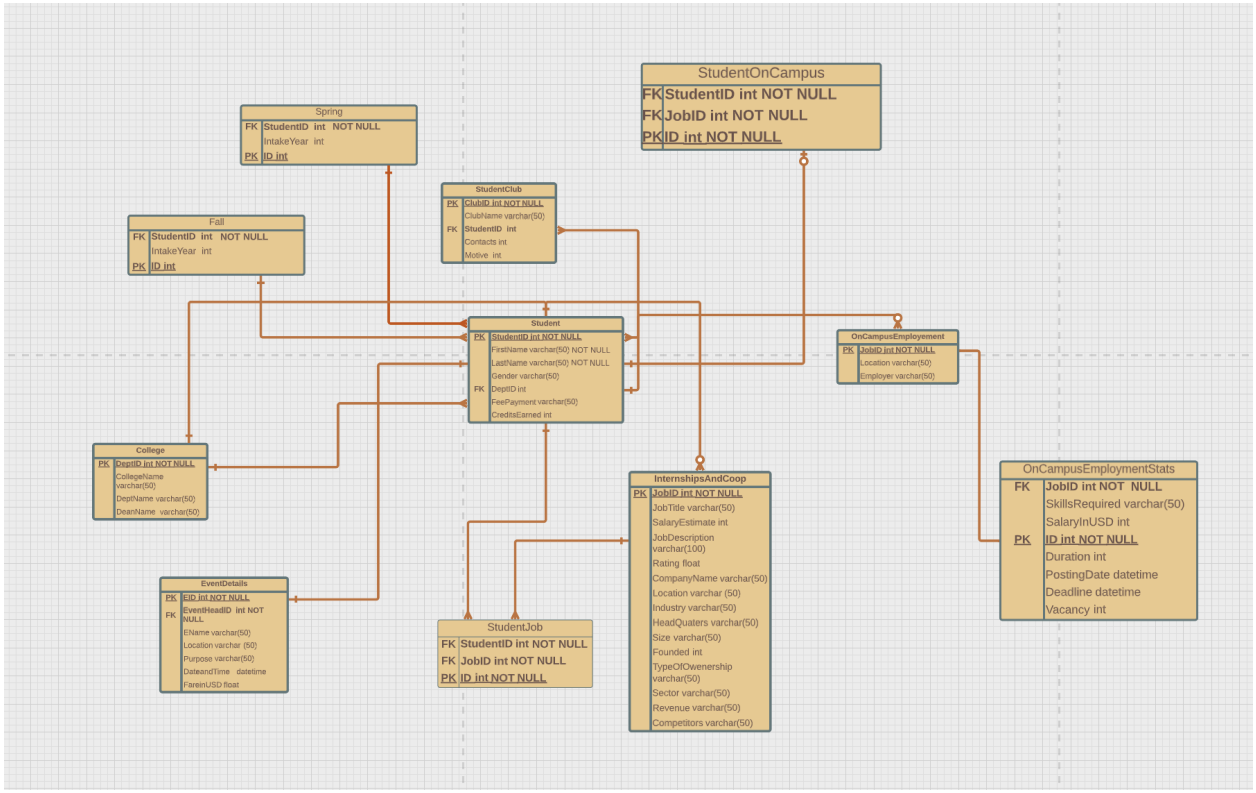
DAMG 6210 Data Management & Database Design
Final Submission

Team Name: MStudents (Team: 13)
Team Members: Renuka Huddar (002750433)
Darshana Rokade (002705752)

1] Project Description

MStudents will help students to monitor their data and gain useful information. The main objective of this project is to keep track of students' profiles like their personal details, school details and the skills that they will gain throughout the course, etc. Students table displays records of students and their profile for eg. their personal details, college information, credits earned, etc. Students can check their eligibility, deadlines, etc and apply for On Campus Employment. In addition to that, students can also apply for Internships and Coop. We will also manage their profile and help them fetch better career opportunities both on-campus and full time jobs. InternshipsAndCoop table is highly useful for students which will help them to retrieve basic details about job posting. Some students are part of many clubs and events. All this data is stored in StudentClub and EventDetails. In addition to that, we will also manage their profile and help them fetch better career opportunities both on-campus and full time jobs. Students can also check the skills that they need to apply for a specific position at some organization. They can also check out the companies that they are interested in working with. Students can check out various events happening in the university and participate in them.

2] ER diagram


StudentOnCampus
FK StudentID int NOT NULL
FK JobID int NOT NULL
PK ID int NOT NULL
Spring
FK StudentID int NOT NULL
IntakeYear int
PK ID int
StudentClub
PK ClubID int NOT NULL
ClubName varchar(50)
FK StudentID int
Contacts int
Motive int
Fall
FK StudentID int NOT NULL
IntakeYear int
PK ID int
Student
PK StudentID int NOT NULL
FirstName varchar(50) NOT NULL
LastName varchar(50) NOT NULL
Gender varchar(50)
FK DeptID int
FeePayment varchar(50)
CreditsEarned int
OnCampusEmployement
PK JobID int NOT NULL
Location varchar(50)
Employer varchar(50)
College
PK DeptID int NOT NULL
CollegeName varchar(50)
DeptName varchar(50)
DeanName varchar(50)
InternshipsAndCoop
PK JobID int NOT NULL
JobTitle varchar(50)
SalaryEstimate int
JobDescription varchar(100)
Rating float
CompanyName varchar(50)
Location varchar (50)
Industry varchar(50)
HeadQuaters varchar(50)
Size varchar(50)
Founded int
TypeOfOwenership varchar(50)
Sector varchar(50)
Revenue varchar(50)
Competitors varchar(50)
OnCampusEmploymentStats
FK JobID int NOT NULL
SkillsRequired varchar(50)
SalaryInUSD int
PK ID int NOT NULL
Duration int
PostingDate datetime
Deadline datetime
Vacancy int
EventDetails
PK EID int NOT NULL
FK EventHeadID int NOT NULL
EName varchar(50)
Location varchar (50)
Purpose varchar(50)
DateandTime datetime
FareinUSD float
StudentJob
FK StudentID int NOT NULL
FK JobID int NOT NULL
PK ID int NOT NULL

