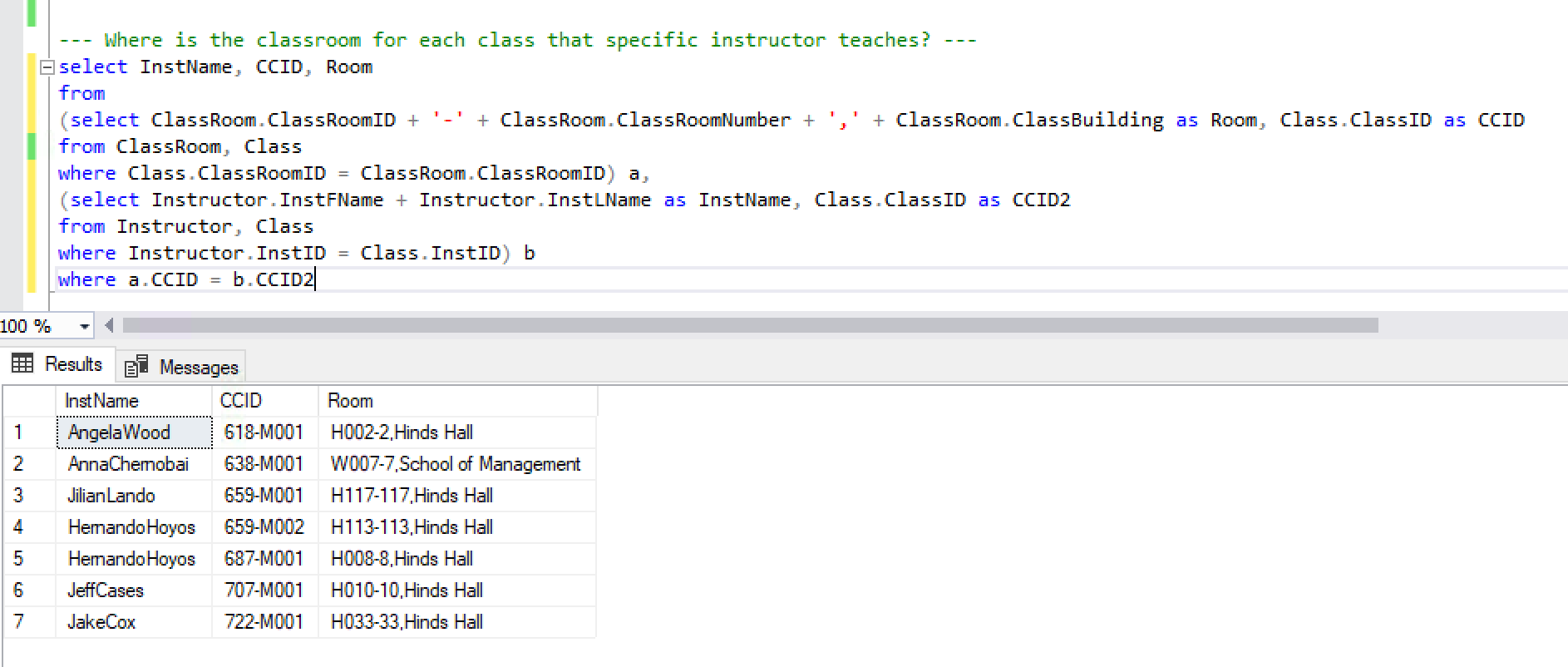
where Class.ClassRoomID = ClassRoom.ClassRoomID) a,

(select Instructor.InstFName + Instructor.InstLName as InstName, Class.ClassID as CCID2

from Instructor, Class

where Instructor.InstID = Class.InstID) b

where a.CCID = b.CCID2



**For Companies:**

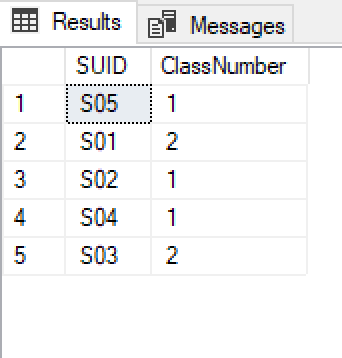
**Q6: How many and what kind of courses each student has taken?**

select Student.SUID, count(Registration.ClassID) as ClassNumber

from Student, Registration

where Student.SUID = Registration.SUID

group by Student.SUID



select SUID, StudName, CourseName, TechTool

from

(select Student.SUID as SUID, Student.StudFName + '' + Student.StudLName as StudName, Registration.ClassID as RCID

from Registration, Student

where Student.SUID = Registration.SUID) a,

(select Course.CourseName as CourseName, TechTool.TTName as TechTool, TechTool.TTID as TTID, Class.ClassID as CCID

