Egypt-Japan University of Science and Technology
الجامعة المصرية اليابانية للعلوم و التكنولوجيا
エジプト日本科学技術大学

University Management System

This document is submitted for Database Systems project, by:

Ebrahim Ramadan – 320220029 (Team Leader)

Momen Hesham – 320220060

Ahmed Hamed– 320230024

Renada Tamer – 320230143

Youssef Kamal - 320220025

Under supervision of /

Business and System Requirements Specification

1. Key Stakeholders

1.1 Students Requirements:

- Personal profile management (NID, Name, Email, Address)
- Academic record access and management
- Course registration and tracking
- Access to transcripts and GPA information
- Library loan management
- Research assistant role management
- Attendance tracking
- Academic advisory services
- Club membership management

1.2 Academic Staff

Professors Requirements:

- Profile management (Name, NationalID, Email)
- Department affiliation management
- Course teaching assignments

- Research project supervision

- Faculty relationship management

- Professional ID management

Instructors Requirements:

- Profile management (NationalID, Email)

- Course teaching assignments

- Department affiliation

- Office hour management

1.3 Administrative Entities

Department Management Requirements:

- Department information management

- Staff assignment management

- Building and room allocation

- Course offering management

- Faculty relationship management

Faculty Administration Requirements:

- Faculty profile management

- Department oversight

- Research project management
- Staff allocation
- Resource management

1.4 Facility Management

Building Management Requirements:

- Building information management
- Room allocation and tracking
- Capacity management
- Equipment tracking

2. System Requirements

2.1 Academic Management

Course Management Requirements:

- Course creation and updates
- Prerequisites management
- Grade tracking
- Course Code management
- Credit hour tracking

- **Course material management**

Examination System Requirements:

- **Exam scheduling**
- **Score recording**
- **Weight management**
- **Exam type classification**
- **Duration tracking**

Transcript Management Requirements:

- **GPA calculation**
- **Course history tracking**
- **Academic performance recording**
- **Semester-wise grade management**

2.2 Administrative Functions

Attendance System Requirements:

- **Student attendance tracking**
- **Attendance date recording**
- **Attendance status management**

Library System Requirements:

- **Loan management**
- **Due date tracking**
- **Book return status**
- **Fine management**

Club Management Requirements:

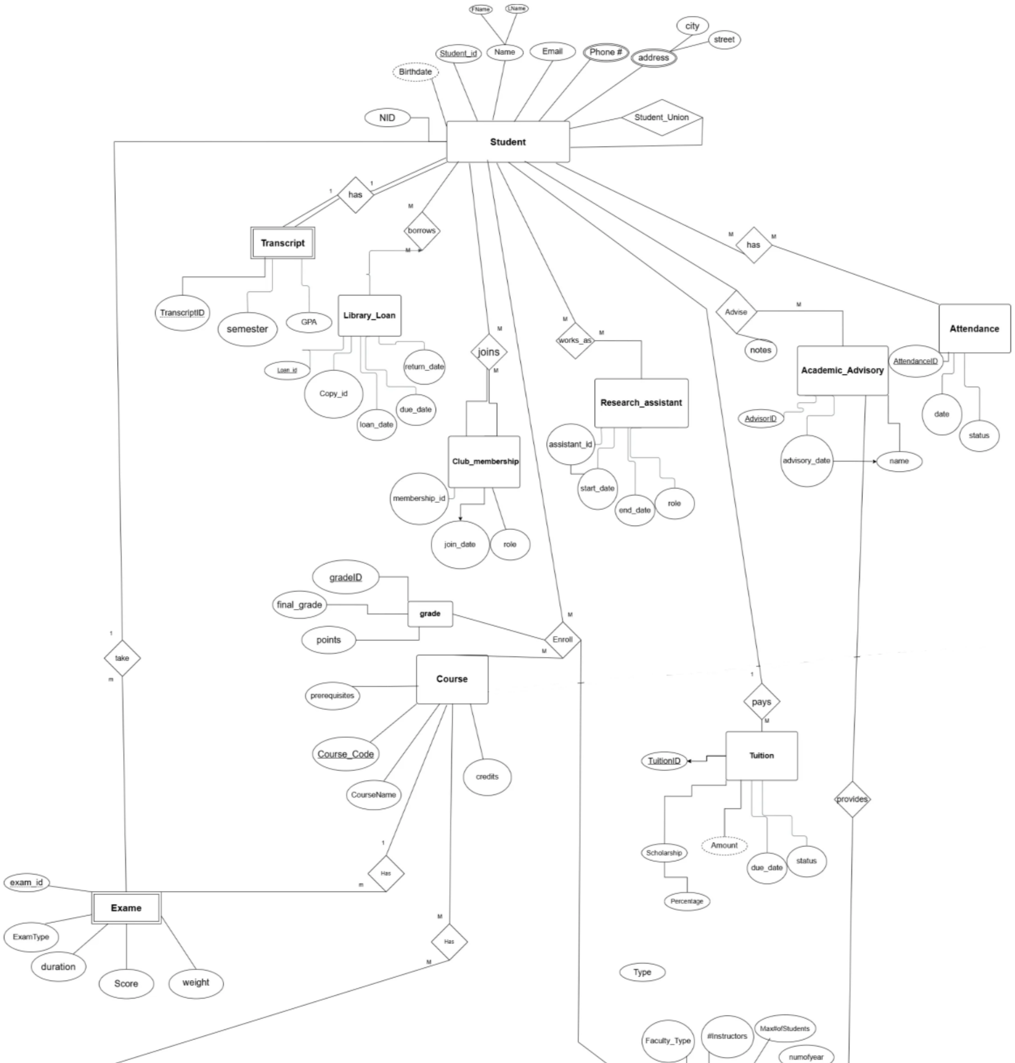
- **Club membership tracking**
- **Student participation records**
- **Club activity management**

2.3 Research Management

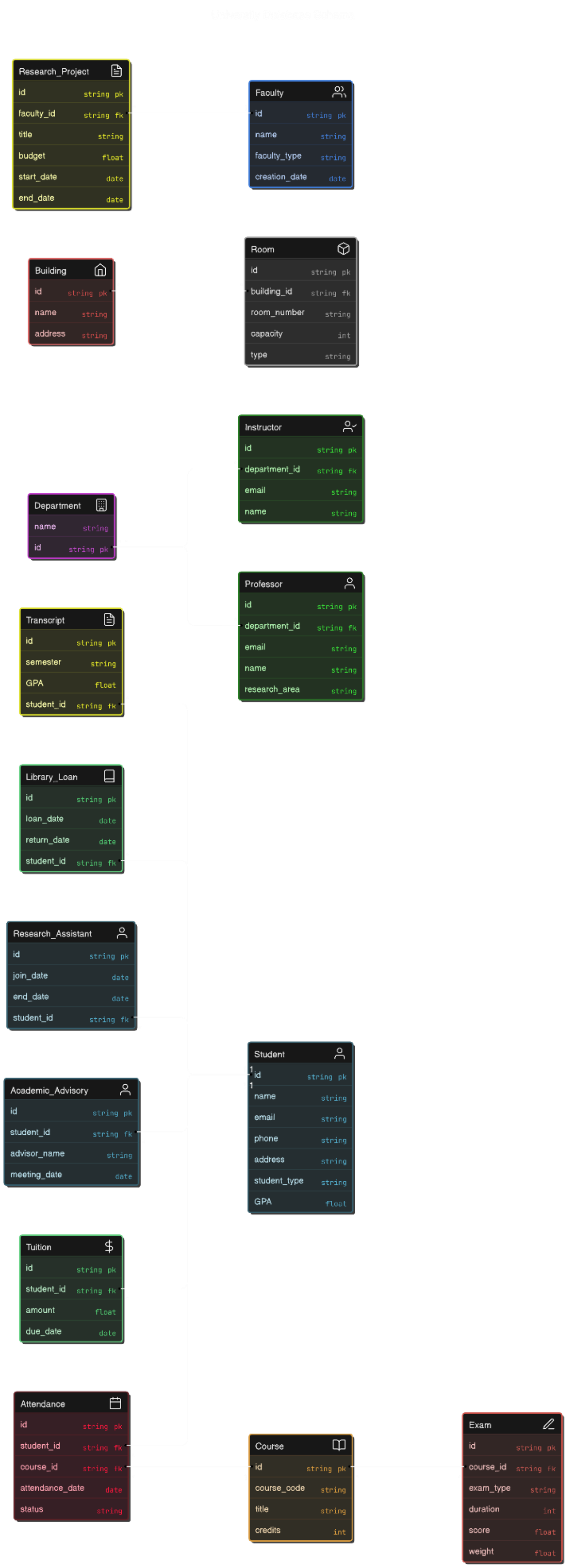
Research Projects Requirements:

- **Project tracking**
- **Student assistant assignment**
- **Faculty supervision**
- **Project status management**
- **Timeline management**