3] Snippets from Database- sample tables, output of views

Table 1 : Student

Schema:

	Field	Type	Null	Key	Default	Extra
▶	StudentID	int	NO	PRI	NULL	
	FirstName	varchar(50)	YES		NULL	
	LastName	varchar(50)	YES		NULL	
	Gender	varchar(5)	YES		NULL	
	DeptID	int	NO	MUL	NULL	
	FeePayment	varchar(5)	YES		NULL	
	CreditsEarned	int	YES		NULL	

Sample Data:

	StudentID	FirstName	LastName	Gender	DeptID	FeePayment	CreditsEarned
▶	101	Antwan	Davies	M	207	No	16
	102	Keshawn	Frey	F	203	No	16
	103	Isla	Andrade	M	218	Yes	24
	104	Leilani	Wilkerson	F	205	Yes	17
	105	Ivan	Ochoa	F	219	Yes	16
	106	Emilia	Guerrero	F	202	No	16
	107	John	zhho	M	220	No	16
	108	Mark	Orozco	F	204	Yes	16

student 81 ×

Table 2 : Fall

Schema:

	Field	Type	Null	Key	Default	Extra
▶	StudentID	int	NO	MUL	NULL	
	IntakeYear	int	NO		NULL	
	ID	int	NO	PRI	NULL	

Sample Data:

	StudentID	IntakeYear	ID
▶	101	2021	1
	102	2021	2
	106	2021	3
	111	2021	4
	113	2021	5
	115	2021	6
	116	2021	7
	124	2021	8
	125	2021	9
	126	2021	10

Fall 64 ×

Table 3 : Spring

Schema:

	Field	Type	Null	Key	Default	Extra
▶	StudentID	int	NO	MUL	NULL	
	IntakeYear	int	NO		NULL	
	ID	int	NO	PRI	NULL	

Sample Data:

StudentID	IntakeYear	ID
103	2021	1
104	2022	2
105	2022	3
107	2022	4
108	2022	5
109	2022	6
110	2022	7
112	2021	8
114	2022	9
117	2022	10

Spring 66 ×

Table 4 : StudentJob

Schema:

Field	Type	Null	Key	Default	Extra
StudentID	int	NO	UNI	NULL	
JobID	int	NO		NULL	
ID	int	NO	PRI	NULL	

Sample Data:

StudentID	JobID	ID
101	2084	1
102	2591	2
103	1370	3
104	690	4
105	1775	5
106	1283	6
107	2642	7
108	1141	8
109	1270	9

Table 5 : StudentOnCampus

Schema:

Field	Type	Null	Key	Default	Extra
StudentID	int	NO	MUL	NULL	
JobID	int	YES		NULL	
ID	int	NO	PRI	NULL	

Sample Data:

StudentID	JobID	ID
108	324	1
112	319	2
113	314	3
115	321	4
119	318	5
126	314	6
128	309	7
131	303	8
133	312	9

StudentOnCampus 63 ×

Table 6 : OnCampusEmployment

Schema:

Field	Type	Null	Key	Default	Extra
JobID	int	NO	PRI	NULL	
Employer	varchar(100)	NO		NULL	
Location	varchar(100)	NO		NULL	

Sample Data:

	JobID	Employer	Location
▶	301	Dunkin Donuts (Shillman Hall)	Shillman Hall
	302	Dunkin Donuts (Richards/Hayden Hall)	Richards/Hayden Hall
	303	Caffe Strega (ISEC)	ISEC
	304	Chartwells	Northeastern
	305	Reprographics (Ell Hall)	Ell Hall
	306	Wollastons	Northeastern
	307	Vanguard (Columbus Parking Garage)	Columbus Parking Garage
	308	B.Good (Marino Center)	Marino Center
	309	Cafe 716	Northeastern

OnCampusEmployment 66 ×

Table 7 : OnCampusEmploymentStats

Schema:

	Field	Type	Null	Key	Default	Extra
▶	ID	int	NO	PRI	NULL	
	JobID	int	NO	MUL	NULL	
	SkillsRequired	varchar(100)	NO		NULL	
	SalaryInUSD	int	NO		NULL	
	DurationInMonths	varchar(100)	YES		NULL	
	PostingDate	datetime	YES		NULL	
	Deadline	datetime	YES		NULL	
	Vacancy	int	YES		NULL	

Sample Data:

ID	JobID	SkillsRequired	SalaryInUSD	DurationInMonths	PostingDate	Deadline	Vacancy
1	301	Fluency In English	10	4	2022-11-01 20:30:00	2022-12-01 20:30:00	2
2	302	Fluency In English	10	4	2022-11-02 20:30:00	2022-12-02 20:30:00	4
3	303	Fluency In English	12	6	2022-11-03 20:30:00	2022-12-03 20:30:00	8
4	304	Fluency In English	11	4	2022-11-04 20:30:00	2022-12-04 20:30:00	12
5	305	Fluency In English	13	4	2022-11-05 20:30:00	2022-12-05 20:30:00	15
6	306	Fluency In English	12	4	2022-11-06 20:30:00	2022-12-06 20:30:00	10
7	307	Fluency In English	13	6	2022-11-07 20:30:00	2022-12-07 20:30:00	4
8	308	Fluency In English	12	4	2022-11-08 20:30:00	2022-12-08 20:30:00	5
9	309	Fluency In English	10	4	2022-11-09 20:30:00	2022-12-09 20:30:00	6

OnCampusEmploymentStats 73 ×

Table 8 : College

Schema:

Field	Type	Null	Key	Default	Extra
CollegeName	varchar(100)	NO		NULL	
DeptName	varchar(100)	NO		NULL	
DeptID	int	NO	PRI	NULL	
DeanName	varchar(100)	NO		NULL	

Sample Data:

