LEETCODE – Database Design and Data Flow

ARLRASE Assignment – LEETCODE Data base design and flow

Author – Balpreet Kaur

# ENTITIES

1. **LEET User**: LEET\_USER is an entity which is created when a user creates a login user with the LEETCODE. Basically, this entry point of the user when a user creates and login to the Leet code application. The entity would have attributes like:

|  |  |
| --- | --- |
| Attribute: | Comments |
| userId | Primary key |
| User\_name | User name, would be unique and this is the user name with which user login |
| email | Email address, this would be unique and this the attribute with which user is registered to. So that user can recover the password |
| Password | Password of the user |
| Leet\_joining\_dt | Date on which the user joined the leet code application |

1. **LEET problem**: LEET\_Problem, is the entity which would contain all various questions of different categories.

It would have attributes like:

|  |  |
| --- | --- |
| Attribute: | Comments |
| problemId | Primary key |
| Problem\_title | In leet code, every question starts with brief statement |
| Problem\_Description | This is the actual problem statement |
| Problem category | This is the category of the problem, that is what is the subject of the question. Ex: Python, SQL, C, C++, etc. |
| Difficulty Level | It can have level: basic, medium, high , very difficult |
| Problem\_creation\_dt | The day the problem statement was created |

1. Submission:

This table would hold information when the user actually submits an answer for a problem statement. With the attribute userID and problemID one can identify which user submitted answer for which problem statement

|  |  |
| --- | --- |
| Attribute: | Comments |
| Submission\_id | Primary key |
| UserId | The foreign key to LEET\_USER, which will be helpful in evaluating which user has submitted |
| problemId | The foreign key to LEET\_Problem, which will be helpful in evaluating submission is made against which problem statement |
| submissionCode | The actual code or answer submitted by the user for a probmem statement |
| SubmissionResult | This would attain values like solved or not solved |
| SubmissionTime | Duration in which the problem statement was solved |

1. Discussion:

Each problem statement are discussed by various users. Topic of discussion can be anything from right answer, or better way to write a particular code.

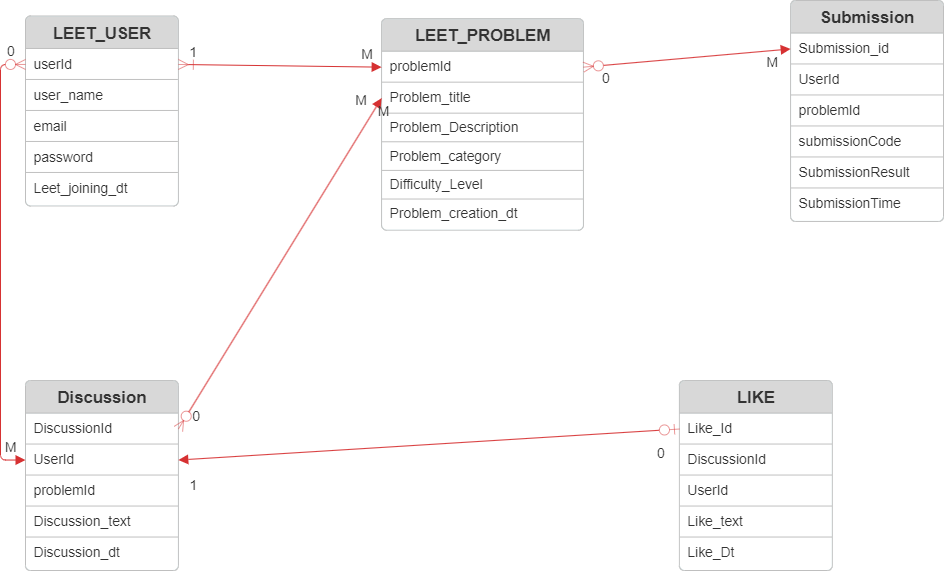
|  |  |
| --- | --- |
| Attribute: | Comments |
| DiscussionId | Primary key |
| UserId | The foreign key to LEET\_USER, which will be helpful in evaluating which user has discussed |
| problemId | The foreign key to LEET\_Problem, which will be helpful in evaluating discussion is for which problem statement |
| Discussion\_text | Actual discussion text which would be posted on the leet code against a problem statement by a user |
| Discussion\_dt | Date on which discussion was posted by the user |

1. Like:

A discussion can be liked by user.

|  |  |
| --- | --- |
| Attribute: | Comments |
| Like\_Id | Primary key |
| DiscussionId | The foreign key to Discussion table |
| UserId | The foreign key to LEET\_USER, which will be helpful in evaluating which user has liked or not like |
| Like\_text | This can be value like Like or Unlike, which means a user has liked or unliked a discussion text |
| Like\_Dt | The date on which discussion was liked or unliked. |

# ER Diagram/Database design



# Data workflow design