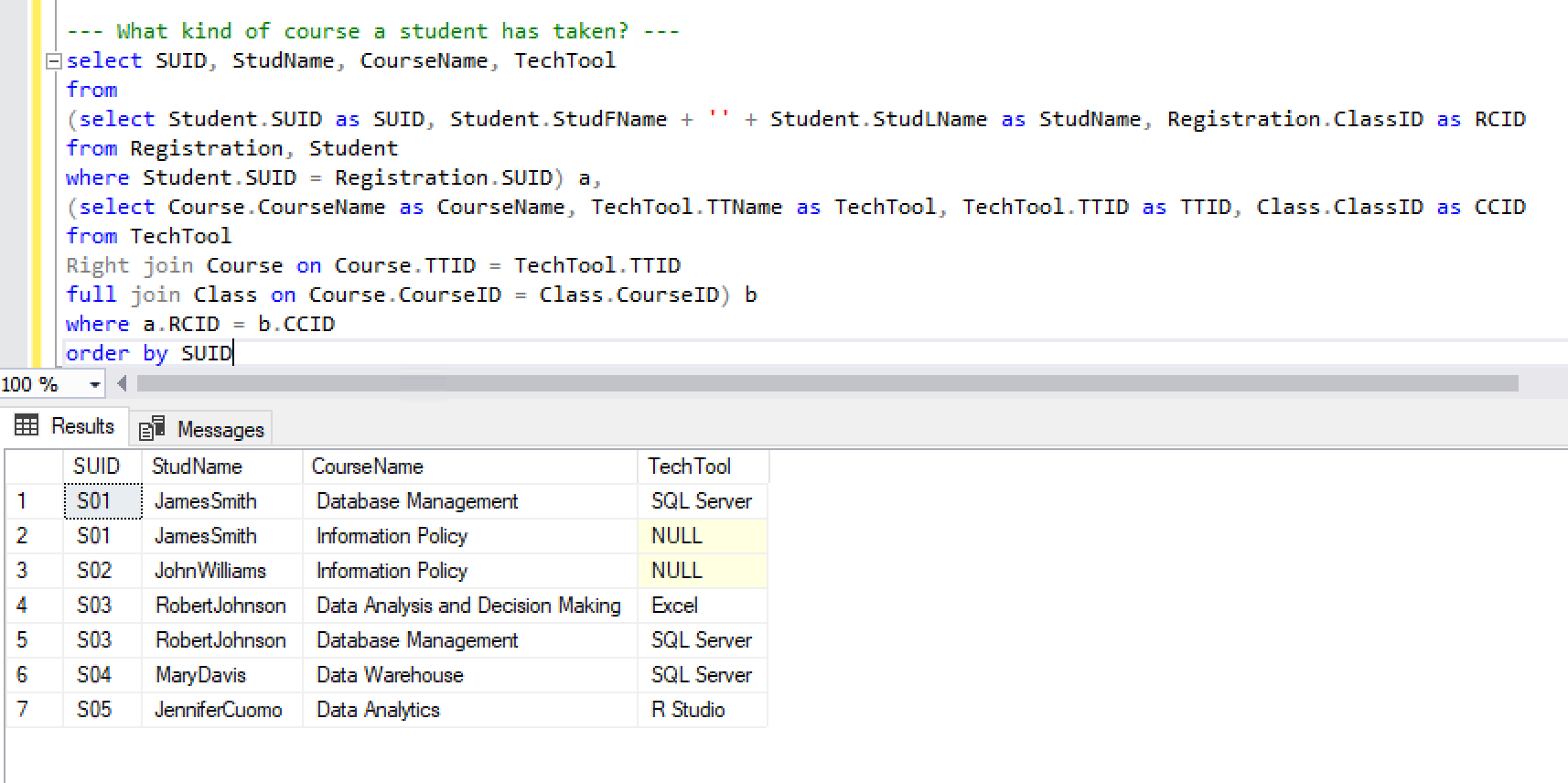
from TechTool

Right join Course on Course.TTID = TechTool.TTID

full join Class on Course.CourseID = Class.CourseID) b

where a.RCID = b.CCID

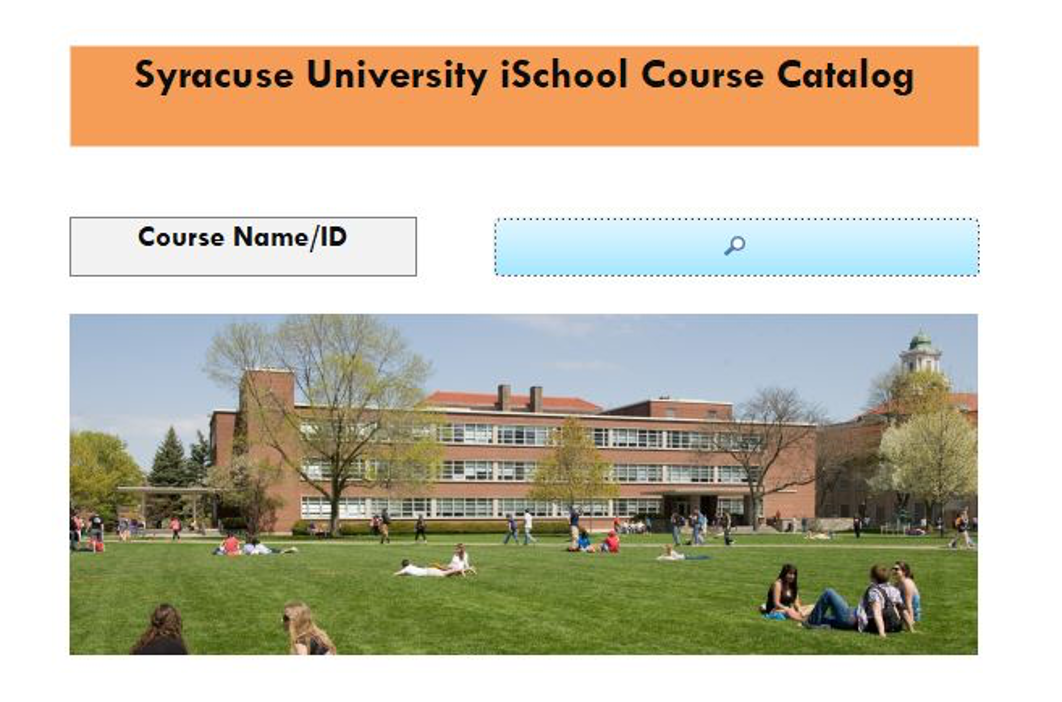