ERD


FName
LName
Student_id
Name
Email
Phone #
address
city
street
Birthdate
NID
Student
Student_Union
1
has
1
Transcript
TranscriptID
semester
GPA
M
borrows
M
Library_Loan
Loan_id
Copy_id
loan_date
due_date
return_date
M
joins
M
Club_membership
membership_id
join_date
role
M
works_as
M
Research_assistant
assistant_id
start_date
end_date
role
M
has
M
Advise
M
notes
Academic_Advisory
AdvisorID
advisory_date
name
Attendance
AttendanceID
date
status
gradeID
final_grade
grade
points
M
Enroll
M
Course
prerequisites
Course_Code
CourseName
credits
1
take
m
1
Has
m
exam_id
Exame
ExamType
duration
Score
weight
M
Has
M
1
pays
M
Tuition
TuitionID
Scholarship
Percentage
Amount
due_date
status
provides
Type
Faculty_Type
#Instructors
Max#ofStudents
numofyear

Schema



University Database Schema

Research_Project	
id	string pk
faculty_id	string fk
title	string
budget	float
start_date	date
end_date	date

Faculty	
id	string pk
name	string
faculty_type	string
creation_date	date

Building	
id	string pk
name	string
address	string

Room	
id	string pk
building_id	string fk
room_number	string
capacity	int
type	string

Instructor	
id	string pk
department_id	string fk
email	string
name	string

Department	
name	string
id	string pk

Professor	
id	string pk
department_id	string fk
email	string
name	string
research_area	string

Transcript	
id	string pk
semester	string
GPA	float
student_id	string fk

Library_Loan	
id	string pk
loan_date	date
return_date	date
student_id	string fk

Research_Assistant	
id	string pk
join_date	date
end_date	date
student_id	string fk

Student	
id	string pk
name	string
email	string
phone	string
address	string
student_type	string
GPA	float

Academic_Advisory	
id	string pk
student_id	string fk
advisor_name	string
meeting_date	date

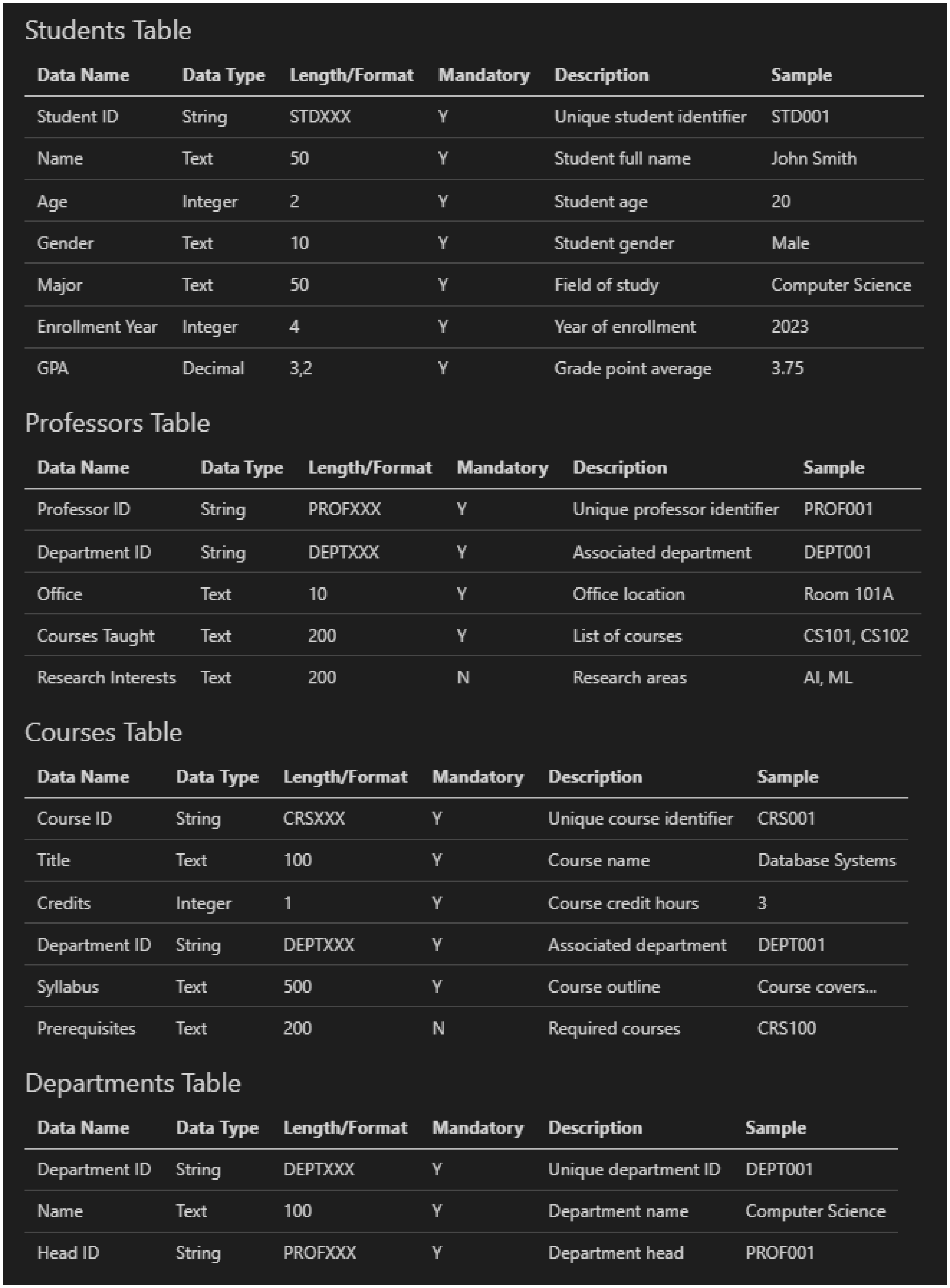
Tuition	
id	string pk
student_id	string fk
amount	float
due_date	date

Attendance	
id	string pk
student_id	string fk
course_id	string fk
attendance_date	date
status	string

Course	
id	string pk
course_code	string
title	string
credits	int

Exam	
id	string pk
course_id	string fk
exam_type	string
duration	int
score	float
weight	float

Data Dictionary



Students Table

Data Name	Data Type	Length/Format	Mandatory	Description	Sample
Student ID	String	STDXXX	Y	Unique student identifier	STD001
Name	Text	50	Y	Student full name	John Smith
Age	Integer	2	Y	Student age	20
Gender	Text	10	Y	Student gender	Male
Major	Text	50	Y	Field of study	Computer Science
Enrollment Year	Integer	4	Y	Year of enrollment	2023
GPA	Decimal	3,2	Y	Grade point average	3.75

Professors Table

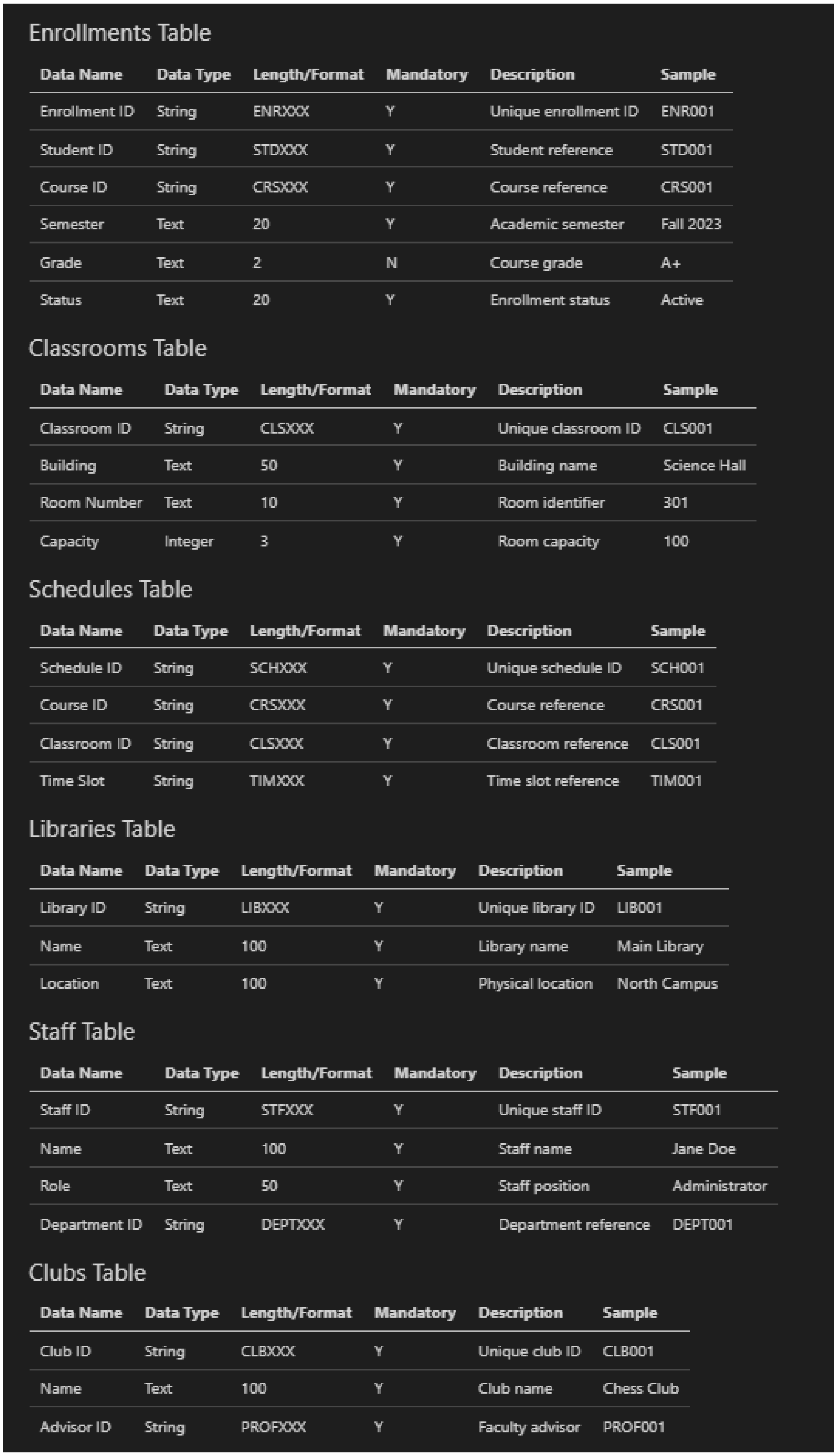
Data Name	Data Type	Length/Format	Mandatory	Description	Sample
Professor ID	String	PROFXXX	Y	Unique professor identifier	PROF001
Department ID	String	DEPTXXX	Y	Associated department	DEPT001
Office	Text	10	Y	Office location	Room 101A
Courses Taught	Text	200	Y	List of courses	CS101, CS102
Research Interests	Text	200	N	Research areas	AI, ML

