

## ASSIGNMENT 2: Scraping Twitter

**Team Name:** MStudents

**Team Members:** Renuka Huddar (002750433)  
Darshana Rokade (002705752)

### USE CASES:

1]

#### **Use Case: Search for student clubs**

**Description:** student search for student clubs in the database

**Actors:** student

**Precondition:** Student must have a unique student ID to search for clubs

**Steps:**

**Actor action** – student enters club name to search for respective clubs

**System Responses** – student club is displayed

**Post Condition:** student views and join the student club

**Error:** Student club not available

#### **SQL Query:**

```
SELECT * FROM StudentClubs;
```

#### **Relational Algebra:**

```
 $\sigma$  ID,Name,Head,Contacts,Motive (StudentClubs)
```

2]

#### **Use Case: View current employers on campus**

**Description:** student can view details of on campus employers

**Actors:** students

**Precondition:** Student should be eligible to apply for positions

**Steps:**

**Actor action** – students view employers

**System Responses** – displays On campus employment positions

**Post Condition:** student can view and apply for relevant job positions

**Error:** Student is not eligible to apply for on campus jobs

**SQL Query:**

```
SELECT Employer from OnCampusEmployment;
```

**Relational Algebra:**

$\sigma$  JobID,Employer,Location,SkillsRequired,SalaryInUSD,  
DurationInMonths,PostingDate, Deadline, Vacancy (OnCampusEmployment)

3]

**Use Case: View the events above \$10**

**Description:** student views the event details above a particular fare

**Actor:** student

**Steps:**

**Actor action:** student views the events whose fare is above \$10

**System Responses:** list of events whose fare is above \$10 is displayed

**Post Condition:** system displays the list of events for the condition

**SQL Query:**

```
SELECT *FROM EventDetails WHERE FareinUSD>10;
```

**Relational Algebra:**

$\sigma$  (EID, EName,Location, Purpose, Dateandtime,(FareinUSD>10))(Events)

4]

**Use Case: Registration of student**

**Description:** student details are entered in student table

**Actor:** system admin

**Steps:**

**Actor action:** Student request for registration

**System Responses:** If student information is correct then the student is enrolled and the use case ends

**Post Condition:** student enrolled successfully

**Alternate Path:** The student details are invalid and system throws an error

**Error:** Student details are incorrect

**SQL Query:**

```
INSERT INTO student (StudentID, StudentName, Gender, Course, College, FeePayment, CreditsEarned, IntakeYear) VALUES(1951, "Joshua Lewis", "M", "Information Systems", "College of engineering", "Yes", 8, 2022);
```

```
INSERT INTO student (StudentID, StudentName, Gender, Course, College, FeePayment, CreditsEarned, IntakeYear) VALUES(1952, "Andrew Mack", "F", "Health Informatics", "Bouve College of Health Sciences", "yes", 7, 2021);
```

**Note:** Relational Algebra is only used to perform operations.

5]

**Use Case:** View course and college details

**Description:** student views the colleges and courses

**Actor:** student

**Precondition:** student must be enrolled in at least one course

**Steps:**

**Actor action:** student views list of courses and colleges

**System Responses:** Displays list of colleges and courses offered

**Alternate Path:** There is no such course offered/ invalid college name

**Error:** No course available.

**SQL Query:**

SELECT CollegeName,Course FROM College;

**Relational Algebra:**

$\sigma$  CollegeName,Course(College)

6]

**Use Case:** View internships and coops

**Description:** student views the jobs available to him/her

**Actor:** Student

**Precondition:** Student must satisfy eligibility criteria

**Steps:**

**Actor action:** student view list of jobs

**System Responses:** Display all jobs

**Alternate Path:** There is no vacancy

**Error:** No internships/coops found

**SQL Query:**

SELECT \*FROM InternshipsAndCoop;

**Relational Algebra:**

$\sigma$  ( JobID,Company, SkillsRequried, Salary,  
Duration,PostingDate, Deadline,Position(InternshipsAndCoop))

7]

**Use Case:** View tweets by person related to Northeastern Events

**Description:** Students view tweets by person

**Actor:** student

**Precondition:** there should be at least one tweet by person

**Steps:**

**Actor action:** student views the list of tweets

**System Responses:** Displays list of tweets

**Alternate Path:** There are no related tweets

**Error:** No related tweets available.

**SQL Query:**

```
SELECT * FROM TWITTER WHERE QUERY = "Northeastern Events";
```

**Relational Algebra:**

$$\sigma (\text{CollegeName}, \text{DeptName}, \text{DeptID}, \text{DeanName}) (\text{Query} = \text{"Northeastern Events"}) (\text{College})$$

8]