order by SUID



# Interfaces

1. **SU iSchool course catalog main page**

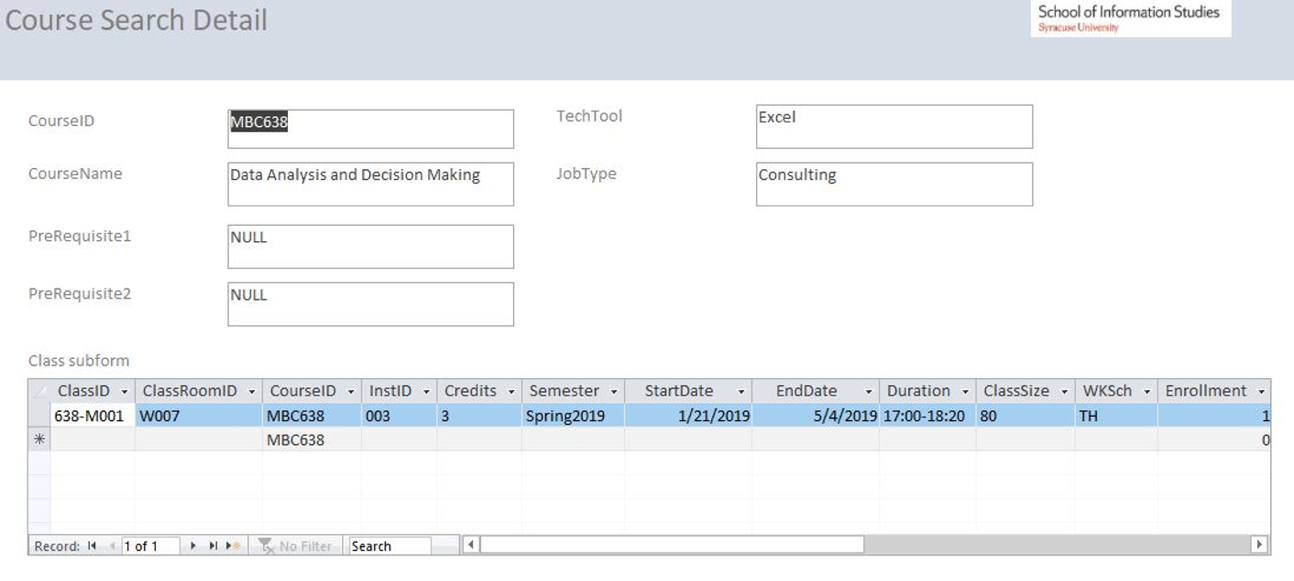
Users can search for available courses by inputting course name or course ID