Courses Table

Data Name	Data Type	Length/Format	Mandatory	Description	Sample
Course ID	String	CRSXXX	Y	Unique course identifier	CRS001
Title	Text	100	Y	Course name	Database Systems
Credits	Integer	1	Y	Course credit hours	3
Department ID	String	DEPTXXX	Y	Associated department	DEPT001
Syllabus	Text	500	Y	Course outline	Course covers...
Prerequisites	Text	200	N	Required courses	CRS100

Departments Table

Data Name	Data Type	Length/Format	Mandatory	Description	Sample
Department ID	String	DEPTXXX	Y	Unique department ID	DEPT001
Name	Text	100	Y	Department name	Computer Science
Head ID	String	PROFXXX	Y	Department head	PROF001



Enrollments Table

Data Name	Data Type	Length/Format	Mandatory	Description	Sample
Enrollment ID	String	ENRXXX	Y	Unique enrollment ID	ENR001
Student ID	String	STDXXX	Y	Student reference	STD001
Course ID	String	CRSXXX	Y	Course reference	CRS001
Semester	Text	20	Y	Academic semester	Fall 2023
Grade	Text	2	N	Course grade	A+
Status	Text	20	Y	Enrollment status	Active

Classrooms Table

Data Name	Data Type	Length/Format	Mandatory	Description	Sample
Classroom ID	String	CLSXXX	Y	Unique classroom ID	CLS001
Building	Text	50	Y	Building name	Science Hall
Room Number	Text	10	Y	Room identifier	301
Capacity	Integer	3	Y	Room capacity	100

Schedules Table

Data Name	Data Type	Length/Format	Mandatory	Description	Sample
Schedule ID	String	SCHXXX	Y	Unique schedule ID	SCH001
Course ID	String	CRSXXX	Y	Course reference	CRS001
Classroom ID	String	CLSXXX	Y	Classroom reference	CLS001
Time Slot	String	TIMXXX	Y	Time slot reference	TIM001

Libraries Table

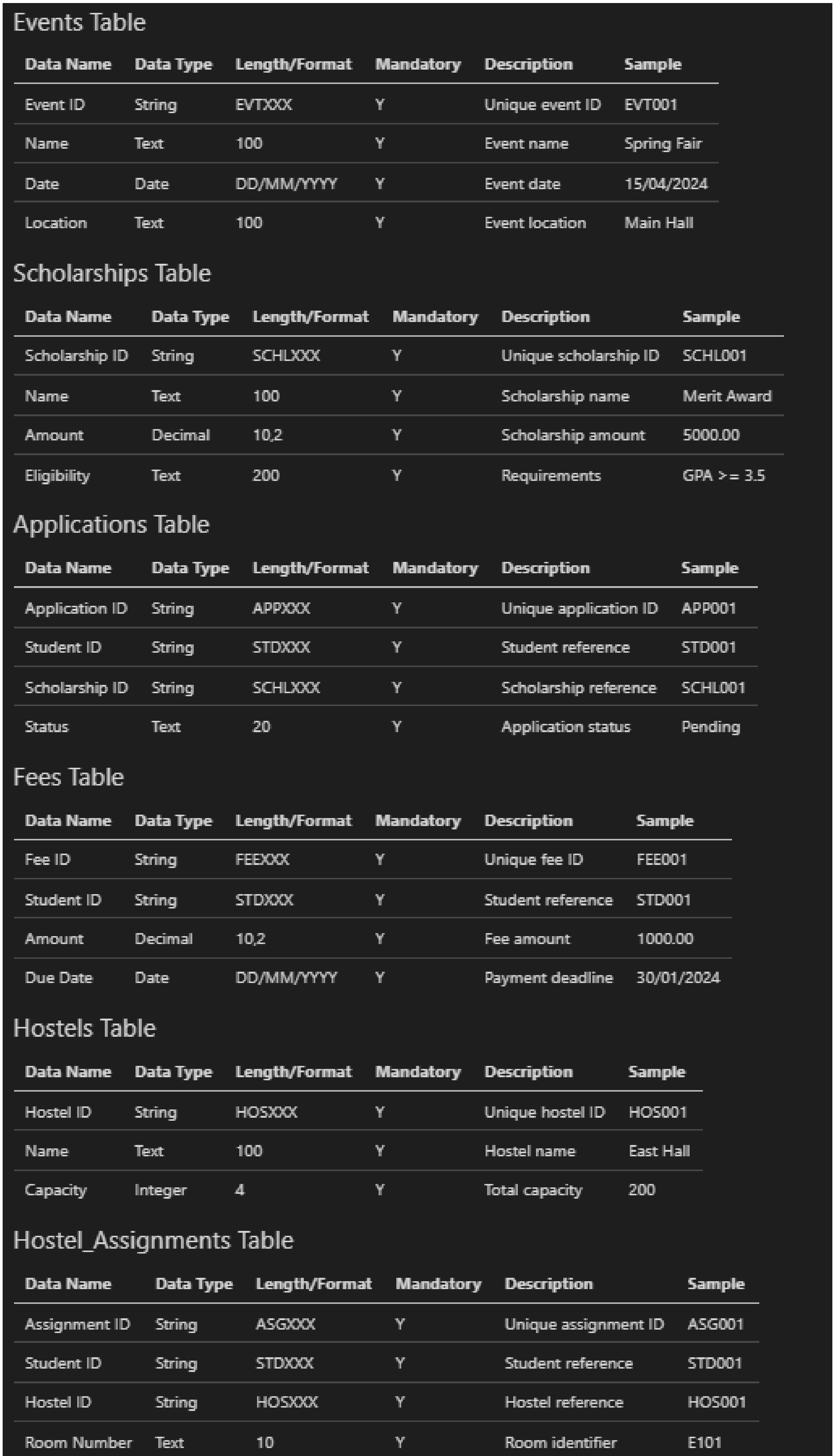
Data Name	Data Type	Length/Format	Mandatory	Description	Sample
Library ID	String	LIBXXX	Y	Unique library ID	LIB001
Name	Text	100	Y	Library name	Main Library
Location	Text	100	Y	Physical location	North Campus

Staff Table

Data Name	Data Type	Length/Format	Mandatory	Description	Sample
Staff ID	String	STFXXX	Y	Unique staff ID	STF001
Name	Text	100	Y	Staff name	Jane Doe
Role	Text	50	Y	Staff position	Administrator
Department ID	String	DEPTXXX	Y	Department reference	DEPT001

Clubs Table

Data Name	Data Type	Length/Format	Mandatory	Description	Sample
Club ID	String	CLBXXX	Y	Unique club ID	CLB001
Name	Text	100	Y	Club name	Chess Club
Advisor ID	String	PROFXXX	Y	Faculty advisor	PROF001



