

I Entity Definitions

COUNTRY:

A country is the country the users, universities and subdivisions belong to. Eg Russia. The country is registered every time a user from the country identifies it, and it is never deleted from the system.

SUBDIVISIONS:

A subdivision is the province/state/part of a country. Eg Alberta. It is a weak entity as it is dependent on the Country. A state/province/part cannot exist without a country. The subdivision is registered every time a user/university from the subdivision identifies it, and it is never deleted from the system.

UNIVERSITY:

An university is an institution that a user associates itself to. Eg University of Toronto. It is a weak entity as it is dependent on the country it is on. An university instance is created every time any user, associated from it, requests for it, and is never deleted unless the university itself goes under.

CONTEST:

A contest is a competition which consists of set of problems and various users join teams or play solo to solve those problems within given time limit. Eg Montana Tech ACM. It is created by the users in the system and ends when its past its time limit.

PROBLEM:

A problem is a coding problem that is to be solved by the users using any programming language. The problems are already created by the database developers from various sources. Each problem accepts unlimited solutions. Eg A1 Paper is a problem from KTH Challenge 2015 created by Ulf Lundstrom.

TEAM:

A team consists of users that compete in contests. Each team can contain one or many Users. A team is created anytime a user wants to compete in a contest. After creating a Team he can then add more users to his team. Eg Team1 will take part in contest Montana Tech ACM. It is weak entity as its existence is dependent on a contest.

USERS:

A user is a person who has signed up to the open.kattis site. The person has to provide some personal information to sign up which will then create an account for themselves. The user can delete their account whenever they want to. After signing up, the user will be able to take part in contests and solve problems that are available. Eg Thomas Shelby

USER_HAS_TEAM:

It is an associative entity between users and team.
This should have been a many-to-many relationship.

SUBMISSIONS:

A submissions contains information about every solution that a user submits for a Problem. Not a weak entity but is dependent on the user. As submission object is Created everytime a user makes a submission for any problem. A submission can be Updated by the user.

JOB:

A job is a job posting by a company looking to hire users, an example being a company like Paradox looking to hire C++ programmers. This entity is only related to corporations and doesn't interact directly with users, contests, problems, and is simply a redirection to a different website. This entity will be created when a job is posted and deleted when the position(s) are/is filled.

II Attribute Definitions (don't need to show foreign keys thankfully)

Attributes of Country:

a) C_Name

Example: "Canada"

This is a required attribute.

Type: VARCHAR(45)

This is a primary key.

b) Ranking

Example: 12

This is the ranking of this country against all others.

This attribute is required.

Type: Integer

The domain for ranking is 1 to the total number of countries.

b) C_Score

Example: 11.2

This is the average score of users from the country.

This attribute is required.

Type: DECIMAL(1)

The domain for this attribute is the lowest user score of the country to the highest score.

This is a derived attribute.

c) Flag

This attribute is required.

Type: BLOB

Attributes of Subdivisions:

a) Sb_Name

Example: "New Brunswick"

A subdivision of a country can be a province, state, etc.

This is a required attribute.

Type: VARCHAR(45)

This is part of a composite primary key.

b) Sb_Ranking

Example: 3

This is the ranking of this subdivision against all others.

This attribute is required.

Type: Integer

The domain for ranking is 1 to the total number of subdivisions.

b) Sb_Score

Example: 5.2

This is the average score of users from the subdivision.

This attribute is required.

Type: DECIMAL(1)

The domain for this attribute is the lowest user score of the subdivision to the highest score.

This is a derived attribute.

c) Sb_Flag

This attribute is required.

Type: BLOB

Attributes from University

a) U_Name

Example: "University of New Brunswick"

This is a required attribute.

Type: VARCHAR(45)

This is a primary key.

b) U_Ranking

Example: 17

This is the ranking of this university against all others.

This attribute is required.

Type: Integer

The domain for ranking is 1 to the total number of universities.

c) U_Score

Example: 6.2

This is the average score of users from the university.