1. **Course search details**

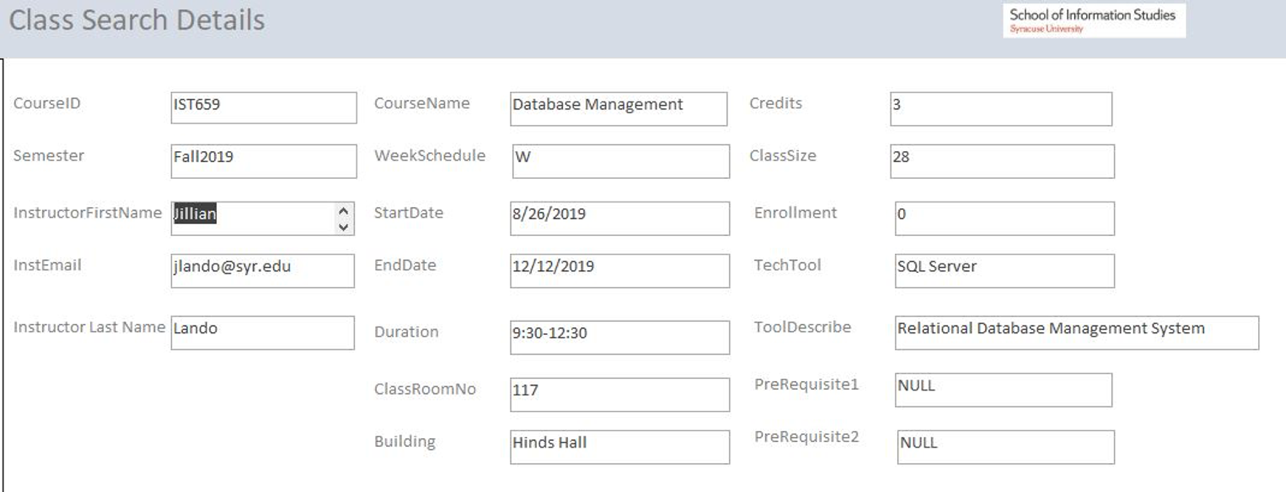
When users enter course name or course ID, they move on to this page and

view detailed information about the course they searched.