CollegeName	DeptName	DeptID	DeanName
Bouvé College of Health Sciences	Applied Behavior Analysis (CAGS, MS, Grad Cer...	201	Carmen Sceppa
Bouvé College of Health Sciences	Exercise Science Certificate for Clinicians	202	Carmen Sceppa
Bouvé College of Health Sciences	Health Informatics	203	Carmen Sceppa
Bouvé College of Health Sciences	Human Movement and Rehabilitation Sciences	204	Carmen Sceppa
College of Arts, Media and Design	MS in Information Design and Data Visualization	205	Elizabeth Hudson
College of Arts, Media and Design	MS in Game Science and Design	206	Elizabeth Hudson
College of Engineering	Information Systems	207	Gregory D. Abowd
College of Engineering	Pharmaceutical Engineering	208	Gregory D. Abowd
College of Engineering	Telecommunication Networks	209	Gregory D. Abowd

Table 9 : InternshipsAndCoop

Schema:

Field	Type	Null	Key	Default	Extra
JobID	int	NO	PRI	NULL	
JobTitle	varchar(500)	YES		NULL	
SalaryEstimateLower	int	YES		NULL	
SalaryEstimateUpper	int	YES		NULL	
JobDescription	varchar(1000)	YES		NULL	
Rating	float	YES		NULL	
CompanyName	varchar(500)	YES		NULL	
Location	varchar(500)	YES		NULL	
HeadQuaters	varchar(500)	YES		NULL	

Sample Data:

JobID	JobTitle	SalaryEstimateLower	SalaryEstimateUpper	JobDescription	Rating	C
401	Data Analyst, Center on Immigration and Justic...	37000	66000	Are you eager to roll up your sleeves and harne...	3.2	V
402	Quality Data Analyst	37000	66000	Overview Provides analytical and technical sup...	3.8	Vi
403	Senior Data Analyst, Insights & Analytics Team ...	37000	66000	We���re looking for a Senior Data Analyst w...	3.4	S
404	Data Analyst	37000	66000	Requisition NumberRR-0001939 Remote:Yes W...	4.1	C
405	Reporting Data Analyst	37000	66000	ABOUT FANDUEL GROUP FanDuel Group is a w...	3.9	F
406	Data Analyst	37000	66000	About Cubist Cubist Systematic Strategies is on...	3.9	P
407	Business/Data Analyst (FP&A)	37000	66000	Two Sigma is a different kind of investment man...	4.4	T
408	Data Science Analyst	37000	66000	Data Science Analyst Job Details Level Experie...	3.7	G

Table 10 : EventDetails

Schema:

Field	Type	Null	Key	Default	Extra
EID	int	NO	PRI	NULL	
Ename	varchar(100)	YES		NULL	
EventHeadID	int	NO	MUL	NULL	
Location	varchar(50)	YES		NULL	
Purpose	varchar(1000)	YES		NULL	
Dateandtime	datetime	YES		NULL	
FareinUSD	int	YES		NULL	

Sample Data:

EID	Ename	EventHeadID	Location	Purpose	Dateandtime	FareinUSD
501	Northeastern University Professor on Marketing...	756	Hayden Hall	Northeastern University Distinguished Professor...	2022-11-09 20:30:00	20
502	Carry the Love: Northeastern University	456	Snell Engg center	Uniting students on campus for two days of wor...	2022-02-10 19:00:00	80
503	Northeastern State University- College Jazz Night	1000	Snell Library	NSU College Jazz Combos night at LowDown!	2022-11-19 14:00:00	0
504	Chinese Language and Culture Seminar	235	Online	Learn more about Chinese culture, Mandarin, a...	2022-11-11 23:00:00	0
505	Demo Day Fall 2022	456	Ruggles	Demo Day is Northeastern University's largest c...	2022-11-09 20:30:00	0
506	College Admission 101	214	Tremont Street	This presentation will help families separate coll...	2022-12-07 20:00:00	0
507	International Education Week Kick-Off Event	1023	West Village	Drop by for some snacks and to find out more a...	2022-11-17 12:45:00	0
508	Next Steps for Critical Infrastructure & Cyber S...	1257	Church Hill	Next Steps for Critical Infrastructure & Cyber S...	2022-11-26 10:00:00	60
509	The Promise and Progress Towards A Green Ne...	689	Ell Hall	Join us for an informative discussion on the pro...	2022-11-18 11:00:00	23

Table 11 : StudentClubs

Schema:

	Field	Type	Null	Key	Default	Extra
▶	ClubID	int	NO	PRI	NULL	
	ClubName	varchar(100)	YES		NULL	
	StudentID	int	NO	MUL	NULL	
	Contacts	varchar(100)	YES		NULL	
	Motive	varchar(1000)	YES		NULL	

Sample Data:

	ClubID	ClubName	StudentID	Contacts	Motive
▶	601	Association for Computing Machinery	107	cbw@ccs.neu.edu	engage informative speakers, public forums, an...
	602	CS Student Association	128	khoury-prospect@northeastern.edu	connect students at Northeastern through the ...
	603	Code 4 Culture	130	c4cneu@gmail.com	social and networking club for Black and Latinx s...
	604	Code4Community	134	c4cneu@gmail.com	student organization that partners with non-pro...
	605	CS Mentoring Organisation	145	b.hescott@northeastern.edu	connects students across the discipline of comp...
	606	DATA Club	154	khoury@northeastern.edu	The DATA Club provides speaker events, works...
	607	FirstBYTE	241	e.strange@northeastern.edu	provides affordable and user-friendly resource...
	608	American Institute of Chemical Engineers	273	nuaiche@gmail.com	focuses around enriching the education of chem...
	609	Alliance for Diversity in Science and Engineering	284	northeasternadse@gmail.com	organizations that reach out to students and sci...