**Use Case:** Search for tweets using hashtags

**Description:** student views the tweets under specific hashtags

**Actor:** Student

**Precondition:** Tweet should be made using valid hashtag

**Steps:**

**Actor action:** student views the list of tweets under that hashtag

**System Responses:** displays list of tweets accordingly

**Alternate Path:** No hashtag found

**Error:** Invalid Hashtag

**SQL QUERY:**

```
SELECT * FROM Twitter WHERE Hashtags LIKE '%NORTHEASTERN EVENTS';
```

**Relational Algebra:**

$\sigma$  (TwitterID,  
TwitterHandle,userID,TwitterText,Hashtags,CreatedAt,Likes,retweetCount,replyCount,  
Query (Date=2022-11-01 AND 2022-11-11))(Hashtags='%NORTHEASTERN  
EVENTS'))(Twitter)

9]

**Use Case:** View latest tweets

**Description:** students view recent tweets

**Actor:** student

**Precondition:** Students must select latest tweet

**Steps:**

**Actor action:** student view list of recent tweets

**System Responses:** display list of latest tweets

**Alternate Path:** there are no recent tweets

**Error:** No tweets found

**SQL Query:**

SELECT \* FROM Twitter WHERE

Date BETWEEN 2022-11-01 AND 2022-11-11;

**Relational Algebra:**

$\sigma$  TwitterID,  
TwitterHandle,userID,TwitterText,Hashtags,CreatedAt,Likes,retweetCount,replyCount,  
Query (Date=2022-11-01 AND 2022-11-11))(Twitter)

10]

**Use Case:** View profile details of person who made the tweet

**Description:** students can view profile details of person who made the tweet

**Actor:** Student

**Precondition:** Person must have valid credentials

**Steps:**

**Actor action:** Student view persons' profile

**System Responses:** displays persons' profile

**Error:** No person found

**SQL Query:**

```
SELECT TwitterHandle FROM twitter;
```

**Relational Algebra:**

$\sigma$  TwitterHandle (Twitter)

**TWITTER QUERIES:**

```
CREATE TABLE 'Twitter' ( 'TwitterID' BIGINT, 'TwitterHandle' VARCHAR(100), 'TwitterText' VARCHAR(1000), 'HashTags' VARCHAR(100), 'CreatedAt' DATETIME, 'Query' VARCHAR(100), PRIMARY KEY ('TwitterID'));
```

```
CREATE TABLE `Tweet_Tags` (`TwitterID` INT NOT NULL, `HashTags` VARCHAR(20) PRIMARY KEY (`TwitterID`));
```

```
CREATE TABLE `TwitterUser` (`TwitterID` INT NOT NULL, `UserName` VARCHAR(50), `Name` VARCHAR(50), `CreatedAt` DATETIME, `Followers_Count` INT, `FRIENDS_Count` INT, `Account_Verified` VARCHAR(50), `Description` VARCHAR(100), `Profile_image_url` String, PRIMARY KEY (`TwitterID`));
```

**Queries you must answer about your physical model (In Relational algebra & SQL):****1. What user posted this tweet?****SQL Query:**

```
SELECT TwitterUser.UserName, TwitterUser.Name, Twitter.TwitterText from TwitterUser  
Inner join Twitter ON Twitter.userid=TwitterUser.userid where Twitter.Hashtags like  
'%Northeastern%';
```

**Relational Algebra:**

$\pi$  a . username, b . name, c . twittertext

$\sigma a . \text{userid} = b . \text{userid} \text{ AND } c . \text{hashtags LIKE } \% \text{Northeastern} \%$   
 $(\rho a \text{ twitteruser } \times \rho b \text{ twitter})$

## 2. When did the user post this tweet?

### SQL Query:

SELECT CreatedAt, TwitterText FROM Twitter Where Query like '%Co-op%';

### Relational Algebra:

$\sigma \text{ CreatedAt, Twitter} = \% \text{Co-op}\% (\text{Twitter})$

## 3. What tweets have this user posted in the past 24 hours?

### SQL Query:

SELECT \* FROM Twitter WHERE userid= 34012970 and CreatedAt >= NOW() - INTERVAL 1 DAY;

### Relational Algebra:

$\sigma \text{ userid} = 304012970, \text{ CreatedAt} \geq (\text{Now}() - \text{INTERVAL } 1 \text{ DAY}) (\text{Twitter})$

## 4. How many tweets have this user posted in the past 24 hours?

### SQL Query:

Select Count(\*) FROM Twitter where userid= 34012970;

### Relational Algebra:

$\sigma \text{ Count}(\text{userid}=34012970) (\text{Twitter})$

## 5. When did this user join Twitter?

### SQL Query:

Select userid, CreatedAt from TwitterUser;

### Relational Algebra:



$\sigma$  userid, CreatedAt (TwitterUser)

## 6. What keywords/ hashtags are popular?

### SQL Query:

Select Hashtags, count(\*) from tweet\_tags group by Hashtags order by count(\*) desc;

### Relational Algebra:

$\sigma$  count(Hashtags) (tweet\_tags)

## 7. What tweets are popular?

### SQL Query:

Select TwitterText, TwitterID, retweetCount from Twitter order by retweetCount desc;

### Relational Algebra:

$\sigma$  (TwitterText, TwitterID, retweetCount)(Twitter)