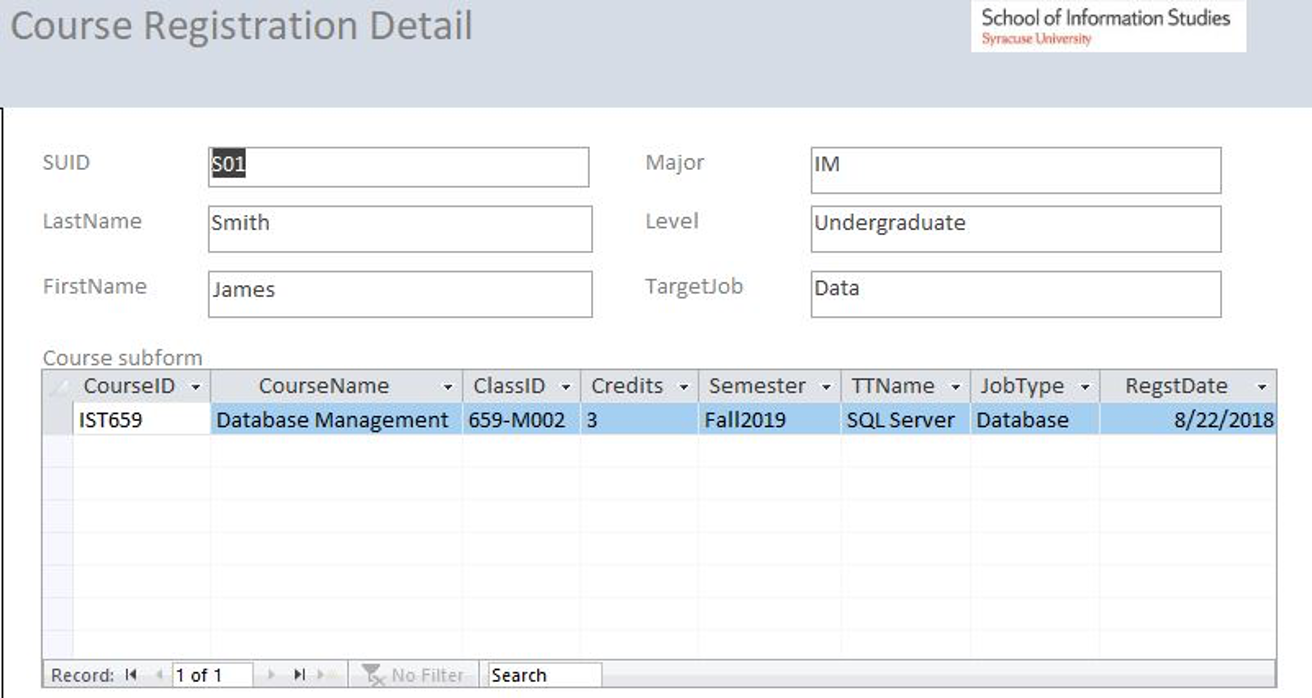
1. **Class search details**

Users can view details of classes for a specific course



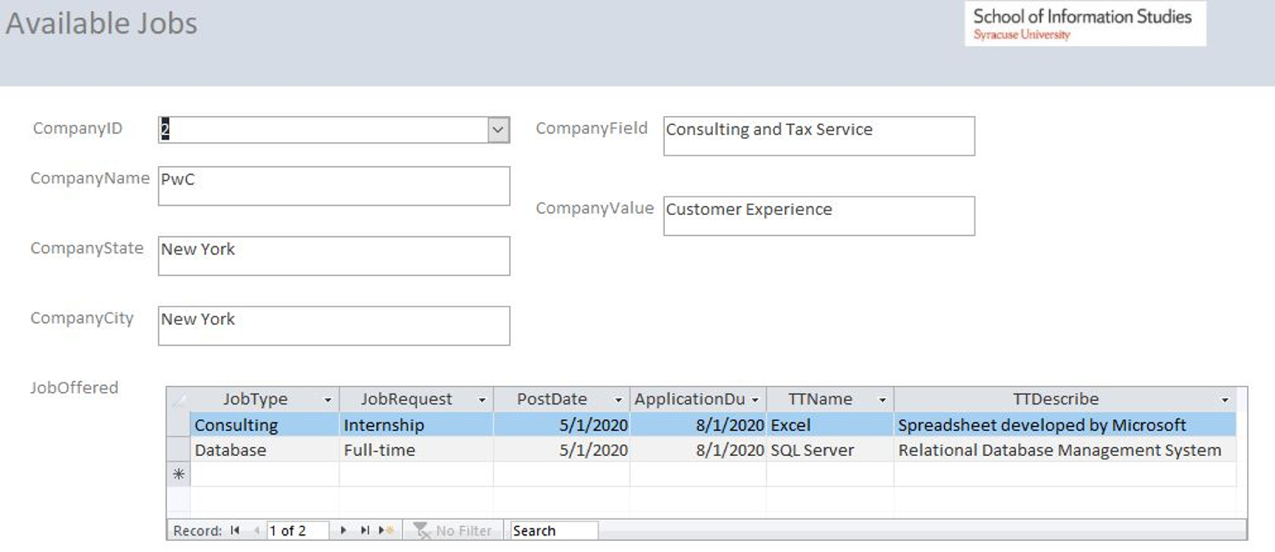
1. **Course registration details**

