

CS 411 Project Stage 2 Report – College Employment & Salary Analysis Platform

Group50-Hajimi

1. Entity Descriptions and Assumptions

Entity	Description / Assumption
User	This entity is used to present user information. Each user can bookmark programs and compare them. Each user can have preferences on job, location and major.
University	This entity is used to present university information. Each university can provide multiple programs. Each university has its Name, Location, and Region. Each university has a unique UniversityID
Major	This entity is used to present major information like Economics, CS, and so on. Each major can be provided by multiple programs. Each major has a unique major ID.
Program	This entity is used to present what programs each university provides. Each program is related to a major. Each program must be provided by one university, and each university can provide multiple programs. Each program can have a unique ProgramID.
Job	This entity is used to present the job information. Each job has a unique JobID. Each job contains a job title, company, company location, and average salary.
Comparison	This entity is used to store the comparisons each user selected to compare. Each comparison stores two selected programs by using ProgramID1 and ProgramID2. Each comparison must be assigned to one user, and each user can create multiple comparisons. Each comparison has a unique ComparisonID.

2. Relationship Explanations

Relationship	Type	Explanation
User-Comparison	1-to-Many	A user can create many comparisons, but each comparison belongs to one user based on UserId.
User-Program (Bookmarks)	Many-to-Many	Users can bookmark many programs, and each program can be bookmarked by many users. We will implement it by creating a Bookmark relation.
University-Program	1-to-Many	Each university offers many programs, but each program is offered by exactly one university.
Major-Program	1-to-Many	Each major can appear in many programs, but each program is tied to one major.
Major-Job (relates-to)	Many-to-Many	A major can lead to multiple job types, and a job can be related to multiple majors. We will implement it via the relates-to relation.
Comparison-Program (creates)	Many-to-Many	Each comparison can include multiple programs, and each program can appear in multiple comparisons. We will implement it via the creates relation.
Job-User (prefers)	Many-to-Many	Users can express interest in multiple jobs, and each job can be preferred by multiple users. We will implement it via the prefers relation.

3. Normalization Justification

Our database design adheres to 3NF, ensuring data integrity and minimal redundancy:

1. 1NF: Every column holds a single value(University.Region is one region, Program.MedianSalary is one number and User.PreferredLocation is a single). There will be no repeating groups.

2. 2NF: In our tables, for example, Major(MajorId, JobId) have no extra attributes that would depend on one side of the key. Base tables use a single-column primary key, so partial dependency isn't an issue there.
3. 3NF: University details (name, region, tuition, graduation rate) live only in University, not in Programs table or User table. Program data will stay only in the Program table not duplicated in Comparison table or User table.
4. M:N: All many-to-many relationships are split into things like Major-to-Job, User-to-Job, Comparison-to-job, User-to-program which will preserve the integrity.

4. Logical Design Relational Schema

SQL

```
User(  
    UserID: INT [Primary Key],  
    Username: VARCHAR(255),  
    Email: VARCHAR(255),  
    PasswordHash: VARCHAR(255),  
    PreferredMajor: INT [Foreign Key to Major.MajorID],  
    PreferredLocation: VARCHAR(255),  
    PreferredJob: INT [Foreign Key to Job.JobID])
```

```
University(  
    UniversityID: INT [Primary Key],  
    Name: VARCHAR(255),  
    Location: VARCHAR(255),  
    Region: VARCHAR(255),
```

Tuition: INT,

Major(

MajorID: INT [Primary Key],

MajorName: VARCHAR(255),

Field: VARCHAR(255))

Program(

ProgramID: INT [Primary Key],

UniversityID: INT [Foreign Key to University.UniversityID],

MajorID: INT [Foreign Key to Major.MajorID],

MedianSalary: INT,

DegreeType: VARCHAR(255))

Job(

JobID: INT [Primary Key],

JobTitle: VARCHAR(255),

Company: VARCHAR(255),

Location: VARCHAR(255),

AvgSalary: INT)

Comparison(

ComparisonID: **INT** [Primary Key],

UserID: **INT** [Foreign Key to User.UserID],

ProgramID1 [Foreign Key to Program.ProgramID],

ProgramID2 [Foreign Key to Program.ProgramID])

Bookmark(

UserID: **INT** [Foreign Key to User.UserID],

ProgramID: **INT** [FK to Program.ProgramID])

ComparisonProgram(

ComparisonID: **INT** [Foreign Key to Comparison.ComparisonID],

ProgramID: **INT** [Foreign Key to Program.ProgramID])

MajorJob(

MajorID: **INT** [Foreign Key to Major.MajorID],

JobID: **INT** [Foreign Key to Job.JobID])

UserJobPreference(

UserID: **INT** [FK to User.UserID],

JobID: **INT** [FK to Job.JobID])

