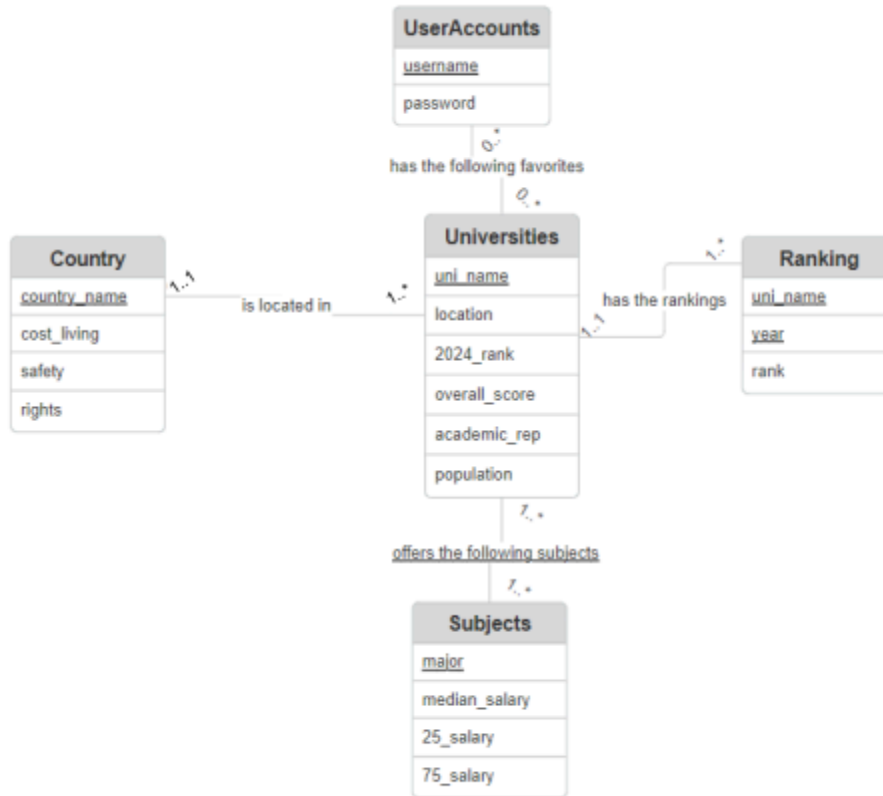


UML Diagram



Entity Assumptions

- UserAccounts: each UserAccount can favorite any number of universities.
- Country: for each Country, there is at least 1 University located inside the country.
- Subjects: for each Subject, there is at least 1 University that offers that subject.
- Ranking: for each Ranking, there is exactly 1 University associated with the ranking..
- Universities: for each University, it can be saved by any number of UserAccounts, located in exactly 1 Country, offers at least 1 Subject, and has many Rankings.

Relationship Descriptions

- has the following favorites: each UserAccount can favorite any number of Universities in their favorites, while each University can be favorited by any number of UserAccounts
- is located in: each Country has at least 1 University located inside of it, while each University is located inside of exactly 1 Country
- offers the following subjects: each Subject is offered by at least 1 University, while each University offers at least 1 Subject
- has the rankings: each Ranking is associated with exactly 1 University, while each University has many years of Rankings

Relationship Cardinalities

- UserAccount to University: many-to-many (many user accounts favorite many universities)
 - UserAccount to Favorites: one-to-many (one user account has many favorites)
 - Favorites to University: many-to-one (many favorited universities by different user accounts have one university)
- Country to University: one-to-many (one country has many universities)
- University to Subject: many-to-many (many universities can offer many subjects)
 - University to OfferedSubjects: one-to-many (one university has many offered subjects)
 - OfferedSubjects to Subjects: many-to-one (many offered subjects by different universities have one subject)
- University to Ranking: one-to-many (one university has many rankings)

Functional Dependencies

Users_Table: Username → Password

Favorites_Table: Users_Table.Username, Universities_Table.Name

Universities_Table: Name → Countries_Table.Name, OverallScore, TeachingScore, ResearchScore, EmployerScore, Population

Countries_Table: Name → LivingCostScore, SafetyScore, RightsScore, ClimateScore

Subjects_Table: Name → 25_Salary, 50_Salary, 75_Salary

OfferedSubjects_Table: Universities_Table.Name, Subjects_Table.Name

Rankings_Table: Universities_Table.Name, Year → Ranking

Normalization

Users(Username, Password)

Favorites(Users.Username, Universities.Name)

Universities(Name, Countries.Name, OverallScore, TeachingScore, ResearchScore, EmployerScore, Population)

Countries(Name, LivingCostScore, SafetyScore, RightsScore, ClimateScore)

Subjects(Name, 25_Salary, 50_Salary, 75_Salary)

OfferedSubjects(Universities.Name, Subjects.Name)

Rankings(Universities.Name, Year, Ranking)

In every table, the underlined attribute is the super key and the relations can all be written in the form $A \rightarrow BC \dots$, where A is the super key. Therefore, the tables are in BCNF.

We chose BCNF because it is a stronger and more restricted form of normalization. Additionally, all of our dependencies are still preserved, despite that being a potential drawback of BCNF.

Relational Schema

Table-Users(

Username: VARCHAR(50) [PK],

Password: VARCHAR(50)

)

Table-Favorites(

Username: VARCHAR(50) [PK] [FK to Users.Username],

University: VARCHAR(50) [PK] [FK to Universities.Name]

)

Table-Universities(

Name: VARCHAR(50) [PK],

Location: VARCHAR(50) [FK to Countries.Name],

OverallScore: DECIMAL(4, 1),

TeachingScore: DECIMAL(4, 1),

ResearchScore: DECIMAL(4, 1),

EmployerScore: DECIMAL(4, 1),

Population: INT

)

Table-Countries(

Name: VARCHAR(50) [PK],

LivingCostScore: INT,

SafetyScore: INT,

RightsScore: INT,

ClimateScore: INT

)

Table-Subjects(

Name: VARCHAR(50) [PK],

25_Salary: INT,

50_Salary: INT,

75_Salary: INT

)

Table-OfferedSubjects(

University: VARCHAR(50) [PK] [FK to Universities.Name],

Subject: VARCHAR(50) [PK] [FK to Subjects.Name]

)

```
Table-Rankings(  
    University: VARCHAR(50) [PK] [FK to Universities.Name],  
    Year: INT [PK],  
    Ranking: INT  
)
```