Events Table

Data Name	Data Type	Length/Format	Mandatory	Description	Sample
Event ID	String	EVTXXX	Y	Unique event ID	EVT001
Name	Text	100	Y	Event name	Spring Fair
Date	Date	DD/MM/YYYY	Y	Event date	15/04/2024
Location	Text	100	Y	Event location	Main Hall

Scholarships Table

Data Name	Data Type	Length/Format	Mandatory	Description	Sample
Scholarship ID	String	SCHLXXX	Y	Unique scholarship ID	SCHL001
Name	Text	100	Y	Scholarship name	Merit Award
Amount	Decimal	10,2	Y	Scholarship amount	5000.00
Eligibility	Text	200	Y	Requirements	GPA >= 3.5

Applications Table

Data Name	Data Type	Length/Format	Mandatory	Description	Sample
Application ID	String	APPXXX	Y	Unique application ID	APP001
Student ID	String	STDXXX	Y	Student reference	STD001
Scholarship ID	String	SCHLXXX	Y	Scholarship reference	SCHL001
Status	Text	20	Y	Application status	Pending

Fees Table

Data Name	Data Type	Length/Format	Mandatory	Description	Sample
Fee ID	String	FEEXXX	Y	Unique fee ID	FEE001
Student ID	String	STDXXX	Y	Student reference	STD001
Amount	Decimal	10,2	Y	Fee amount	1000.00
Due Date	Date	DD/MM/YYYY	Y	Payment deadline	30/01/2024

Hostels Table

Data Name	Data Type	Length/Format	Mandatory	Description	Sample
Hostel ID	String	HOSXXX	Y	Unique hostel ID	HOS001
Name	Text	100	Y	Hostel name	East Hall
Capacity	Integer	4	Y	Total capacity	200

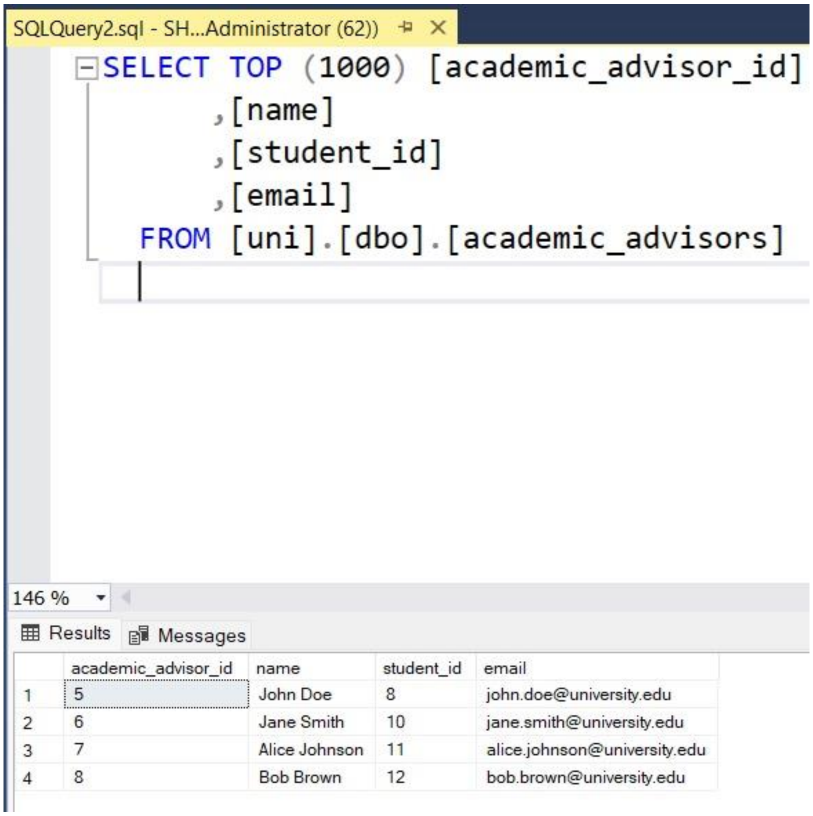
Hostel_Assignments Table

Data Name	Data Type	Length/Format	Mandatory	Description	Sample
Assignment ID	String	ASGXXX	Y	Unique assignment ID	ASG001
Student ID	String	STDXXX	Y	Student reference	STD001
Hostel ID	String	HOSXXX	Y	Hostel reference	HOS001
Room Number	Text	10	Y	Room identifier	E101

DQL / RA

SQL queries and Relational Algebra

1- Advisor Information Retrieval

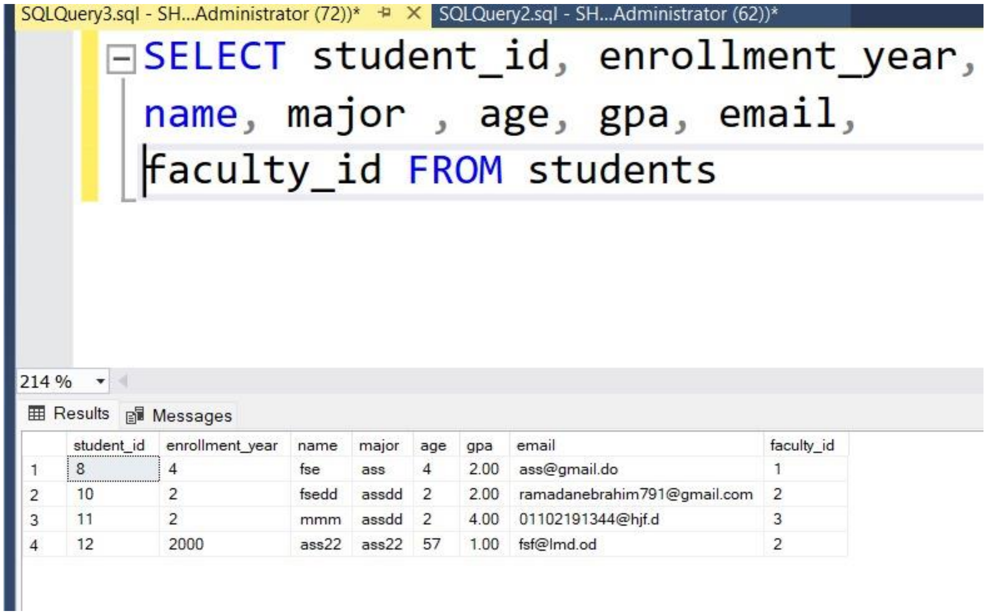


```sql
SELECT TOP (1000) [academic_advisor_id]
      ,[name]
      ,[student_id]
      ,[email]
  FROM [uni].[dbo].[academic_advisors]
```

	academic_advisor_id	name	student_id	email
1	5	John Doe	8	john.doe@university.edu
2	6	Jane Smith	10	jane.smith@university.edu
3	7	Alice Johnson	11	alice.johnson@university.edu
4	8	Bob Brown	12	bob.brown@university.edu

ADVISOR_INFO ← π[academic_advisor_id, name, student_id, email](academic_advisors)

2-Student Information Retrieval



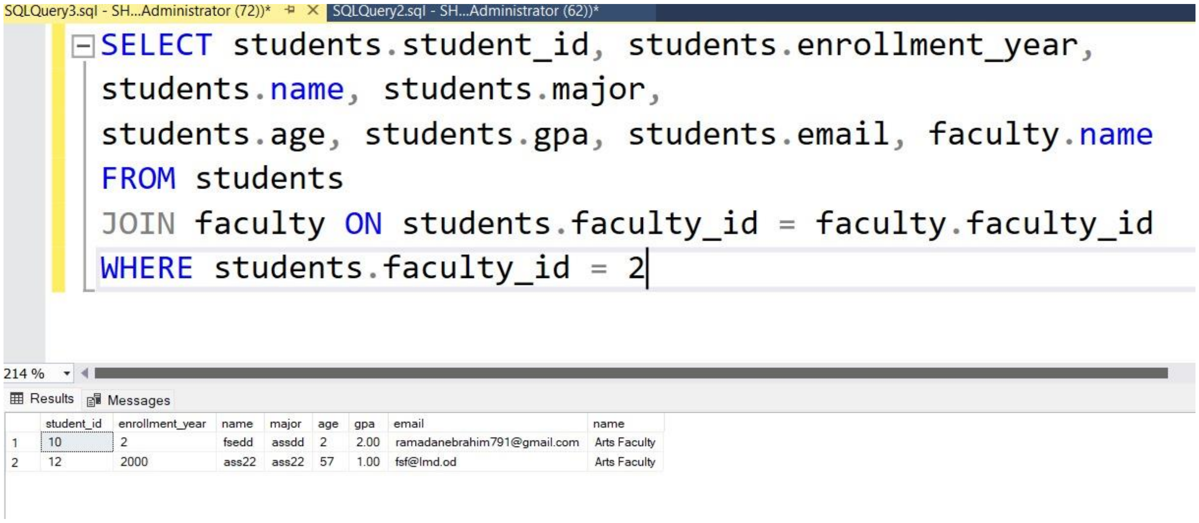
SQLQuery3.sql - SH...Administrator (72))*
SQLQuery2.sql - SH...Administrator (62))*
214 %
Results
Messages

```
SELECT student_id, enrollment_year,
name, major , age, gpa, email,
faculty_id FROM students
```

	student_id	enrollment_year	name	major	age	gpa	email	faculty_id
1	8	4	fse	ass	4	2.00	ass@gmail.do	1
2	10	2	fsedd	assdd	2	2.00	ramadanebrahim791@gmail.com	2
3	11	2	mmm	assdd	2	4.00	01102191344@hjf.d	3
4	12	2000	ass22	ass22	57	1.00	fsf@lmd.od	2