This attribute is required.

Type: DECIMAL(1)

The domain for this attribute is the lowest user score of the university to the highest score.

This is a derived attribute.

d) U_Flag

This attribute is required.

Type: BLOB

Attributes from Contest:

a) Con_Name

Example: "Paradox Contest"

This is the name of the contest being hosted.

This attribute is required.

Type: VARCHAR(45)

This attribute is a primary key.

b) Start_Time

Example: "April 16th, 2023, 6pm"

This is the starting time and date of the contest.

This is a required attribute.

Type: DATETIME

c) Length

This is the length of time that the contest will run for.

This is a required attribute.

Type: DATETIME

d) Remaining

This is the remaining time of the contest.

This is a required attribute.

Type: DATETIME

This is a derived attribute.

e) End_Time

Example: "April 16th, 2023, 6pm"

This is the ending time and date of the contest.

This is a required attribute.

Type: DATETIME

Attributes of PROBLEM

a) P_ID

Example: "3dprinter"

Unique identification code for the problem.

This attribute is required.

Type: VARCHAR(45)

This attribute is a primary key.

b) P_Name

Example: "3D Printer Statues"

Title of the problem.

This attribute is required.

Type: VARCHAR(45)

c) CPU_Time_Limit

Example: 1 second

Amount of time to be taken to run the solution to the problem.

This attribute is optional.

Type: INT

d) Memory_Limit

Example: 1024 MB

Memory space to be used by the solution for the respective problem.

This attribute is optional.

Type: INT

e) P_Desc

Description of the problem including Information on the given input and required output.

This attribute is required.

Type: VARCHAR(45)

f) Difficulty

Example: 2.0

This attribute is required.

Type: DECIMAL(1)

g) Author

Example: Unnar Freyr

The name of the creator of the problem.

This attribute is required.

Type: VARCHAR(45)

h) Source

Example: KTH Challenge 2017

Where the problem came from if it is related to another organization or other material.

This attribute is required.

Type: VARCHAR(45)

i) License

The legal licensing of the problem.

This attribute is required.

Type: VARCHAR(45)

Attributes from Team:

a) T_Name

Example: "The blueberries"

This is a required attribute.

Type: VARCHAR(45)

This is part of a composite primary key.

b) Ranking

Example: 2

This is the rating of this team compared to all others in the contest.

This is a required attribute.

Type: int

c) Score

Example: 6.2

This is the average score of users from the team.

This attribute is required.

Type: DECIMAL(1)

The domain for this attribute is the lowest user score of the team to the highest score.

This is a derived attribute.

Attributes from Users:

a) Username

Example "TheLegend27"

This is the username used by the user.

This is a required attribute.

Type: VARCHAR(45)

This is a primary key.

b) Us_Name

Example "Corey Woodman" or "Sumitr Banik"

This is the real name of the user.

This is a required attribute.

Type: VARCHAR(45)

c) Us_Ranking

Example: 17364

This is the worldwide ranking of the user against all others.

This is a required attribute.

Type: int

The domain for this attribute is 1 to the current number of all users.

d) Profile_Picture

This is a picture selected by the user.

This is an optional attribute.

Type: BLOB

e) Preferred_Language

Example: "C++"

This is the preferred programming language of the user.

This is an optional attribute.

Type: VARCHAR(45)

f) Preferred_Timezone

Example "UTC"

This is the preferred time zone of the user.

This is an optional attribute.

Type: VARCHAR(45)

g) Default_Prog_Lang

Example: "Java"

This is the default programming language for problems the user wants to solve.

This is an optional attribute.

Type: VARCHAR(45)

h) U_Password

This is the password used to login to the user account.

This is a required attribute.

Type: VARCHAR(45)

i) Email

This is the email used to login to the user account and for contact purposes.

This is a required attribute.

