ASSIGNMENT 3: Gathering, Scraping, Munging and Cleaning Data

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

Use Cases:

1]Use Case: Count of students in each department

Description: Person requires count of students in each department

Actor: Person

Precondition: student must be enrolled in at least one course from one department

Steps:

Actor action – Person will enter DepartmentID

System Responses – Return count of students under the entered departmentID

Post Condition: Person can see number of students in each department

Error: Invalid DepartmentID

SQL Query:

```
SELECT s.DeptName, Count(FirstName) FROM Student s INNER JOIN College c ON
s.DeptID=c.DeptID
GROUP BY s.DeptID, c.DeptID ORDER BY Count(FirstName);
```

2]Use Case: Which college has more number of student head

Description: person requires max number of student head in college

Actor: person

Precondition: Every college should have at least one student head

Steps:

Actor action – Person will enter College name

System Responses – Return maximum count of student head under college

Post Condition: search for colleges with highest number of student head

Error: No college has student head

SQL Query:

```
SELECT College,Count(College) AS NumberOfStudentHeads FROM Student s INNER JOIN
StudentClub sc ON s.StudentID=sc.StudentID
GROUP BY s.College ORDER BY College;
```

Example for use case 2:

```
SELECT College, s.StudentID FROM Student s INNER JOIN StudentClub sc ON
s.StudentID=sc.StudentID WHERE s.College="School of Law";
```

3]Use Case: Number of students who got co-op

Description: Person requires number of students wo got co-op

Actor: Person

Precondition: every student must have studentID

Steps:

System Responses – Return number of students who got co-op under that jobID

Post Condition: Get number of students on co-op

Error: Invalid JobID

SQL Query:

```
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;
```

4]Use Case: Which department got maximum placements

Description: Person requires maximum count of placements

Actor: Person

Precondition: every student must be enrolled in at least one course in a department

Steps:

Actor action – Person will enter departmentID

System Responses – Return number of students got placed under that departmentID

Post Condition: Get maximum number of students got placed

Error: Invalid departmentID

SQL Query:

SELECT s.DeptID, s.DeptName, COUNT(sj.JobID) AS NumberOfStudents

FROM Student s INNER JOIN StudentJob sj ON s.StudentID=sj.StudentID GROUP BY s.DeptID ORDER BY COUNT(sj.JobID) DESC;

5]Use Case: Average salary offered to students in each department

Description: person search for average salary of students

Actor: person

Steps:

Actor action – Persons will enter departmentID

System Responses – Average salary of students in each department will be displayed

Error: invalid departmentID

SQL Query:
SELECT s.DeptName, 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.deptname;

6]Use Case: Maximum number of JobOpenings

Description: person requires maximum JobOpenings

Actor: person

Precondition: every job must have unique jobID

Steps:

Actor action – Person will enter job title to get number of openings

System Responses – Return number of job openings

Post Condition: Get maximum number of job openings

Error: No job openings found

SQL Query:

```
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;
```

7]Job Count for each intake

Description: Person requires job count for each semester intake

Actor: Person

Precondition: Every intake must have at least one job

Steps:

Actor action – Person will search for job count

System Responses – Return number of job for each intake

Post Condition: Get job count

Error: No jobs found

SQL Query:

```
SELECT s.IntakeSemester, s.IntakeYear, COUNT(sj.StudentID) FROM Student s
INNER JOIN StudentJob sj ON s.StudentID=sj.StudentID GROUP BY
s.IntakeSemester, s.IntakeYear;
```

8]Use Case: Booming Industry

Description: Person requires companies in descending order in terms of students placed

Actor: person

Precondition: every company must have some students placed

Steps:

Actor action – Person will search for industries

System Responses – Return growing industries

Post Condition: Get knowledge of growing industries in terms of placements

Error: No placements

SQL Query:

```
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;
```

9] Which company has the second highest salary?

Description: Student requires details of company/industry with second highest salary

Actor: student

Precondition: every company must have a salary estimate

Steps:

Actor action – Student will look for company/industry with second highest salary

System Responses – Returns number maximum salary estimate from InternshipsAndCoop

SQL Query:

```
SELECT MAX(SalaryEstimateUpper) from InternshipsAndCoop  WHERE SalaryEstimateUpper
< (SELECT Max(SalaryEstimateUpper) FROM InternshipsAndCoop);
```

10]Use Case: Students who got internship and on-campus job

Actors student

Steps:

Actor action – student will look for count of students with internships and on-campus

System Responses – Return student ID of students who have on-campus and internship

SQL Query:

```
SELECT sj.StudentID FROM StudentOnCampus soc INNER JOIN StudentJob sj ON
soc.StudentID=sj.StudentID;
```