STUDENT_INFO ← π[student_id, enrollment_year, name, major, age, gpa, email, faculty_id](students)

3-Students Information by Specific Faculty (Faculty ID 2)



SQLQuery3.sql - SH...Administrator (72))*
SQLQuery2.sql - SH...Administrator (62))*
214 %
Results
Messages

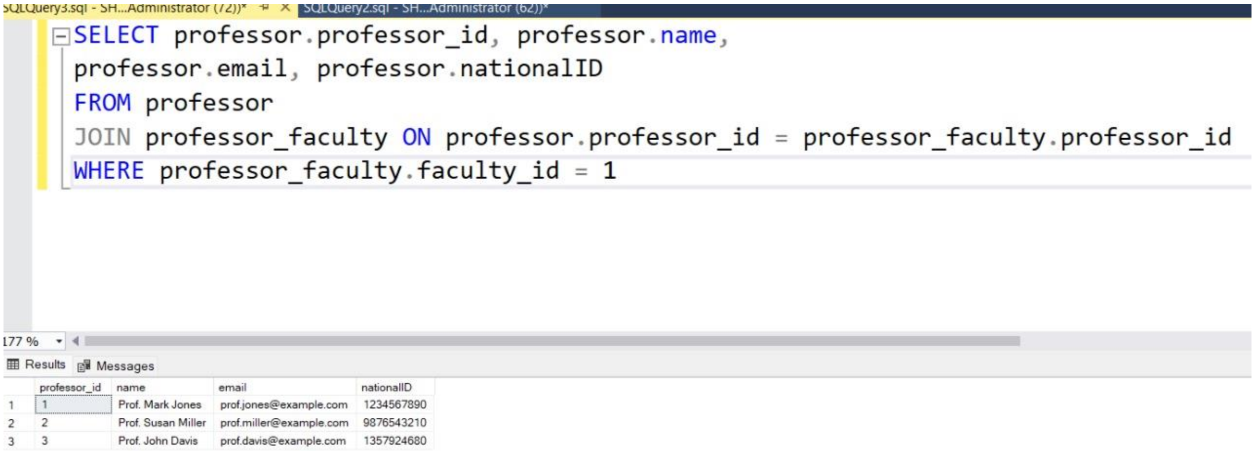
```
SELECT students.student_id, students.enrollment_year,
students.name, students.major,
students.age, students.gpa, students.email, faculty.name
FROM students
JOIN faculty ON students.faculty_id = faculty.faculty_id
WHERE students.faculty_id = 2
```

	student_id	enrollment_year	name	major	age	gpa	email	name
1	10	2	fsedd	assdd	2	2.00	ramadanebrahim791@gmail.com	Arts Faculty
2	12	2000	ass22	ass22	57	1.00	fsf@lmd.od	Arts Faculty

STUDENT_FACULTY ← (students ⋈students.faculty_id=faculty.faculty_id faculty)

RESULT_3 ← π[student_id, enrollment_year, name, major, age, gpa, email, faculty.name](σ[faculty_id=2](STUDENT_FACULTY))

4-Professors Working in Specific Faculty (Faculty ID 1)


SQLQuery3.sql - SH...Administrator (72))*
SQLQuery2.sql - SH...Administrator (62))*
177 %
Results
Messages

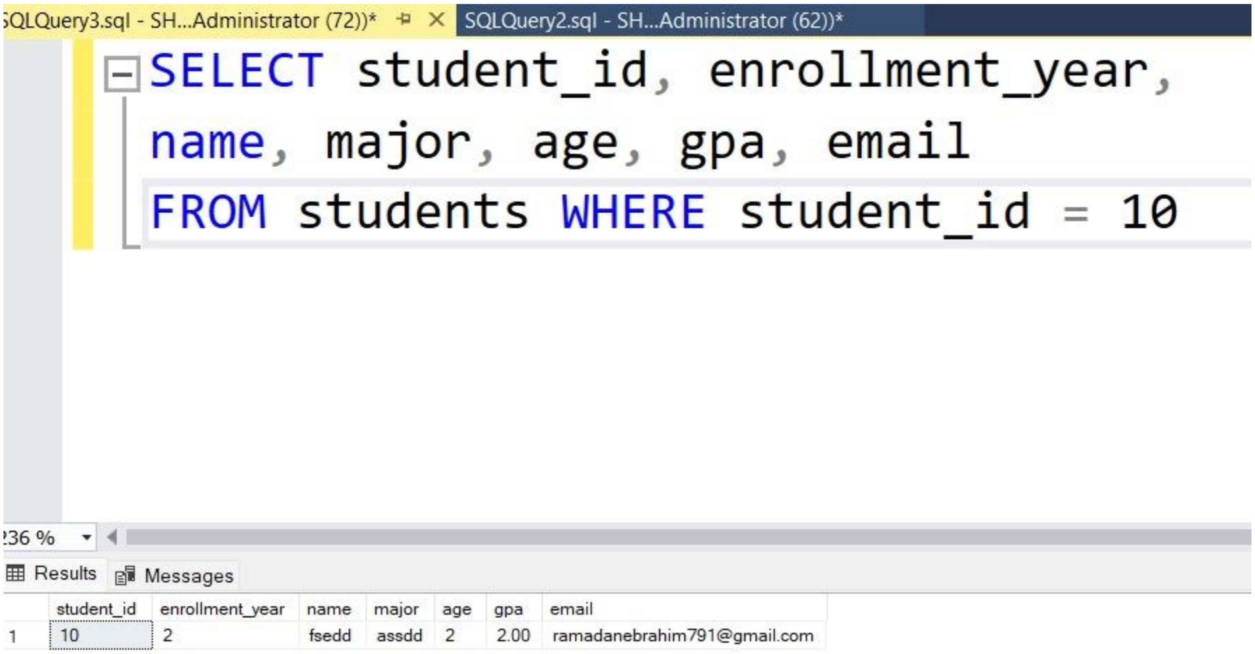
```sql
SELECT professor.professor_id, professor.name,
professor.email, professor.nationalID
FROM professor
JOIN professor_faculty ON professor.professor_id = professor_faculty.professor_id
WHERE professor_faculty.faculty_id = 1
```

	professor_id	name	email	nationalID
1	1	Prof. Mark Jones	prof.jones@example.com	1234567890
2	2	Prof. Susan Miller	prof.miller@example.com	9876543210
3	3	Prof. John Davis	prof.davis@example.com	1357924680

FACULTY_PROFESSORS ← (professor ⋈professor.professor_id=professor_faculty.professor_id professor_faculty)

RESULT_4 ← π[professor_id, name, email, nationalID](σ[faculty_id=1](FACULTY_PROFESSORS))

5-Single Student Record Lookup (Student ID 10)

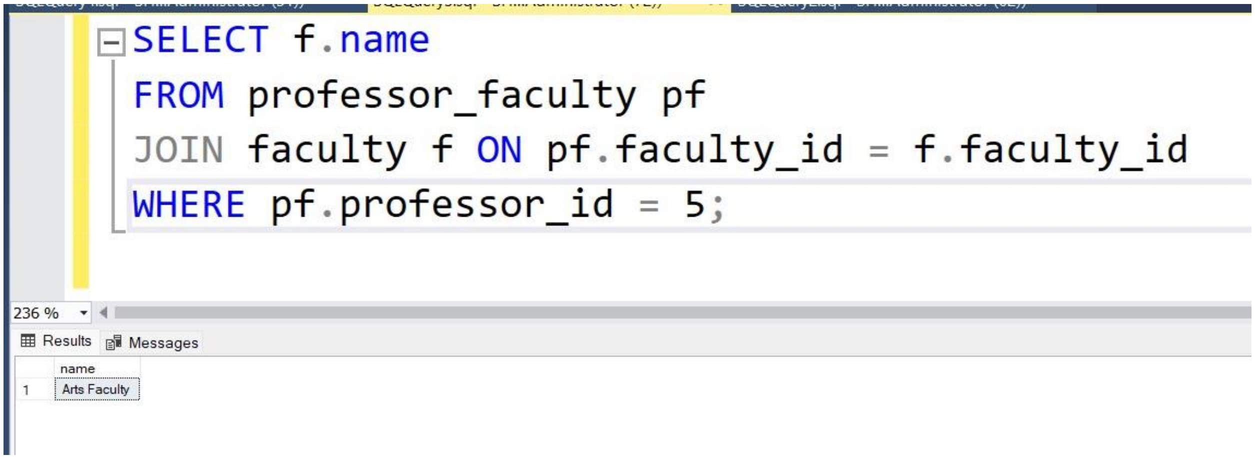

SQLQuery3.sql - SH...Administrator (72))*
SQLQuery2.sql - SH...Administrator (62))*
236 %
Results
Messages

```sql
SELECT student_id, enrollment_year,
name, major, age, gpa, email
FROM students WHERE student_id = 10
```

```
   student_id  enrollment_year  name   major  age  gpa   email
1  10          2                fsedd  assdd  2    2.00  ramadanebrahim791@gmail.com
```