Type: VARCHAR(45)

j) User_Score

This is the overall user score defaulted to 1.0

This is a required attribute.

Type: DECIMAL(1)

Attributes from Submissions:

a) Time_Taken

Example: 0.02 ms

This is the time taken for a program to execute.

This is a required attribute.

Type: DECIMAL(2)

b) Memory_Used

Example: 37 kb

This is the memory used by a program while executing.

This is a required attribute.

Type: int

c) Date

This is the date the submission is made.

This is a required attribute.

Type: DATETIME

d) Language

This is the programming language used to solve the problem.

This is a required attribute.

Type: VARCHAR(45)

Attributes of JOB:

a) Job_Title

Example: "Do you want to use your C++ to help make great games?"

This is the title/description for any user that may be interested in the job.

This attribute is required.

Type: VARCHAR(45)

This value is a primary key.

b) Company_Name

Example: "Paradox"

This attribute is required.

Type: VARCHAR(45)

c) Company_Logo

This attribute holds the company logo that posted the job offering.

This is an optional attribute.

Type: BLOB

III Relationship Definitions

1. makes

Relation between USER and SUBMISSIONS. It's a one-to-many relation. A user can make several submissions for a single problem. For eg, User TMuller can submit a solution for problem "Hello World!". He can submit several solutions for that problem. He can also update his solution. The relationship is removed when the user deletes its submission.

2. for

Relation between SUBMISSIONS and PROBLEM. It's a one-to-many relation. Every solution for a problem is submitted. The relationship is removed when the user deletes its submission.

3. makes

Relation between TEAM and SUBMISSIONS. It's a one-to-many relation. A team can make several submissions for a single problem in the contest. The relationship is removed when the team deletes its submission.

4. participates

Relation between TEAM and CONTEST. It's a one-to-many relation. A user joins a contest, and then later adds members to his team. A user is allowed to participate in several contests at the same time. After the contest ends, the team is deleted from the respective contest.

5. In

Relation between PROBLEM and CONTEST. It's a one-to-many relation. Every contest consists of several problems. The site gives the user the option to let the system choose the problems automatically. The relationship is removed at the end of the contest.

6. has

Relation between USERS and TEAM. Supposed to be a many-to-many relation. Every user needs to be in a team to participate in contest. Each team can either consist of one or many users. The relationship is removed at the end of the contest.

7. belong from

Relation between USERS and COUNTRY. It's a zero-to-many relation. Every user has to specify the country they belong from. But the user has the option to not specify the country. The relationship is removed when user decides to remove the country from their profile.

8. live in

Relation between USERS and SUBDIVISIONS. It's a zero-to-many relation. It's similar to the 'belong from' relation. Every User has the option to specify the state/province/part of a country they're from or they can choose not to specify as well. The relationship is removed when user decides to remove the subdivision from their profile.

9. Studies in/ graduated

Relation between USERS and UNIVERSITY. It's a zero-to-many relation. Similar to 'belong from' relation. Every user has the option to specify the university they study in or have graduated from or choose not to specify as well. The relation is removed when the user decides to remove the university from its profile.

10. Is a subdivision of

Relation between SUBDIVISIONS and COUNTRY. It's one-to-many relation. It's a one-to-many relation. Every subdivision is a part of a country, but not every country has a subdivision. There are several subdivisions from one single country. The relation is created whenever a subdivision from a specific country is added to the database. And is not deleted unless the subdivision in the globe is changed.

11. Is located in

Relation between UNIVERSITY and SUBDIVISION. It's a one-to-many relation. It's created whenever a user adds a university that is located in a subdivision, and is not removed unless the university is deleted from the database.

12. Is located in

Relation between UNIVERSITY and COUNTRY. It's a one-to-many relation. This relation was created cause of the possibility of countries without subdivisions. Therefore the university is directly located in the country. This relationship is created when a university refers itself directly by the country. For eg. National University of Singapore. And similarly the relation is not removed unless the university is deleted from the database.