**DMDD Assigment 3**

**Project Description:**

This project aims to create a database containing all relevant resources. The resources database will include:

Programs offered on the Boston campus of Northeastern University in the College of Engineering, and details about their duration, offerings, and specializations.

The relevant study material (paid and free resources) for all courses offered by COE.

Tools and software required by the courses.

Program contacts (Professors/Advisors/Current Teaching Assistants).

Future career outcomes post-degree completion.

We aspire to help learners through our initiative by helping them collate the resources they need to navigate their course.

Scope:

We have shortlisted 3 graduate degree programs offered by Northeastern University’s College Of Engineering. Our database is built around these programs.

Masters of Science in Information Systems

Masters of Science in Data Analytics Engineering

Masters of Science Software Engineering Systems

Plan of Action:

We have used 3 sources to collect the data:

Website Scraping: As most of the details about the course offerings are available on the Northeastern university’s website, we have scraped multiple pages to gather information in one place.

API Scraping: We have scraped LinkedIn API to collect information about the jobs a student can secure after graduating from the courses within our scope. We have used Python and Scrapy to get job-related data. We used the ai-skunks GitHub as a starting point to get jobs-related data.

Google Forms: To get more on-ground data, we spoke to a few students and circulated google forms to collect data pertaining to resources that are relevant to the aforementioned programs.

We have collated all the above data sources to create a database and established connections between program offerings, jobs related to these programs, events conducted to promote jobs and programs, and the resources learners need to excel in their coursework.

**Data Cleaning:**

1. NEU\_Programs Table:

- Removed all programs name which were not in Boston in the dataframe using drop duplicates command in python

2. Neu\_Course\_Catalog Table:

- Unwanted text removed in python after scraping data and then inserted into table.

3. NEU\_Course\_Faculty Table:

- Data inserted from csv file by collecting responses from Students

4. Resource\_Material1 Table:

- Data inserted from csv file as per responses given by students

5. NEU\_Event Table:

- This data is manually inserted in SQL by using insert query. Hence, the data is validated and then inserted into table

6. NEU\_Specialization Table:

- Data is validated beffore inserting into table.

7. Course\_Core\_Requirement Table:

- Accurate data scraped from Northeastern University website. No cleaning required

8. Jobs\_Info Table:

- Duplicate jobs are removed from the scraped data in jupyter notebook

**Assignment Execution Steps:**

Run the below Queries for the complete code Execution by following the given steps:

1. Create NEU\_Programs table

CREATE TABLE NEU\_Programs (

program\_name varchar(100),

duration varchar(100),

program\_url varchar(255),

PRIMARY KEY (program\_name)

);

2. Create NEU\_Course\_Catalog

CREATE TABLE NEU\_Course\_Catalog (

program\_name varchar(50),

course\_id varchar(50),

course\_name varchar(100),

course\_description longtext,

PRIMARY KEY (course\_id)

);

## Add Foreign Key Constraint

ALTER TABLE NEU\_Course\_Catalog

ADD CONSTRAINT NEU\_Course\_Catalog\_fk1 FOREIGN KEY (program\_name)

REFERENCES NEU\_Programs(program\_name);

3. Create Resources\_Material1 table

CREATE TABLE Resource\_material1(

resource\_id integer auto\_increment,

course\_id varchar(255),

software\_download\_url longtext,

professional\_certificate longtext,

online\_platform longtext,

course\_rating integer,

Primary Key (resource\_id)

);

## Set auto increment value to 3100

ALTER TABLE Resource\_material1 AUTO\_INCREMENT=3100;

## Add Foreign Key Constraint

ALTER TABLE Resource\_material1

ADD CONSTRAINT Resource\_material1\_fk FOREIGN KEY (course\_id)

REFERENCES NEU\_Course\_Catalog(course\_id);

4. Create NEU\_Course\_Faculty Table

CREATE TABLE NEU\_Course\_Faculty (

faculty\_id integer auto\_increment,

program\_name varchar(50),

course\_id varchar(100),

faculty\_name varchar(100),

faculty\_rating integer,

Primary Key (faculty\_id)

);

## Set auto increment value to 4100

ALTER TABLE NEU\_Course\_Faculty AUTO\_INCREMENT=4100;

## Add Foreign Key Constraint

ALTER TABLE NEU\_Course\_Faculty

ADD CONSTRAINT NEU\_Course\_Faculty\_fk1 FOREIGN KEY (course\_id)

REFERENCES NEU\_Course\_Catalog(course\_id);

5. Create Course\_Core\_Requirement table

CREATE TABLE course\_core\_requirement (

core\_course\_id integer auto\_increment,

program\_name varchar(50),

course\_id varchar(50),

course\_name varchar(100),

PRIMARY KEY (core\_course\_id)