SINGLE_STUDENT ← π[student_id, enrollment_year, name, major, age, gpa, email](σ[student_id=10](students))

6-Faculty Name Lookup for Specific Professor (Professor ID 5)



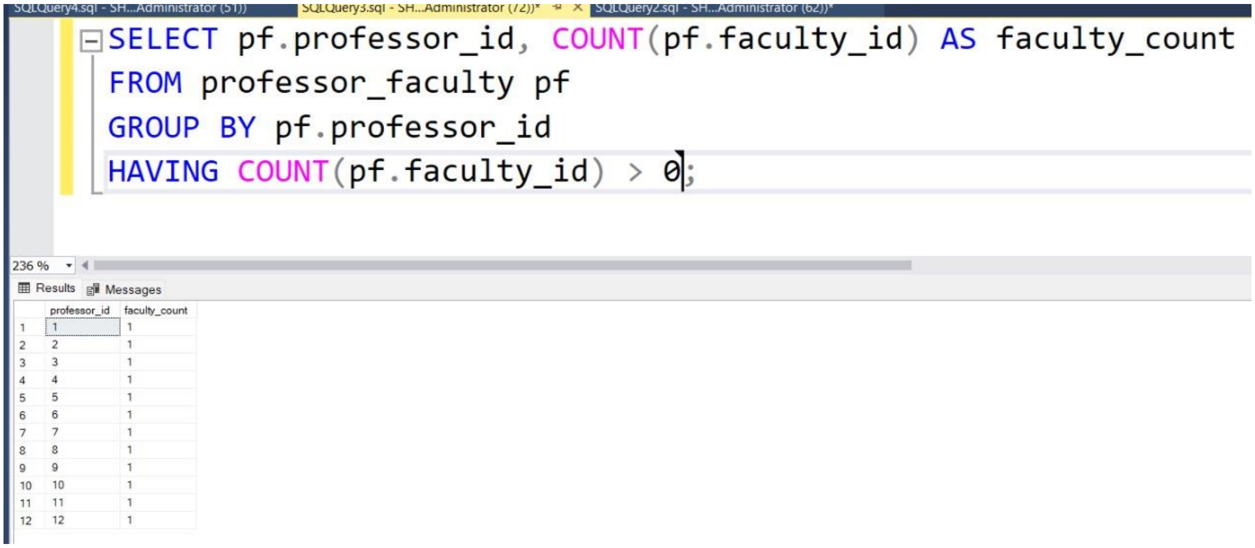
```sql
SELECT f.name
FROM professor_faculty pf
JOIN faculty f ON pf.faculty_id = f.faculty_id
WHERE pf.professor_id = 5;
```

```
   name
1  Arts Faculty
```

PROF_FACULTIES ← (professor_faculty ⋈professor_faculty.faculty_id=faculty.faculty_id faculty)

RESULT_6 ← π[faculty.name](σ[professor_id=5](PROF_FACULTIES))

7-Professor Teaching Load Analysis (Faculty Count per Professor)

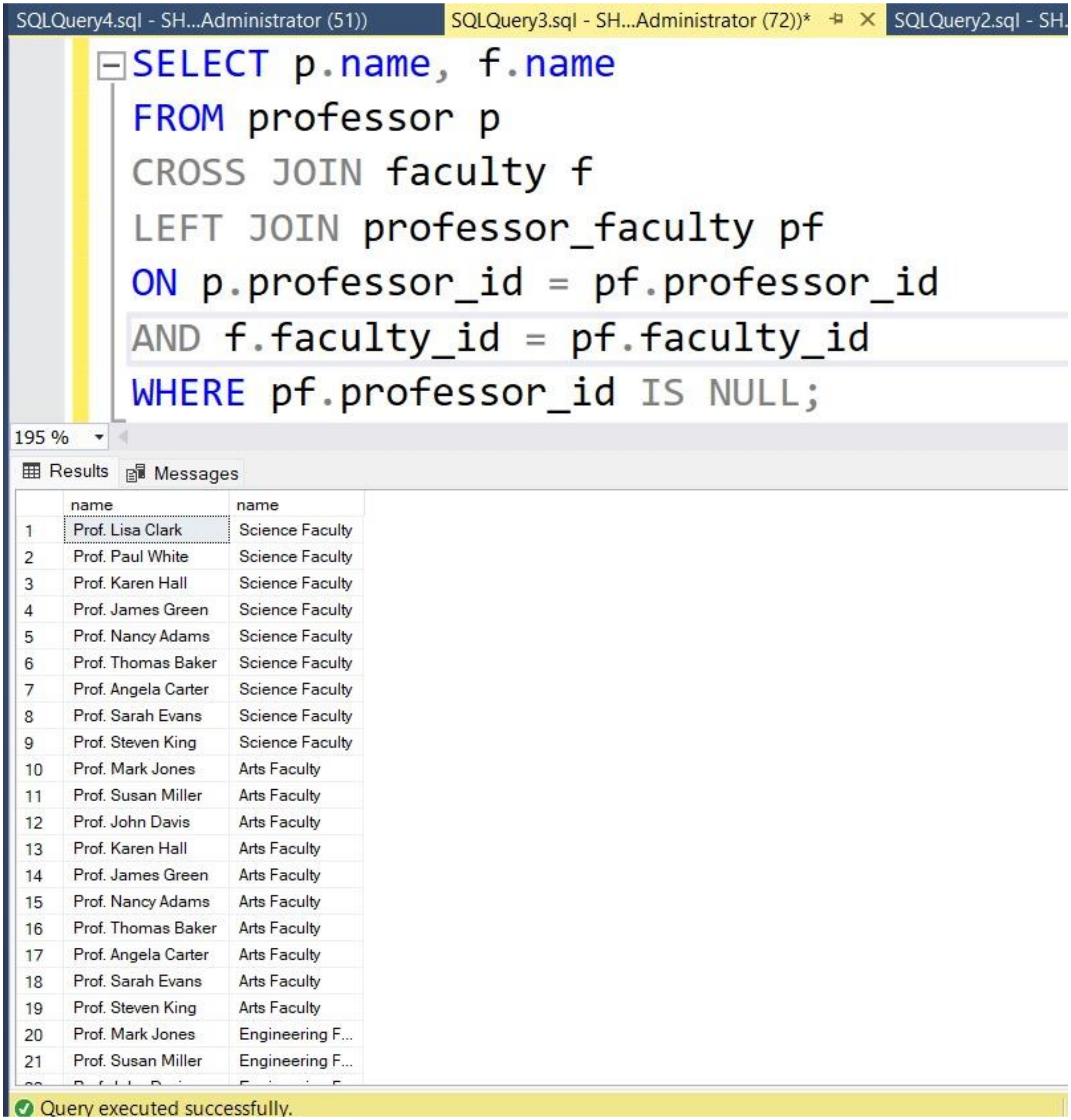

SQLQuery4.sql - SH...Administrator (51))
SQLQuery3.sql - SH...Administrator (72))*
SQLQuery2.sql - SH...Administrator (62))*
236 %
Results
Messages

```sql
SELECT pf.professor_id, COUNT(pf.faculty_id) AS faculty_count
FROM professor_faculty pf
GROUP BY pf.professor_id
HAVING COUNT(pf.faculty_id) > 0;
```

	professor_id	faculty_count
1	1	1
2	2	1
3	3	1
4	4	1
5	5	1
6	6	1
7	7	1
8	8	1
9	9	1
10	10	1
11	11	1
12	12	1

PROF_COUNT ← G[professor_id]COUNT(faculty_id)->faculty_count(σ[COUNT(faculty_id)>0](professor_faculty))

8-Unassigned Professor-Faculty Combinations


SQLQuery4.sql - SH...Administrator (51))
SQLQuery3.sql - SH...Administrator (72))*
SQLQuery2.sql - SH
195 %
Results
Messages

```sql
SELECT p.name, f.name
FROM professor p
CROSS JOIN faculty f
LEFT JOIN professor_faculty pf
ON p.professor_id = pf.professor_id
AND f.faculty_id = pf.faculty_id
WHERE pf.professor_id IS NULL;
```