Users can view detailed information about courses in which each student enrolled



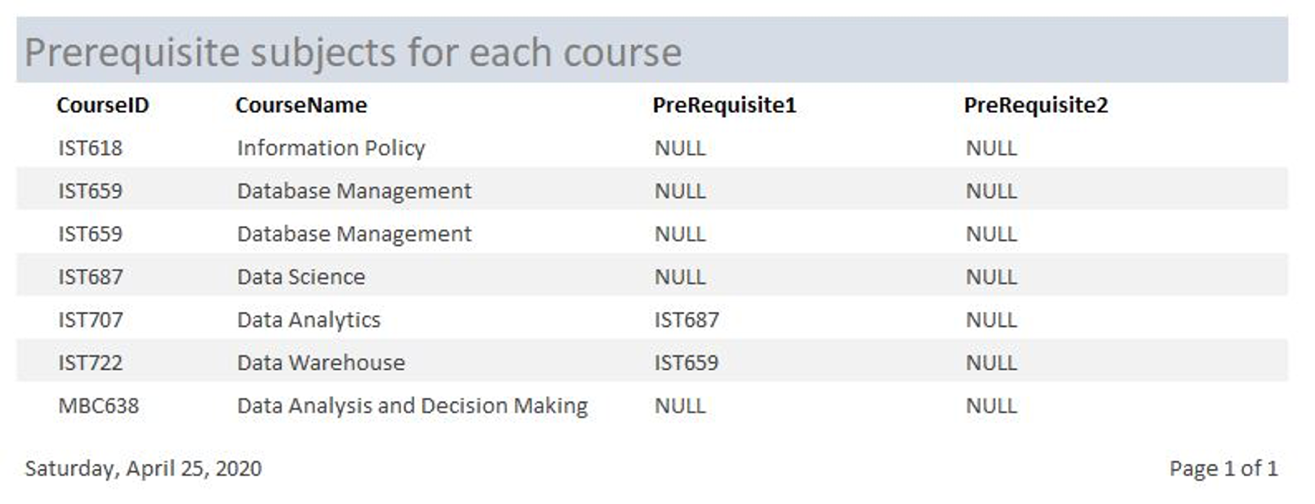
1. **Available job details**

Users can view available job information



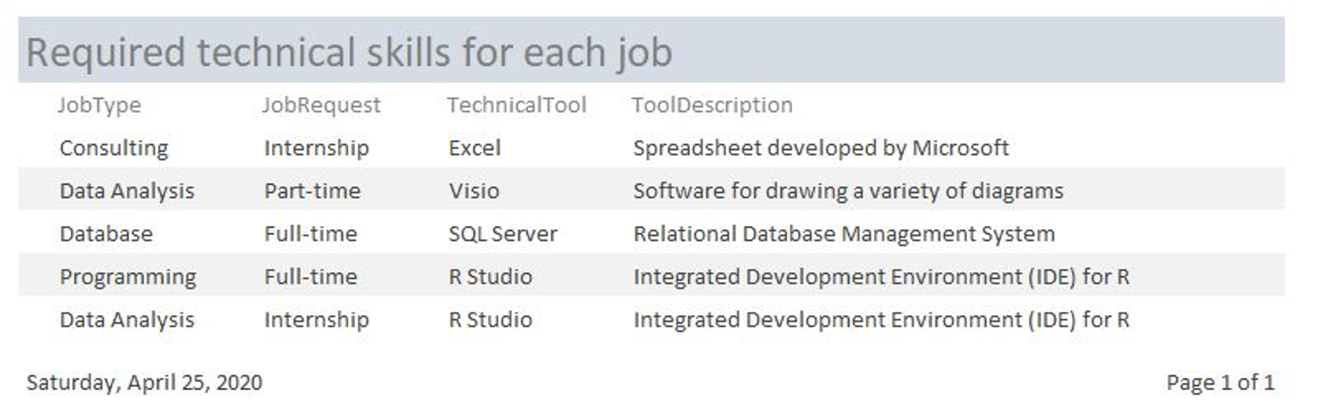