4] Use cases & output snippets

#1 Display count of students in each department

```
CREATE VIEW StudentCount AS
SELECT c.DeptName, Count(s.StudentID) FROM Student s INNER JOIN College c ON
s.DeptID=c.DeptID
GROUP BY c.DeptID ORDER BY Count(s.StudentID);
SELECT * from StudentCount;
```

Output:

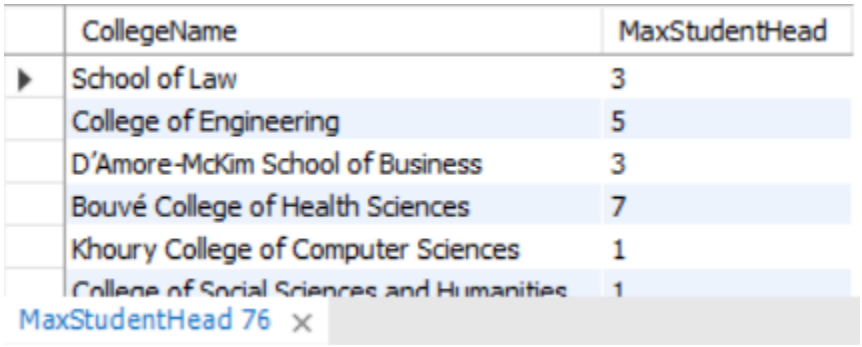
	DeptName	Count(s.StudentID)
▶	MS in Game Science and Design	43
	Human Factors	43
	Industrial Engineering	43
	Criminal Justice	43
	Economics	43
	ID	43

StudentCount 75

#2 Which college has more number of student head?

```
CREATE VIEW MaxStudentHead AS
Select c.CollegeName, Count(c.CollegeName) AS MaxStudentHead FROM College c
INNER JOIN Student s ON c.DeptID=s.DeptID
INNER JOIN StudentClub sc ON s.StudentID=sc.StudentID GROUP BY c.CollegeName;
SELECT * FROM MaxStudentHead;
# Example for question 2
SELECT c.CollegeName, s.StudentID FROM Student s INNER JOIN StudentClub sc ON
s.StudentID=sc.StudentID WHERE s.College="School of Law";
```

Output:



	CollegeName	MaxStudentHead
▶	School of Law	3
	College of Engineering	5
	D'Amore-McKim School of Business	3
	Bouvé College of Health Sciences	7
	Khoury College of Computer Sciences	1
	College of Social Sciences and Humanities	1

MaxStudentHead 76 ×

#3 Number of students who got coop

```
CREATE VIEW StudentsOnCoop AS
SELECT sj.JobID, COUNT(s.StudentID) AS NumberOfStudents FROM Student s INNER
JOIN StudentJob sj ON s.StudentID=sj.StudentID
GROUP BY sj.JobID ORDER BY JobID;
SELECT * from StudentsOnCoop;
```

Output:

	JobID	NumberOfStudents
▶	401	2
	402	3
	404	2
	413	2
	414	1
	415	1

StudentsOnCoop 77 ×

#4 Which department got maximum placements?

```
CREATE VIEW MaxPlacements AS
SELECT s.DeptID, c.DeptName, COUNT(sj.JobID) AS NumberOfStudents FROM College c
INNER JOIN Student s ON c.DeptID=s.DeptID
INNER JOIN StudentJob sj ON s.StudentID=sj.StudentID
GROUP BY s.DeptID ORDER BY COUNT(sj.JobID) DESC;
SELECT * FROM MaxPlacements;
```

Output:

	DeptID	DeptName	NumberOfStudents
▶	216	Medicinal Chemistry and Drug Discovery	146
	203	Health Informatics	111
	208	Pharmaceutical Engineering	109
	213	Electrical & Computer Engineering Leadership	109
	202	Exercise Science Certificate for Clinicians	74
	204	Human Movement and Rehabilitation Sciences	74

MaxPlacements 78 ×

#5 Average salary offered to students in each department

```
CREATE VIEW AvgSalary AS
SELECT c.DeptID, c.DeptName, AVG(i.SalaryEstimateUpper) AS Average FROM College c
INNER JOIN Student s ON c.DeptID=s.DeptID
INNER JOIN StudentJob sj ON s.StudentID = sj.StudentID
INNER JOIN InternshipsAndCoop i on sj.JobID = i.JobID GROUP BY c.deptid, c.deptName;
SELECT * FROM AvgSalary;
```

Output:

	DeptID	DeptName	Average
▶	201	Applied Behavior Analysis (CAGS, MS, Grad Cer...	98114.2857
	202	Exercise Science Certificate for Clinicians	93746.2687
	203	Health Informatics	88915.0943
	204	Human Movement and Rehabilitation Sciences	84294.1176
	205	MS in Information Design and Data Visualization	93407.4074
	206	MS in Game Science and Design	95628.5714

AvgSalary 79 ×

#6 Maximum number of JobOpenings

```
CREATE VIEW MaxJobOpening AS
SELECT JobID, JobTitle, MAX(mycount) AS Maximum_Number_Of_Job_Openings FROM
(SELECT COUNT(JobTitle) mycount, JobTitle, JobID FROM InternshipsAndCoop GROUP
BY JobTitle) as sample;
SELECT * FROM MaxJobOpening;
```

Output:

	JobID	JobTitle	Maximum_Number_Of_Job_Openings
▶	401	Data Analyst, Center on Immigration and Justic...	379

#7 Job Count for each intake

```
-- Count for Fall Intake
CREATE VIEW FallIntakeCount AS
SELECT COUNT(sj.StudentID) AS CountOfStudents, f.IntakeYear FROM Student s INNER
JOIN StudentJob sj ON s.StudentID=sj.StudentID
INNER JOIN Fall f ON s.StudentID=f.StudentID GROUP BY f.IntakeYear;
SELECT * FROM FallIntakeCount;
```

Output:

	CountOfStudents	IntakeYear
▶	325	2021
	451	2022

```
-- Count for Spring Intake
CREATE VIEW SpringIntakeCount AS
```

```
SELECT COUNT(sj.StudentID) AS CountOfStudents, sp.IntakeYear FROM Student s
INNER JOIN StudentJob sj ON s.StudentID=sj.StudentID
INNER JOIN Spring sp ON s.StudentID=sp.StudentID GROUP BY sp.IntakeYear;
SELECT * FROM SpringIntakeCount;
```

OutPut:

	CountOfStudents	IntakeYear
▶	305	2021
	192	2022

#8 Booming Industry

```
CREATE VIEW BoomingIndustry AS
SELECT ic.Industry, COUNT(ic.Industry) BoomingIndustry FROM InternshipsAndCoop ic
INNER JOIN StudentJob sj ON ic.JobID=sj.JobID
GROUP BY ic.Industry ORDER BY COUNT(ic.Industry) DESC;
SELECT * FROM BoomingIndustry;
```

Output:

	Industry	BoomingIndustry
▶	Staffing & Outsourcing	195
	IT Services	180
	Unknown	103
	Health Care Services & Hospitals	82
	Computer Hardware & Software	60
	Consulting	52

BoomingIndustry 83 ×

#9 Which Second Highest Salary

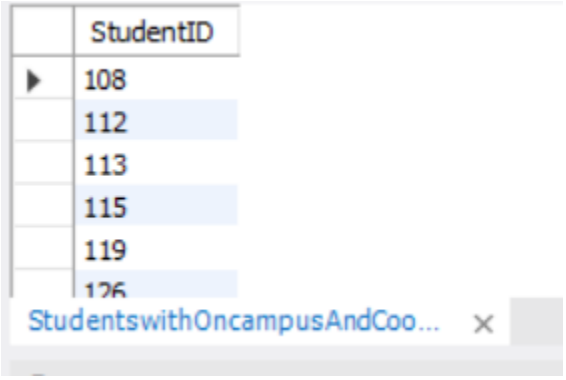
```
CREATE VIEW SecongHighestSalary AS
SELECT MAX(SalaryEstimateUpper) AS Second_Highest_Salary from InternshipsAndCoop
WHERE SalaryEstimateUpper < (SELECT Max(SalaryEstimateUpper) FROM
InternshipsAndCoop);
SELECT * FROM SecongHighestSalary;
```

Output:

	Second_Highest_Salary
▶	178000

#10 Students who got internship and oncampus
CREATE VIEW StudentswithOncampusAndCoop AS
SELECT sj.StudentID FROM StudentOnCampus soc INNER JOIN StudentJob sj ON soc.StudentID=sj.StudentID;
SELECT * FROM StudentswithOncampusAndCoop;

Output:



5] SQL Scripts

```
CREATE DATABASE mastersstudents;
use mastersstudents;

SHOW DATABASES;

ALTER TABLE OnCampusEmployment DROP COLUMN SkillsRequired, DROP COLUMN
SalaryInUSD, Drop COLUMN DurationInMonths, DROP COLUMN PostingDate, DROP
COLUMN Deadline, DROP COLUMN Vacancy;

CREATE TABLE OnCampusEmploymentStats (ID INT NOT NULL PRIMARY KEY, JobID
INT NOT NULL, SkillsRequired VARCHAR(100) NOT NULL, SalaryInUSD INT NOT
NULL,
DurationInMonths VARCHAR(100), PostingDate DATETIME, Deadline DATETIME,
Vacancy INT);

select * from OnCampusEmploymentStats;
show columns from OnCampusEmploymentStats;
```

```
ALTER TABLE Student DROP COLUMN IntakeSemester, DROP COLUMN IntakeYear;
ALTER TABLE Student DROP College;
ALTER TABLE Student DROP DeptName;
select * from Student;

create table Fall (StudentID INT NOT NULL, IntakeYear INT NOT NULL, ID INT
NOT NULL PRIMARY KEY);
select * from Fall;
SHOW COLUMNS FROM Spring;

ALTER TABLE Fall ADD CONSTRAINT FOREIGN KEY (StudentID) REFERENCES Student
(StudentID);

create table Spring (StudentID INT NOT NULL, IntakeYear INT NOT NULL, ID
INT NOT NULL PRIMARY KEY);
select * from Spring;

ALTER TABLE Spring ADD CONSTRAINT FOREIGN KEY (StudentID) REFERENCES
Student (StudentID);

ALTER TABLE EventDetails ADD CONSTRAINT FOREIGN KEY (EventHeadID)
REFERENCES Student (StudentID);

ALTER TABLE StudentClub ADD CONSTRAINT FOREIGN KEY (StudentID) REFERENCES
Student (StudentID);

ALTER TABLE StudentOnCampus ADD CONSTRAINT FOREIGN KEY (StudentID)
REFERENCES Student(StudentID);

ALTER TABLE StudentJob ADD CONSTRAINT FOREIGN KEY (StudentID) REFERENCES
Student (StudentID);

ALTER TABLE Student ADD CONSTRAINT FOREIGN KEY (DeptID) REFERENCES COLLEGE
(DeptID);

ALTER TABLE OnCampusEmploymentStats ADD CONSTRAINT FOREIGN KEY (JobID)
REFERENCES OnCampusEmployment (JobID);
```

```
show columns from Student;
ALTER TABLE StudentJob ADD CONSTRAINT FOREIGN KEY (StudentID) REFERENCES
Student (StudentID);

CREATE TABLE STUDENT (StudentID INT NOT NULL PRIMARY KEY, FirstName
VARCHAR(50), LastName VARCHAR(50), Gender VARCHAR(5), DeptName VARCHAR
(100), College VARCHAR(70), DeptID INT NOT NULL, FeePayment VARCHAR(5),
CreditsEarned INT, IntakeSemester VARCHAR(10), IntakeYear INT);
INSERT INTO STUDENT (StudentID, FirstName, LastName, Gender, DeptName,
College, DeptID, FeePayment, CreditsEarned, IntakeSemester, IntakeYear)
VALUES(1951, "Joshua", "Lewis","M","Information Systems", "College of
engineering", 201, "Yes",8, "Fall", 2022);
select * from student;

INSERT INTO STUDENT (StudentID, FirstName, LastName, Gender, DeptName,
College, DeptID, FeePayment, CreditsEarned, IntakeSemester, IntakeYear)
VALUES(1952, "Darshana", "Roakde","F","Information Systems", "College of
engineering", 201, "Yes",8, "Fall", 2022);

CREATE TABLE COLLEGE (CollegeName varchar(100) NOT NULL, DeptName
varchar(100) NOT NULL, DeptID INT NOT NULL PRIMARY KEY, DeanName
varchar(100)NOT NULL);
SELECT * FROM COLLEGE;
SHOW COLUMNS FROM cOLLEGE;

INSERT INTO COLLEGE (CollegeName, DeptName, DeptID, DeanName)
VALUES("College of Engineering", "Mechantronics", 221, "Kevin Pipe");

CREATE TABLE OnCampusEmployment (JobID int NOT NULL PRIMARY KEY,
Employer VARCHAR(100) NOT NULL, Location VARCHAR(100) NOT NULL,
SkillsRequired VARCHAR(100) NOT NULL, SalaryInUSD INT NOT NULL,
DurationInMonths VARCHAR(100), PostingDate DATETIME, Deadline DATETIME,
Vacancy INT);
SELECT * from OnCampusEmployment;

show columns from OnCampusEmployment;
```

```
INSERT INTO OncampusEmployment (JobID, Employer, Location, SkillsRequired,
SalaryinUSD, DurationInMonths, PostingDate, Deadline, Vacancy)
VALUES(324, "Office of Global Services", "Snell Engineering Center",
"Fluency in English", 11, 6, '2022-11-12 20:30:00', '2022-12-23 20:30:00',
8);
CREATE TABLE InternshipsAndCoop
(JobID INT NOT NULL PRIMARY KEY, JobTitle varchar(500),
SalaryEstimateLower INT, SalaryEstimateUpper INT, JobDescription
varchar(1000),
Rating FLOAT, CompanyName VARCHAR(500), Location VARCHAR(500), HeadQuaters
VARCHAR(500),
Size varchar(100), FoundedYear INT, TypeofOwnership VARCHAR(500), Industry
VARCHAR(500),
Sector VARCHAR(500), Revenue VARCHAR(500), Competitor VARCHAR(100));

SELECT * FROM InternshipsAndCoop;
SHOW COLUMNS FROM iNTERNSHIPSaNDcOOP;

INSERT INTO InternshipsAndCoop(JobID, JobTitle, SalaryEstimateLower,
SalaryEstimateUpper, JobDescription, Rating, CompanyName, Location,
HeadQuaters, Size, FoundedYear, TypeOfOwnership, Industry, Sector, Revenue,
Competitor)
VALUES(1483, "Data Analyst", 41000, 78000, "Job Description
Job #: 1076450
Role: Systems Analyst
Location: Philadelphia, PA

Duties:
Analyze data from a particular business domain and define how best to
organize it in support of Group's cloud data architecture runway
Work with stakeholders to understand the domain's functional and data
requirements
Work with stakeholders to identify use cases for data consumption
Support Data Architecture by rationalizing redundant data and developing
the approach building out the Group cloud data-warehouse and analytics
platforms
Support broader architecture teams in defining a business domain's master
data management approach
```

```
Define and document the Data Lineage for a business domain
Skills:
7+ years of experience working in the Data or Systems Analysis field Strong