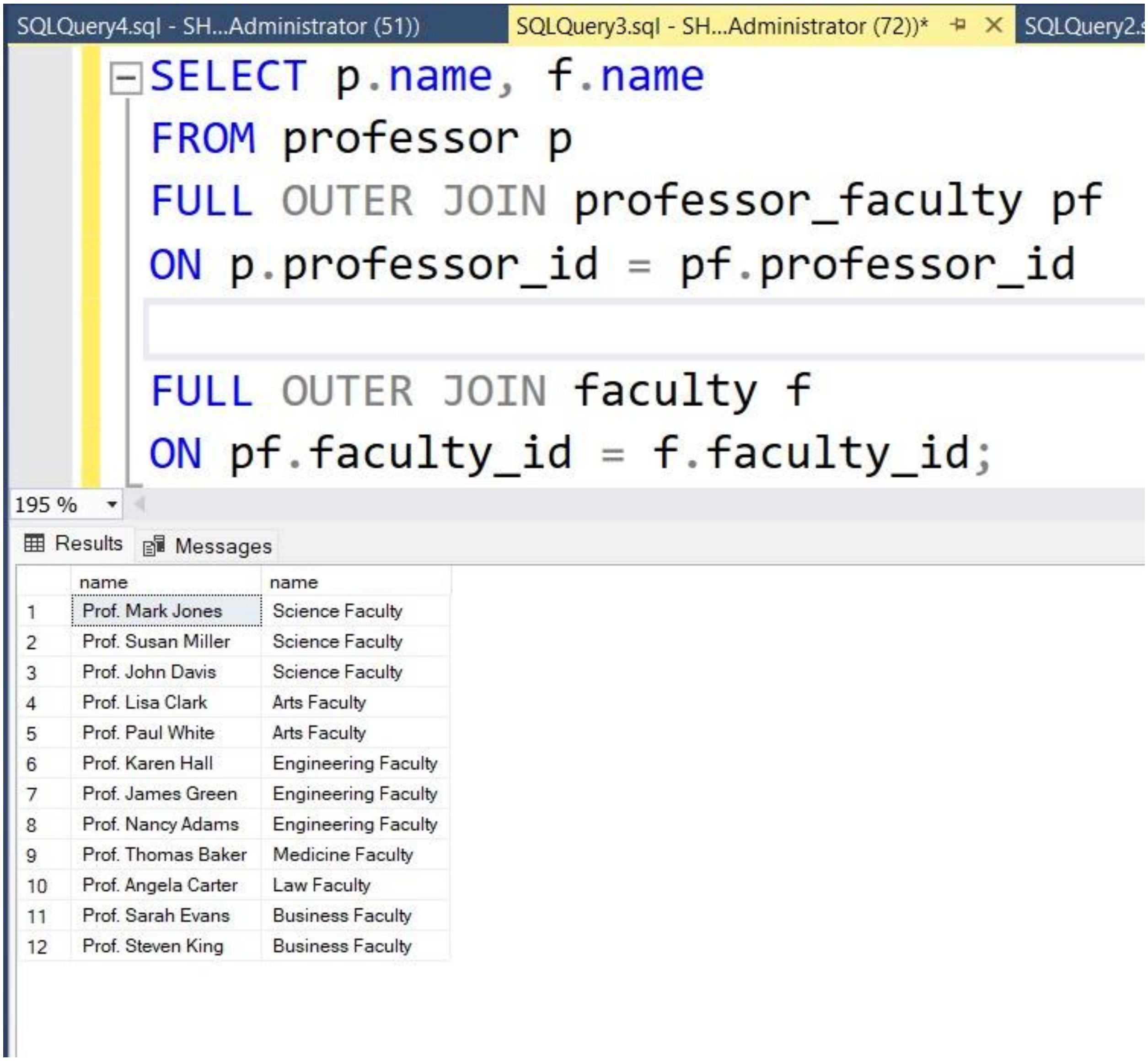
	name	name
1	Prof. Lisa Clark	Science Faculty
2	Prof. Paul White	Science Faculty
3	Prof. Karen Hall	Science Faculty
4	Prof. James Green	Science Faculty
5	Prof. Nancy Adams	Science Faculty
6	Prof. Thomas Baker	Science Faculty
7	Prof. Angela Carter	Science Faculty
8	Prof. Sarah Evans	Science Faculty
9	Prof. Steven King	Science Faculty
10	Prof. Mark Jones	Arts Faculty
11	Prof. Susan Miller	Arts Faculty
12	Prof. John Davis	Arts Faculty
13	Prof. Karen Hall	Arts Faculty
14	Prof. James Green	Arts Faculty
15	Prof. Nancy Adams	Arts Faculty
16	Prof. Thomas Baker	Arts Faculty
17	Prof. Angela Carter	Arts Faculty
18	Prof. Sarah Evans	Arts Faculty
19	Prof. Steven King	Arts Faculty
20	Prof. Mark Jones	Engineering F...
21	Prof. Susan Miller	Engineering F...

Query executed successfully.

ALL_COMBINATIONS ← (professor ⋈p.professor_id=pf.professor_id AND f.faculty_id=pf.faculty_id faculty)

RESULT_8 ← π[professor.name, faculty.name](σ[pf.professor_id IS NULL](ALL_COMBINATIONS))

9-Complete Professor-Faculty Relationship Overview


SQLQuery4.sql - SH...Administrator (51))
SQLQuery3.sql - SH...Administrator (72))*
SQLQuery2.s
195 %
Results
Messages

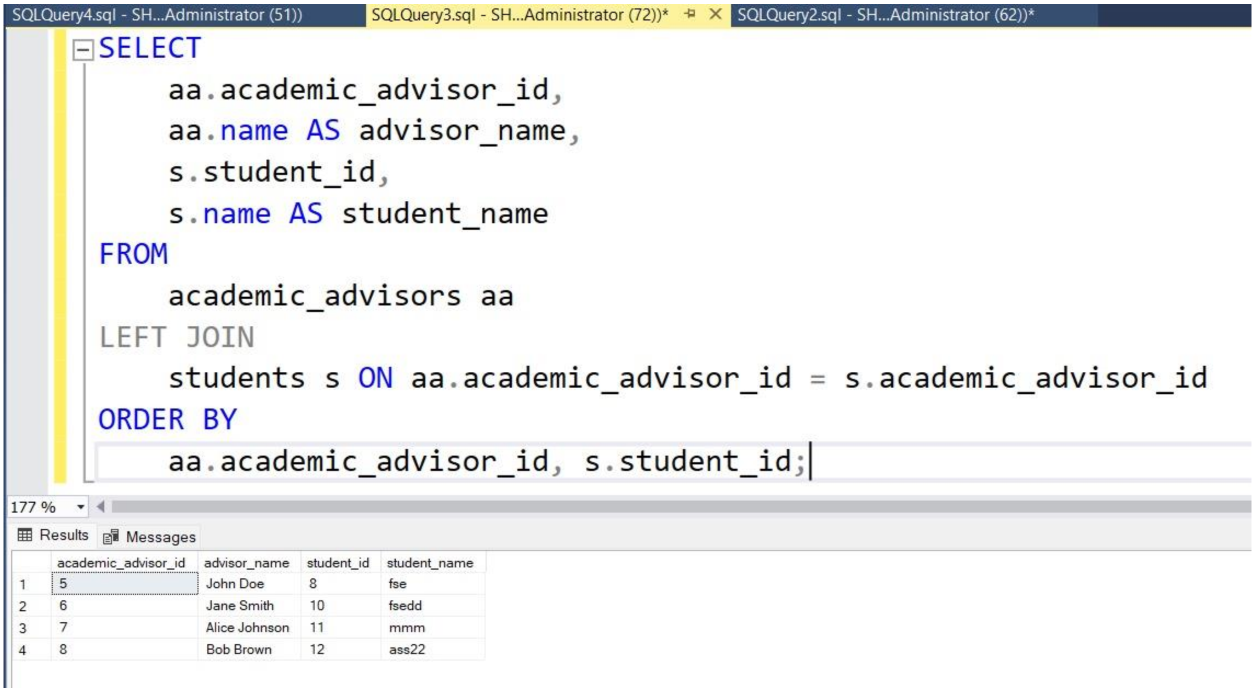
```sql
SELECT p.name, f.name
FROM professor p
FULL OUTER JOIN professor_faculty pf
ON p.professor_id = pf.professor_id

FULL OUTER JOIN faculty f
ON pf.faculty_id = f.faculty_id;
```

	name	name
1	Prof. Mark Jones	Science Faculty
2	Prof. Susan Miller	Science Faculty
3	Prof. John Davis	Science Faculty
4	Prof. Lisa Clark	Arts Faculty
5	Prof. Paul White	Arts Faculty
6	Prof. Karen Hall	Engineering Faculty
7	Prof. James Green	Engineering Faculty
8	Prof. Nancy Adams	Engineering Faculty
9	Prof. Thomas Baker	Medicine Faculty
10	Prof. Angela Carter	Law Faculty
11	Prof. Sarah Evans	Business Faculty
12	Prof. Steven King	Business Faculty