);

## Set auto increment value to 1101

ALTER TABLE course\_core\_requirement AUTO\_INCREMENT=1101;

## Add Foreign Key Constraint

ALTER TABLE Course\_Core\_Requirement

ADD CONSTRAINT NEU\_Course\_Core\_Requirement\_fk1 FOREIGN KEY (program\_name)

REFERENCES NEU\_Programs(program\_name);

6. Create Jobs\_Info Table

CREATE TABLE Jobs\_Info (

Job\_No integer auto\_increment,

program\_name varchar(255),

title varchar(255),

job\_id varchar(255),

company\_name varchar(255),

location varchar(255),

date\_posted date ,

link longtext,

description longtext,

seniority\_level varchar(255) ,

employement\_type varchar(255),

job\_function varchar(255),

industry varchar(255),

primary key (job\_No)

);

## Set auto increment value to 900

ALTER TABLE jobs\_info AUTO\_INCREMENT=900;

## Add Foreign Key constraints

ALTER TABLE Jobs\_Info

ADD CONSTRAINT Jobs\_Info\_fk1 FOREIGN KEY (program\_name)

REFERENCES NEU\_Programs(program\_name);

7. Run the Python Script in Jupyter Notebook

8. Run Pycharm Script for Scraping jobs on Linkedin

9. NEU\_Event table:

CREATE TABLE NEU\_Event (

event\_id integer auto\_increment,

program\_name varchar(50),

event\_name varchar(255),

Primary Key (event\_id)

);

## Set auto increment value to 101

ALTER TABLE neu\_event AUTO\_INCREMENT=101;

## Add Foreign Key constraints

ALTER TABLE NEU\_Event

ADD CONSTRAINT NEU\_Event\_fk1 FOREIGN KEY (program\_name)

REFERENCES NEU\_Programs(program\_name);

## SQL to insert data in NEU\_Event Table

INSERT INTO neu\_event (program\_name,event\_name) VALUES ("Information Systems","Ethics Institute Speaker, Michael Hannon ( Friday, November 18, 2022 12pm to 1:30pm )");

INSERT INTO neu\_event (program\_name,event\_name) VALUES ("Information Systems","Ethics Institute Speaker, Don Fallis ( Friday, December 2, 2022 12pm to 1:30pm)");

INSERT INTO neu\_event (program\_name,event\_name) VALUES ("Information Systems","Using a Systems-based Culturally Responsive Evaluation (SysCRE) Framework (Wednesday, November 16, 2022 10am to 11:30am)");

INSERT INTO neu\_event (program\_name,event\_name) VALUES ("Information Systems","Designing an Engaging Academic Presentation (Friday, November 11, 2022)");

INSERT INTO neu\_event (program\_name,event\_name) VALUES ("Information Systems","Listener Research Study (Friday, November 11, 2022)");

INSERT INTO neu\_event (program\_name,event\_name) VALUES ("Data Analytics Engineering","Problem Play (Friday, November 11, 2022 8pm to 10pm)");

INSERT INTO neu\_event (program\_name,event\_name) VALUES ("Data Analytics Engineering","Graduate programs open house ");

INSERT INTO neu\_event (program\_name,event\_name) VALUES ("Data Analytics Engineering","Women in STEM: An Evening with Distinguished Alumni (6th March 2023)");

INSERT INTO neu\_event (program\_name,event\_name) VALUES ("Data Analytics Engineering","Fall Graduate Programs Open House ");

INSERT INTO neu\_event (program\_name,event\_name) VALUES ("Data Analytics Engineering","Northeastern University Mumbai Education Exhibition");

INSERT INTO neu\_event (program\_name,event\_name) VALUES ("Software Engineering Systems","Graduate Studies Information Session");

INSERT INTO neu\_event (program\_name,event\_name) VALUES ("Software Engineering Systems","Next Steps for Critical Infrastructure & Cyber Security: CISA Dir. Wales (Tue, Nov 15, 11:00 PM)");

INSERT INTO neu\_event (program\_name,event\_name) VALUES ("Software Engineering Systems","International Education Week Kick-Off Event (Mon, Nov 14, 2:00 PM)");

INSERT INTO neu\_event (program\_name,event\_name) VALUES ("Software Engineering Systems","Greatest MINDS Generations - A Gathering of Black Bostonians (All Ages)2023 (Thu, Jan 19, 9:00 AM)");

INSERT INTO neu\_event (program\_name,event\_name) VALUES ("Software Engineering Systems","Can AI Help Create a More Sustainable World? (Tue, Nov 15, 5:00 PM)");