SQL skills and knowledge of various database types and technologies
Familiarity with relevant data-related AWS services S3, At", 3.8, "Apex
Systems", "Philadelphia, PA", "Glen Allen, VA", "1001 to 5000 employees",
1995, "Subsidiary or Business Segment", "Staffing & Outsourcing", "Business
Services", "$2 to $5 billion (USD)", "TEKsystems, Insight Global,
Accenture");

CREATE TABLE EventDetails (EID Int NOT NULL PRIMARY KEY, EName
varchar(100), EventHeadID INT NOT NULL, Location varchar(50), Purpose
varchar(1000), Dateandtime datetime, FareinUSD int);
SELECT * FROM EventDetails;
SHOW columns from StudentClub;

INSERT INTO EventDetails(EID, EName, EventHeadID, Location, Purpose,
DateandTime, FareinUSD)
VALUES(521, "Skills Bootcamp in Digital - Information Event", 1463, "Snell
Eng Center", "Find out more about Northeastern University London, Skills
Bootcamp in Data and Service Management; a free-to-the-participant",
'2023-03-17 08:30:00', 15);

Create TABLE StudentClub (ClubID int NOT NULL primary KEY, ClubName
Varchar(100), StudentID INT NOT NULL, Contacts Varchar(100), Motive
varchar(1000));
select * fROM StudentClub;

INSERT INTO StudentClub (ClubID, ClubName, StudentID, Contacts, Motive)
VALUES(620, "Institute of Industrial and Systems Engineers", 1212,
"m.behroozi@northeastern.edu", "environment that promotes professional and
academic development while collaborating with peers and networking with
industry leaders.");

CREATE TABLE StudentOnCampus(StudentID INT NOT NULL , JobID INT, ID INT NOT
NULL PRIMARY KEY);
SELECT * FROM StudentOnCampus;
```

```
INSERT INTO StudentOnCampus(StudentID, JobID) VALUES(108, 324);

CREATE TABLE StudentJob (StudentID INT NOT NULL UNIQUE, JobID int NOT
NULL,ID INT NOT NULL PRIMARY KEY);
select * from StudentJob;
show columns from StudentOnCampus;

INSERT INTO StudentJob(StudentID, JobID) VALUES(101, 2084);

# Assignment3 Questions

select * from student order by Studentid desc;

#1 Display count of students in each department
CREATE VIEW StudentCount AS
SELECT c.DeptName, Count(s.StudentID) FROM Student s INNER JOIN College c
ON s.DeptID=c.DeptID
GROUP BY c.DeptID ORDER BY Count(s.StudentID);

SELECT * from StudentCount;

#2 Which college has more number of student head?
CREATE VIEW MaxStudentHead AS
Select c.CollegeName, Count(c.CollegeName) AS MaxStudentHead FROM College c
INNER JOIN Student s ON c.DeptID=s.DeptID
INNER JOIN StudentClub sc ON s.StudentID=sc.StudentID GROUP BY
c.CollegeName;

show full tables where table_type='VIEW';

SELECT * FROM MaxStudentHead;


# Example for question 2
SELECT c.CollegeName, s.StudentID FROM Student s INNER JOIN StudentClub sc
ON s.StudentID=sc.StudentID WHERE s.College="School of Law";

#3 Number of students who got coop
```

```sql
CREATE VIEW StudentsOnCoop AS
SELECT sj.JobID, COUNT(s.StudentID) AS NumberOfStudents FROM Student s
INNER JOIN StudentJob sj ON s.StudentID=sj.StudentID
GROUP BY sj.JobID ORDER BY JobID;

SELECT * from StudentsOnCoop;

#4 Which department got maximum placements?
CREATE VIEW MaxPlacements AS
SELECT s.DeptID, c.DeptName, COUNT(sj.JobID) AS NumberOfStudents FROM
College c INNER JOIN Student s ON c.DeptID=s.DeptID
INNER JOIN StudentJob sj ON s.StudentID=sj.StudentID
GROUP BY s.DeptID ORDER BY COUNT(sj.JobID) DESC;

SELECT * FROM MaxPlacements;

#5 Average salary offered to students in each department
SELECT s.DeptID, AVG(i.SalaryEstimateUpper) AS Average
FROM Student s INNER JOIN StudentJob sj ON s.StudentID = sj.StudentID
INNER JOIN InternshipsAndCoop i on sj.JobID = i.JobID GROUP BY s.deptid;

SELECT * FROM AvgSalary;

#6 Maximum number of JobOpenings
CREATE VIEW MaxJobOpening AS
SELECT JobID, JobTitle, MAX(mycount) AS Maximum_Number_Of_Job_Openings FROM
(SELECT COUNT(JobTitle) mycount, JobTitle, JobID FROM InternshipsAndCoop
GROUP BY JobTitle) as sample;

SELECT * FROM MaxJobOpening;

#7 Job Count for each intake

-- Count for Fall Intake
CREATE VIEW FallIntakeCount AS
SELECT COUNT(sj.StudentID) AS CountOfStudents, f.IntakeYear FROM Student s
INNER JOIN StudentJob sj ON s.StudentID=sj.StudentID
INNER JOIN Fall f ON s.StudentID=f.StudentID GROUP BY f.IntakeYear;
```

```sql
SELECT * FROM FallIntakeCount;

-- Count for Spring Intake
CREATE VIEW SpringIntakeCount AS
SELECT COUNT(sj.StudentID) AS CountOfStudents, sp.IntakeYear FROM Student s
INNER JOIN StudentJob sj ON s.StudentID=sj.StudentID
INNER JOIN Spring sp ON s.StudentID=sp.StudentID GROUP BY sp.IntakeYear;

SELECT * FROM SpringIntakeCount;

#8 Booming Industry
CREATE VIEW BoomingIndustry AS
SELECT ic.Industry, COUNT(ic.Industry) BoomingIndustry FROM
InternshipsAndCoop ic INNER JOIN StudentJob sj ON ic.JobID=sj.JobID
GROUP BY ic.Industry ORDER BY COUNT(ic.Industry) DESC;

SELECT * FROM BoomingIndustry;

#9 Which Second Highest Salary
CREATE VIEW SecongHighestSalary AS
SELECT MAX(SalaryEstimateUpper) AS Second_Highest_Salary from
InternshipsAndCoop  WHERE SalaryEstimateUpper < (SELECT
Max(SalaryEstimateUpper) FROM InternshipsAndCoop);

SELECT * FROM SecongHighestSalary;

#10 Students who got internship and oncampus
CREATE VIEW StudentswithOncampusAndCoop AS
SELECT sj.StudentID FROM StudentOnCampus soc INNER JOIN StudentJob sj ON
soc.StudentID=sj.StudentID;

SELECT * FROM StudentswithOncampusAndCoop;
```