# Report

1. **Prerequisite subjects for each course**

It will show up to 2 prerequisite subjects for each course

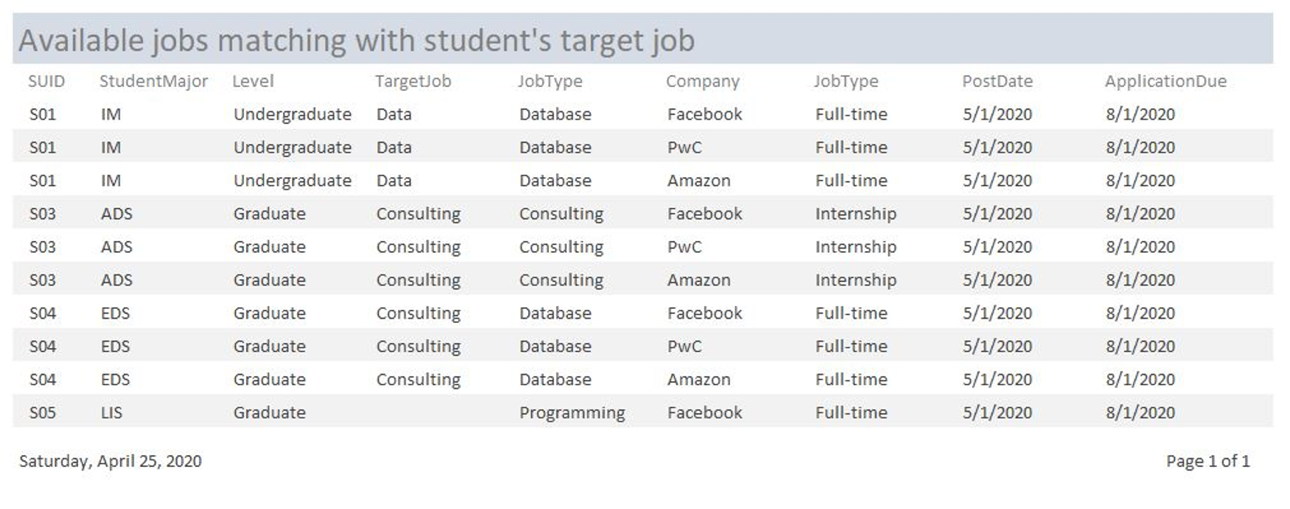
1. **Required technical skills for each job**