10. NEU\_Specialization Table

CREATE TABLE NEU\_Specialization (

spec\_course\_id integer auto\_increment,

course\_id varchar(50),

credit\_hours integer,

specialization varchar(100),

Primary Key (spec\_course\_id)

);

## Set auto increment value to 1000

ALTER TABLE NEU\_Specialization AUTO\_INCREMENT=1000;

## Adding Foreign\_Key Constraints

ALTER TABLE NEU\_Specialization

ADD CONSTRAINT NEU\_Specialization\_fk1 FOREIGN KEY (course\_id)

REFERENCES NEU\_Course\_Catalog(course\_id);

## SQL to insert data in NEU\_Specialization Table

INSERT INTO NEU\_Specialization (course\_id,credit\_hours,specialization) VALUES ("CSYE 6225",4,"Big Data Systems and Analytics Concentration");

INSERT INTO NEU\_Specialization (course\_id,credit\_hours,specialization) VALUES ("DAMG 7245",4,"Big Data Systems and Analytics Concentration");

INSERT INTO NEU\_Specialization (course\_id,credit\_hours,specialization) VALUES ("INFO 7250",4,"Big Data Systems and Analytics Concentration");

INSERT INTO NEU\_Specialization (course\_id,credit\_hours,specialization) VALUES ("INFO 7390",4,"Big Data Systems and Analytics Concentration");

INSERT INTO NEU\_Specialization (course\_id,credit\_hours,specialization) VALUES ("CSYE 7280",4,"Intelligent Systems Concentration");

INSERT INTO NEU\_Specialization (course\_id,credit\_hours,specialization) VALUES ("INFO 7375",4,"Intelligent Systems Concentration");

INSERT INTO NEU\_Specialization (course\_id,credit\_hours,specialization) VALUES ("INFO 7390",4,"Intelligent Systems Concentration");

INSERT INTO NEU\_Specialization (course\_id,credit\_hours,specialization) VALUES ("INFO 7610",4,"Intelligent Systems Concentration");

INSERT INTO NEU\_Specialization (course\_id,credit\_hours,specialization) VALUES ("INFO 7500",4,"Smart Contracts Concentration");

INSERT INTO NEU\_Specialization (course\_id,credit\_hours,specialization) VALUES ("INFO 7510",4,"Smart Contracts Concentration");

INSERT INTO NEU\_Specialization (course\_id,credit\_hours,specialization) VALUES ("INFO 7520",4,"Smart Contracts Concentration");

INSERT INTO NEU\_Specialization (course\_id,credit\_hours,specialization) VALUES ("INFO 7525",2,"Smart Contracts Concentration");

INSERT INTO NEU\_Specialization (course\_id,credit\_hours,specialization) VALUES ("INFO 7535",2,"Smart Contracts Concentration");

INSERT INTO NEU\_Specialization (course\_id,credit\_hours,specialization) VALUES ("CSYE 7280",4,"User Experience Concentration");

INSERT INTO NEU\_Specialization (course\_id,credit\_hours,specialization) VALUES ("INFO 6150",4,"User Experience Concentration");

INSERT INTO NEU\_Specialization (course\_id,credit\_hours,specialization) VALUES ("INFO 6245",4,"User Experience Concentration");

INSERT INTO NEU\_Specialization (course\_id,credit\_hours,specialization) VALUES ("INFO 6350",4,"User Experience Concentration");

11. Delete data from table to run the python script again

delete from jobs\_info;

delete from neu\_specialization;

delete from neu\_course\_faculty;

delete from resource\_material1;

delete from neu\_course\_catalog;

delete from neu\_event;

delete from course\_core\_requirement;

delete from neu\_programs;

**Use Cases**

1.How many total courses are available for Information Systems Program in Northeastern University?

SQL Statement:

SELECT

a.program\_name,

COUNT(a.course\_name),

b.program\_url

FROM

neu\_course\_catalog a

LEFT OUTER JOIN

NEU\_Program\_Detail b

ON

a.program\_name = b.program\_name

where b.program\_name = 'Information Systems'

GROUP BY

a.program\_name;

2. Which courses do we need to take to do specialization in User Experience Concentration?

SQL Statement:

select \* from neu\_specialization where specialization = 'User Experience Concentration';

3. List top rated professors at NEU whose rating is above 4. Also mention the courses and its description taught by the professor.