6| Python Scripts

Scripts for establishing DB connection and inserting data into DB:

```
   import tqdm
   import snscrape.modules.twitter as tweet
   import pandas as pd
   import datetime
   import re

   import mysql.connector
   conn = mysql.connector.connect(host='localhost',
                          database='mastersstudents',
                           user='root',
                           password='root')

   cursor = conn.cursor()

   #Executing an MYSQL function using the execute() method
   cursor.execute(\"SELECT DATABASE()\")
   # Fetch a single row using fetchone() method.
   data = cursor.fetchone()
   print(\"Connection established to: \",data)
   cursor = conn.cursor()

   # sheet_names = [\"Fall\",\"Spring\",\"StudentJob\",
\"StudentOnCampus\",\"OnCampusEmployment\",\"OnCampusEmploymentStats\"]
   "sheey_names = [\"OnCampusEmployment\"]

   for sheet in sheet_names:
       df =
pd.read_excel(\"D:\\Darshana\\\\NEU\\Study\\DMDD\\Project\\MStudents\\Assig
nment4\\Mstudents.xlsx\", sheet_name= sheet)
       c = 0
       for i, row in df.iterrows():
           try:
               if sheet == \"Student\":
                   query = \"\"\"INSERT INTO Student (StudentID, FirstName,
LastName, Gender, DeptId, FeePayment, CreditsEarned) VALUES({0}, \"{1}\",
\"{2}\", \"{3}\", {4}, \"{5}\", {6});\"\"\".format(int(row['StudentID']),
row['FirstName'], row['LastName'], row['Gender'], int(row['DeptID']),
row['FeePayment'],int(row['CreditsEarned']))

               if sheet == \"Fall\":
                   query = \"\"\"INSERT INTO Fall (StudentID,
IntakeYear,ID) VALUES({0}, {1},
{2})\"\"\".format(int(row['StudentID']),int(row['IntakeYear']),int(row['ID'
]))
```

```
                if sheet == \"Spring\":
                    query = \"\"\"INSERT INTO Spring (StudentID,
IntakeYear,ID) VALUES({0}, {1},
{2})\"\"\".format(int(row['StudentID']),int(row['IntakeYear']),int(row['ID'
]))

                if sheet == \"College\":\n",
                    query = \"\"\"INSERT INTO College (CollegeName,
DeptName, DeptID, DeanName) VALUES(\"{0}\", \"{1}\", {2},
\"{3}\")\"\"\".format(row['CollegeName'], row['DeptName'],
int(row['DeptID']),row['DeanName'])

                if sheet == \"EventDetails\":
                    query = \"\"\"INSERT INTO EventDetails(EID, EName,
EventHeadID, Location, Purpose, Dateandtime, FareinUSD)
VALUES({0},\"{1}\",{2},\"{3}\",\"{4}\",\"{5}\",\"{6}\")\"\"\".format(int(ro
w['EID']), row['EName'], int(row['EventHeadID']),
row['Location'],row['Purpose'],datetime.datetime.strptime(row['Dateandtime'
].replace(\" EST\", \"\"), \"%a, %b %d, %Y %I:%M %p\"), row['FareinUSD'])

                if sheet == \"OnCampusEmployment\":
                    query = \"\"\"INSERT INTO OnCampusEmployment(JobID,
Employer, Location)
VALUES({0},\"{1}\",\"{2}\")\"\"\".format(int(row['JobID']),
row['Employer'], row['Location'])

                if sheet == \"OnCampusEmploymentStats\"
                    query = \"\"\"INSERT INTO OnCampusEmploymentStats(ID,
JobID, SkillsRequired, SalaryInUSD, DurationInMonths, PostingDate,
Deadline, Vacancy)
VALUES({0},{1},\"{2}\",{3},{4},\"{5}\",\"{6}\",{7})\"\"\".format(int(row['I
D']), int(row['JobID']), row['SkillsRequired'],
int(row['SalaryInUSD']),int(row['DurationInMonths']),
datetime.datetime.strptime(row['PostingDate'].replace(\" EST\", \"\"),
\"%a, %b %d, %Y %I:%M %p\"),
datetime.datetime.strptime(row['Deadline'].replace(\" EST\", \"\"), \"%a,
%b %d, %Y %I:%M %p\"),int(row['Vacancy']))

                if sheet == \"StudentOnCampus\":
                    query = \"\"\"INSERT INTO StudentOnCampus(StudentID,
JobID,ID) VALUES({0},{1},{2})\"\"\".format(int(row['StudentID']),
int(row['JobID']),int(row['ID']))

                if sheet == \"InternshipsAndCoop\":
                    salary = row[\"SalaryEstimate\"].replace(\"(Glassdoor
est.)\", \"\").replace(\"K\", \"000\").replace(\"$\",\"\")
                    salary = salary.split(\"-\")
   #                   import ipdb; ipdb.set_trace()
                    lower_salary = salary[0]
```

```
                    upper_salary = salary[1]
                    rating = row['Rating']
                    company_name =
row['CompanyName'].replace('\\n','').replace(str(rating),'')
                    job_description = row['JobDescription'][:900]
                    job_description = re.sub(r\"\\\"\",\"\\'\",
job_description)
                    job_title = re.sub(r\"\\\"\",\"\\'\", row['JobTitle'])
                    query = \"\"\"INSERT INTO InternshipsAndCoop(JobID,
JobTitle, SalaryEstimateLower, SalaryEstimateUpper, JobDescription, Rating,
CompanyName, Location, HeadQuaters, Size, FoundedYear, TypeOfOwnership,
Industry, Sector, Revenue, Competitor) VALUES({0},
\"{1}\",{2},{3},\"{4}\",{5},\"{6}\",\"{7}\",\"{8}\",\"{9}\",{10},\"{11}\",\
"{12}\",\"{13}\",\"{14}\",\"{15}\")\"\"\".format(int(row['JobID']),
job_title, int(lower_salary), int(upper_salary),
job_description,float(row['Rating']), company_name,row['Location'],
row['HeadQuarters'],row['Size'],int(row['Founded']),row['TypeOfOwnership'],
row['Industry'],row['Sector'], row['Revenue'],row['Competitors'])

                if sheet == \"StudentClub\":
                    query = \"\"\"INSERT INTO StudentClub (ClubID, ClubName,
StudentID, Contacts, Motive)
VALUES({0},\"{1}\",{2},\"{3}\",\"{4}\")\"\"\".format(int(row['ClubID']),
row['ClubName'], row['StudentID'], row['Contacts'], row['Motive'])\n",

                if sheet == \"StudentJob\”
                    query = \"INSERT INTO StudentJob (StudentID, JobID, ID)
VALUES({0}, {1},{2})\".format(int(row['StudentID']),
int(row['JobID']),int(row['ID']))
                cursor.execute(query)

            except Exception as e:
                print (e, sheet)
                 print (query)
                 break
                c+=1
        print (c)
        conn.commit()"
```

Scripts for Data Visualization:

```
#Load the required libraries
import pandas as pd
import numpy as np
import seaborn as sns
#Load the data
df = pd.read_csv(\"D:\\Darshana\\\\NEU\\Study\\DMDD\\dummy.csv\"),
#View the data
df.head()"

#Basic Info

df.info()

#Describe the data

df.describe()

df.duplicated().sum()


df[[\"Rating\"]].plot.hist(bins = 20, title = \"Old Rating\")

df.Rating

df[\"Rating_new\"]=df[\"Rating\"]
df.Rating_new.replace([-1.0],[0.0],inplace=True)
df.Rating_new

df.isnull().sum()

df.corr()

sns.heatmap(df.corr())


df[[\"Founded\"]].plot.hist(bins = 20, title = \"Old Founded\")

df.Founded


df[\"Founded_new\"]=df[\"Founded\"]
df.Founded_new.replace([-1],[0],inplace=True)
df.Founded_new


df.isnull().sum()
df.corr()
```

```
sns.heatmap(df.corr())
```

7] Steps followed right from choosing datasets to getting the final DB.

- Since our project is related to students and their profile management, we have extracted data from kaggle as well as Northeastern University's website and structured it according to our project's needs by using SQL and Python.
- To make the data more presentable and for graphic representation, we performed Data Visualization. Used some python methods and tools to replace null values. For example: df.replace(), df.corr(),df.isnull(), df.describe(), etc.
- To obtain the final database, we checked if the tables are in 1NF,2NF and 3NF. We performed normalization on the table which did not satisfy the conditions.