It will show the required technical skills of each job type



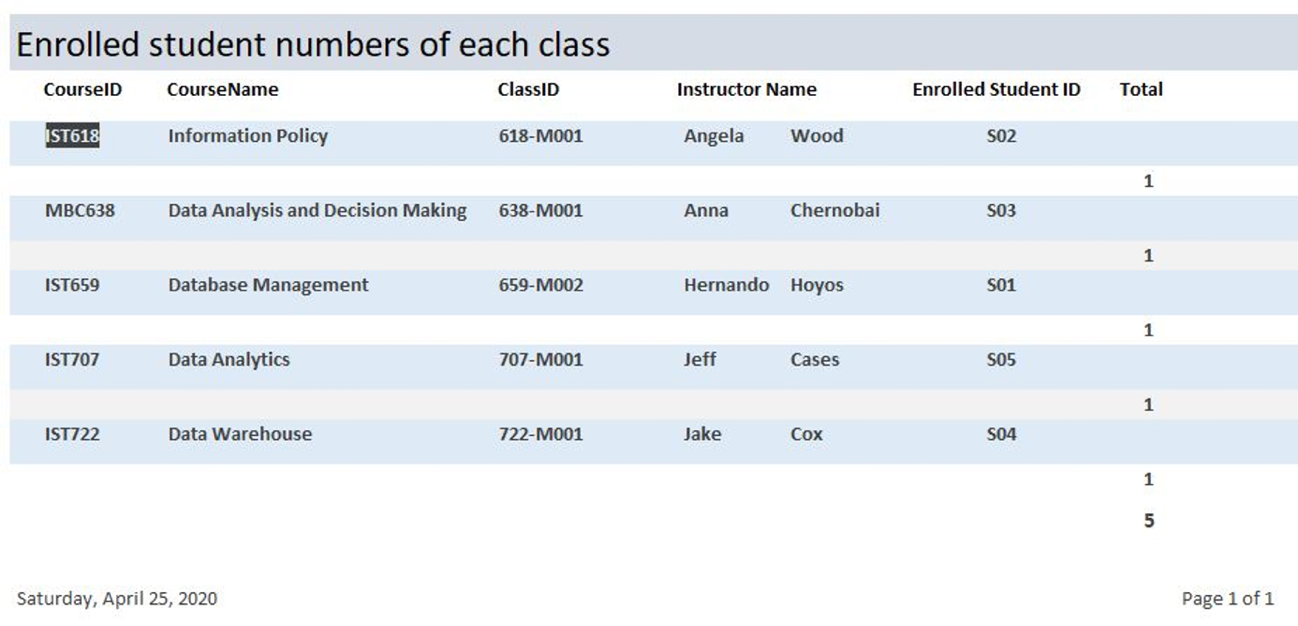
1. **Available jobs matching with students’ target jobs**

It will show a list of available jobs which match with students’ target jobs



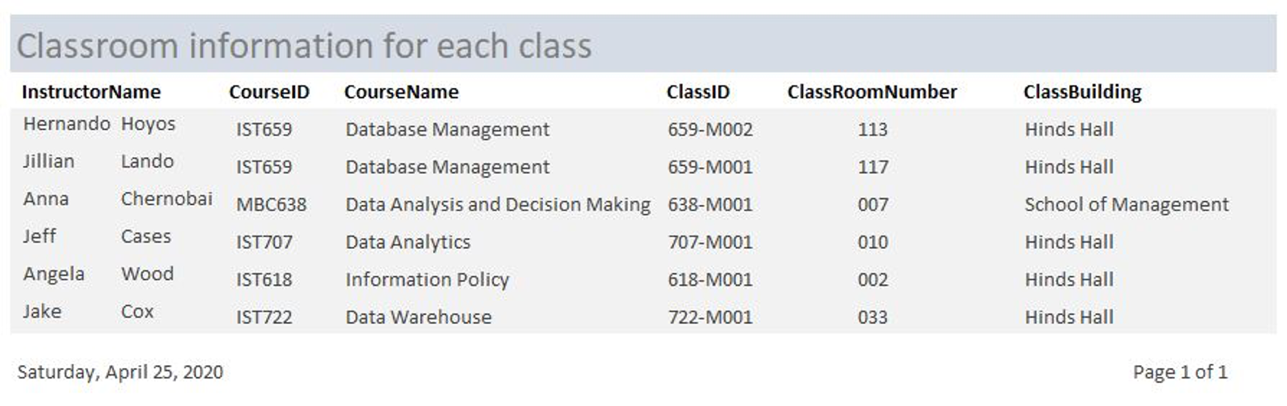
1. **Enrolled student numbers of each class**

It will show how many students enrolled in each class that specific instructor teaches



1. **Classroom information for each class**

It will show the classroom information for each class that specific instructor teaches



1. **Course record of each student**

It will show how many and what kind of courses each student has taken