PROF_FAC_FULL ← ((professor ⋈professor.professor_id=professor_faculty.professor_id professor_faculty)⋈professor_faculty.faculty_id=faculty.faculty_id faculty)

RESULT_9 ← π[professor.name, faculty.name](PROF_FAC_FULL)

10-Advisor-Student Assignment List



SQLQuery4.sql - SH...Administrator (51))
SQLQuery3.sql - SH...Administrator (72))*
SQLQuery2.sql - SH...Administrator (62))*
177 %
Results
Messages

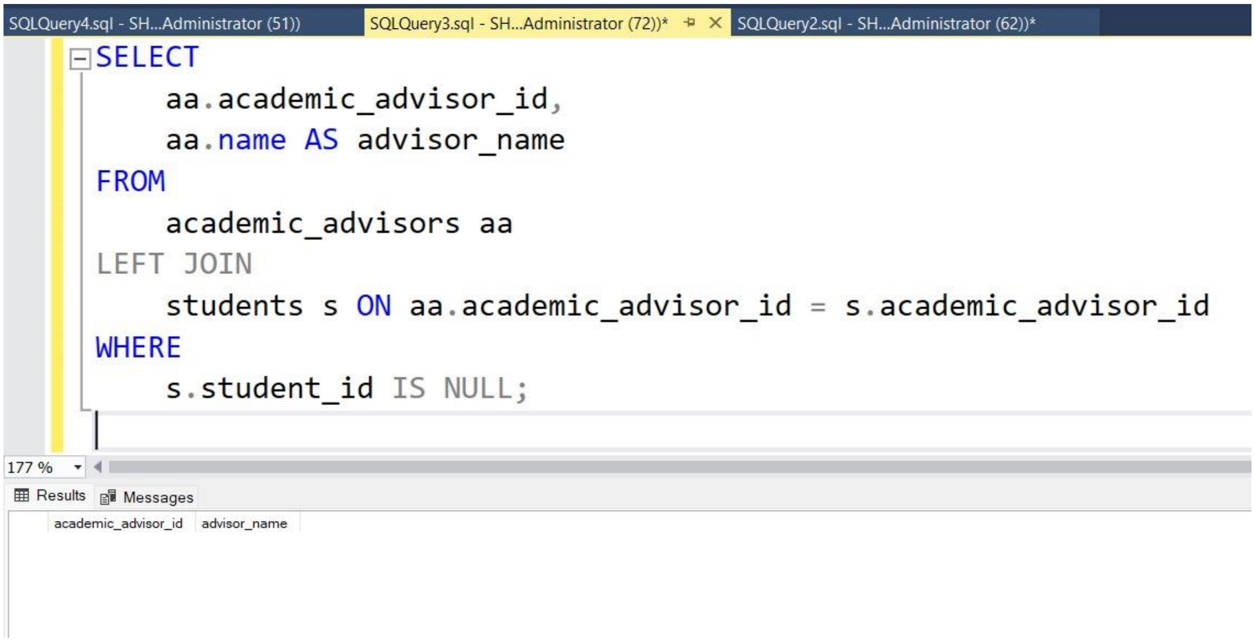
```
SELECT
    aa.academic_advisor_id,
    aa.name AS advisor_name,
    s.student_id,
    s.name AS student_name
FROM
    academic_advisors aa
LEFT JOIN
    students s ON aa.academic_advisor_id = s.academic_advisor_id
ORDER BY
    aa.academic_advisor_id, s.student_id;
```

	academic_advisor_id	advisor_name	student_id	student_name
1	5	John Doe	8	fse
2	6	Jane Smith	10	fsedd
3	7	Alice Johnson	11	mmm
4	8	Bob Brown	12	ass22

ADVISOR_STUDENTS ← (academic_advisors ⋈aa.academic_advisor_id=students.academic_advisor_id students)

RESULT_10 ← π[aa.academic_advisor_id, aa.name->advisor_name, s.student_id, s.name->student_name](ADVISOR_STUDENTS)

11-Advisors Without Assigned Students


SQLQuery4.sql - SH...Administrator (51))
SQLQuery3.sql - SH...Administrator (72))*
SQLQuery2.sql - SH...Administrator (62))*
177 %
Results
Messages

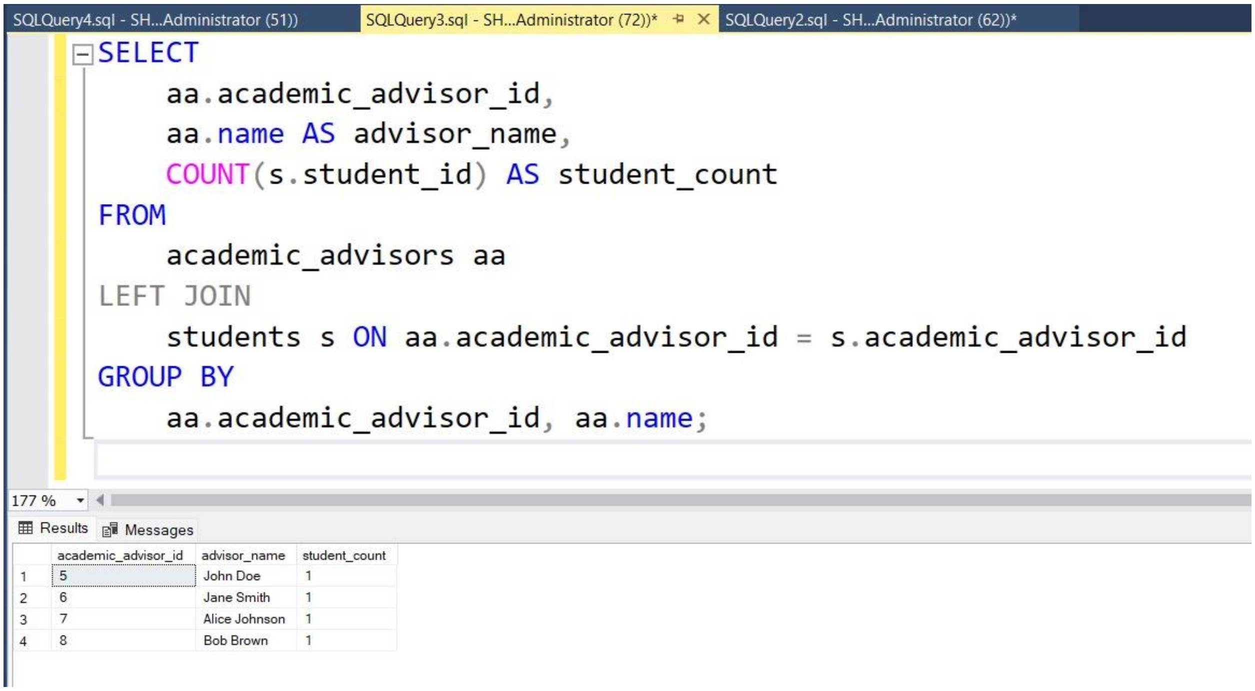
```
SELECT
    aa.academic_advisor_id,
    aa.name AS advisor_name
FROM
    academic_advisors aa
LEFT JOIN
    students s ON aa.academic_advisor_id = s.academic_advisor_id
WHERE
    s.student_id IS NULL;
```

```
academic_advisor_id   advisor_name
```

ADVISORS_NO_STUDENTS ← (academic_advisors ⋈aa.academic_advisor_id=students.academic_advisor_id students)

RESULT_11 ← π[aa.academic_advisor_id, aa.name->advisor_name](σ[s.student_id IS NULL](ADVISORS_NO_STUDENTS))

12-Student Load per Advisor (Count)


SQLQuery4.sql - SH...Administrator (51))
SQLQuery3.sql - SH...Administrator (72))*
SQLQuery2.sql - SH...Administrator (62))*
177 %
Results
Messages

```
SELECT
    aa.academic_advisor_id,
    aa.name AS advisor_name,
    COUNT(s.student_id) AS student_count
FROM
    academic_advisors aa
LEFT JOIN
    students s ON aa.academic_advisor_id = s.academic_advisor_id
GROUP BY
    aa.academic_advisor_id, aa.name;
```

	academic_advisor_id	advisor_name	student_count
1	5	John Doe	1
2	6	Jane Smith	1
3	7	Alice Johnson	1
4	8	Bob Brown	1

ADVISOR_COUNT ← (academic_advisors ⋈aa.academic_advisor_id=students.academic_advisor_id students)

RESULT_12 ← G[aa.academic_advisor_id, aa.name]COUNT(s.student_id)->student_count(ADVISOR_COUNT)