SQL Statement:

SELECT

a.program\_name,b.faculty\_name,b.faculty\_rating, a.course\_name, a.course\_description

FROM

neu\_Course\_Catalog a

RIGHT OUTER JOIN

NEU\_course\_Faculty b

ON

a.course\_id = b.course\_id

WHERE

b.faculty\_rating >= 4

Order by b.faculty\_rating desc;

4. What are the core course requirements for Information Systems, Software Engineering Systems and Data Analytics?

SQL Statement:

SELECT

a.program\_name, GROUP\_CONCAT(a.course\_id SEPARATOR '|'),

COUNT(a.course\_id), b.program\_url

FROM

Course\_Core\_requirement a

LEFT JOIN

NEU\_Program\_Detail b

ON

a.program\_name = b.program\_name

GROUP BY

a.program\_name;

5. List courses available for Data Analytics Engineering along with its course\_outcome.

SQL Statement:

SELECT

program\_name, course\_id,course\_name, course\_description

FROM

NEU\_Course\_Catalog

WHERE program\_name = 'Data Analytics Engineering';

6. What is the average overall rating given by students to a professor for a given course?

SQL Statement:

SELECT faculty\_name, round (AVG(faculty\_rating),1) as Average\_rating

FROM NEU\_course\_Faculty

GROUP BY faculty\_name;

7.Which courses students need to take to do specialization in Big Data Systems concentration and which resources are available for them?

SQL Statement:

SELECT

c.program\_name,a.specialization, a.course\_id, c.course\_name,

b.course\_rating,b.online\_platform

FROM

NEU\_Specialization a

LEFT JOIN

NEU\_course\_catalog c on c.course\_id = a.course\_id

LEFT JOIN

Resource\_material1 b

ON a.course\_id = b.course\_id

WHERE

a.specialization like '%Big Data%';

8. List details about all the Masters programs available in Northeastern University

SQL Statement:

SELECT

\*

FROM

NEU\_program\_Detail;

9. Which job positions for students in Data Analytics Engineering and list the details related to those positions?

SQL Statement:

SELECT

program\_name, title, company\_name,description

FROM

Jobs\_info

WHERE

program\_name LIKE '%Data%';

10. List the events hosted by Northeastern University for students by Data Analytics Engineering Department.

SQL Statement:

SELECT

program\_name, event\_name

FROM

Neu\_Event

WHERE

program\_name = 'Data Analytics Engineering';

11. Which professors teach courses related to Big Data Systems and list the ratings given by students for the course?

SQL Statement:

SELECT

a.faculty\_name, c.course\_rating

FROM

NEU\_Course\_Faculty a

JOIN NEU\_Course\_Catalog b ON a.program\_name = b.program\_name

JOIN Resource\_material1 c ON c.course\_id = b.course\_id

WHERE

b.course\_name like '%Big-Data Systems%'

ORDER BY c.course\_rating;

12. Find the average course rating for Information Systems Program

SQL Statement:

SELECT

b.program\_name, AVG(a.course\_rating)

FROM

Resource\_material1 a

RIGHT JOIN

NEU\_Course\_Catalog b

ON

a.course\_id = b.course\_id

WHERE

b.program\_name ='Information Systems';

13. List courses with lowest rating.

SQL Statement:

SELECT

a.course\_id,a.course\_rating,b.course\_name,b.course\_description

FROM

Resource\_material1 a

LEFT JOIN

NEU\_Course\_catalog b

on a.course\_id = b.course\_id

where

course\_rating = (select min(course\_rating) from resource\_material1);

14. Find the professors who taught DAMG 7350, INFO 5100, DAMG 6105, DAMG 7275 courses.

SQL Statement:

SELECT \* FROM NEU\_course\_Faculty WHERE faculty\_name = ANY (

SELECT

faculty\_name

FROM

NEU\_course\_Faculty

WHERE

course\_id IN ('DAMG 7350', 'INFO 5100', 'DAMG 6105', 'DAMG 7275'));

15. Which resources are available for students who have chosen Information Systems program?

SQL Statement:

SELECT

a.program\_name, b.course\_id, c.software\_download\_url,

c.professional\_certificate, c.online\_platform

FROM

NEU\_Course\_Faculty a

JOIN NEU\_Course\_Catalog b ON a.program\_name = b.program\_name

JOIN Resource\_material1 c ON c.course\_id = b.course\_id

WHERE

a.program\_name = 'Information Systems';

**Sample data from all tables**

Neu\_Programs table:

Graphical user interface, text, application

Description automatically generated

NEU\_Course\_Catalog Table:

Graphical user interface, text, application, email

Description automatically generated

Jobs\_Info Table:

Graphical user interface, text, application, email

Description automatically generated

Course\_Core\_Requirement Table:

Graphical user interface, text, application, email

Description automatically generated

NEU\_Event Table:

Graphical user interface, text, application, email

Description automatically generated

NEU\_Faculty Table:

Graphical user interface, text, email

Description automatically generated

NEU\_Specialization Table:

Graphical user interface, text, application

Description automatically generated

Resource\_Material1 Table:

Graphical user interface, text, application, email

Description automatically generated