

Regras de mapeamento

Modelagem GitHub

Disciplina: Fundamentos de Banco de Dados

Grupo: Nicolle Favero e Matheus Azambuja

Professora: Karin Becker

Mapeamento de Entidades

Com base na modelagem conceitual e nas especificações no dicionário de dados decidimos optar pelas seguintes regras.

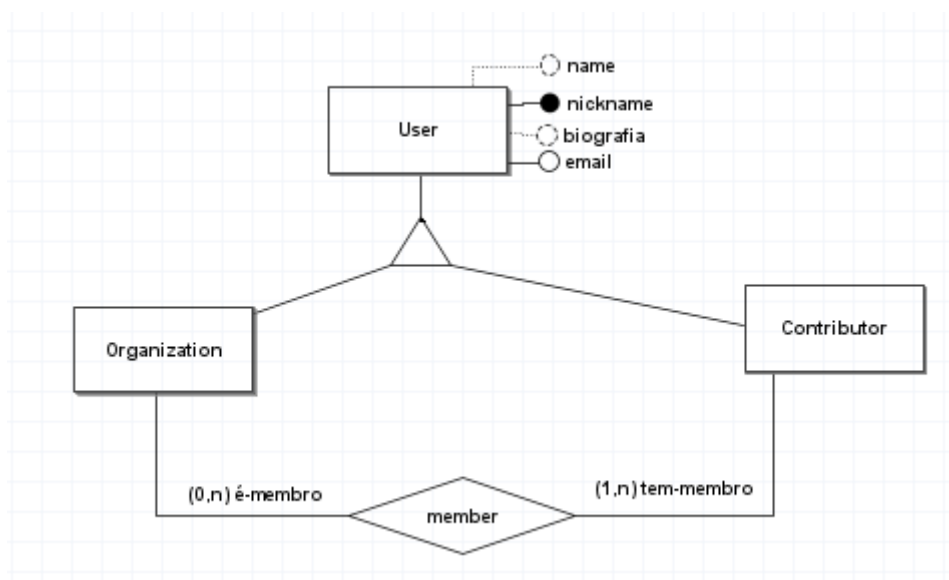
Para melhor apresentação das regras seguiremos a seguinte sequência de transformação do modelo ER em um modelo relacional:

- Entidades e respectivos atributos;
- Generalizações/especializações;
- Relacionamentos e respectivos atributos;

Notações:

- **Chave primário em negrito;**
- *Chave estrangeira em itálico;*
- Atributo opcional destacado com *;
- Atributo único destacado com sublinhado;

Entidade User: especialização



Considerando a especialização que a entidade User possui, as distintas relações que Organization e Contributor participam e pelas operações que serão realizadas nessas tabelas decidimos transformar a especialização nas seguintes tabelas:

Users (**idUser**, name*, nickname, biografia*, email);

Contributors (**idContrib**)

references Users (idContrib);

Organizations (**idOrg**)

references Users (idOrg);

Members (**idContrib**, **idOrg**)

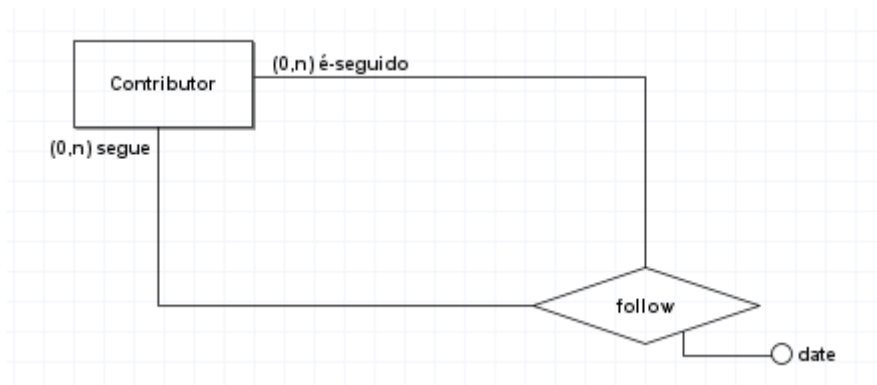
references Contributor (idContrib),

references Organization (idOrg);

A outra opção era apenas criar a tabela User na qual teríamos que colocar, além dos atributos mostrados acima, um atributo chamado type o qual deveria estar assinalado como “contributor” ou “organization”. Não optamos por essa tabela, pois teríamos que ter algum mecanismo de verificação se esse usuário era contribuidor ou organização nas tabelas de relacionamento entre usuários e repositórios e de usuários para usuários. Decidimos simplificar optando por fazer as três tabelas distintas acima.

(Para melhorar as relações das tabelas, criamos um atributo ID. Esse atributo será usado no lugar do nickname, que se transformará em uma chave alternativa).

Auto Relacionamento Follow:



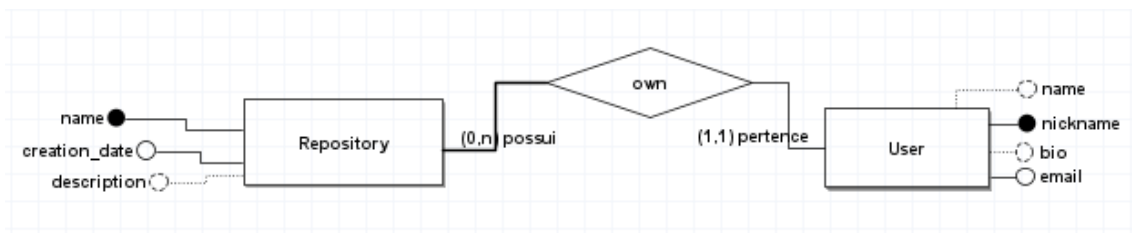
Criado uma tabela para o relacionamento por se tratar de um Relacionamento N-M:

Follows (**idFollower**, **idFollowed**, date)

references Contributors (IdFollower),

references Contributors (idFollowed);

Entidade Repository



Mapeado para:

Esse mapeamento foi feito considerando a resolução do Relacionamento 1-N e a Entidade Fraca.

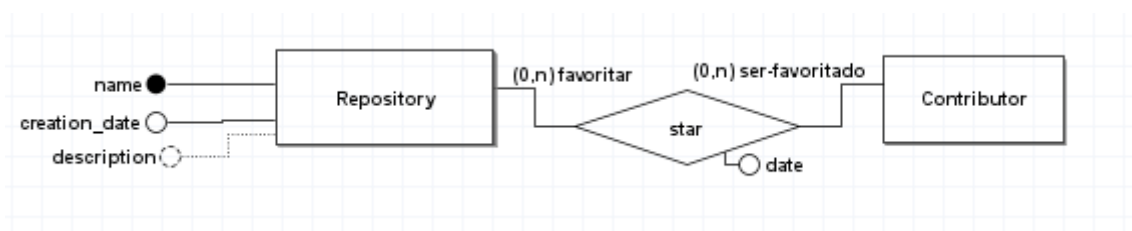
Repositories (**idRepo**, idUser, name, description*, creationDate)

references Users (idUser);

Foi criado um atributo idRepo para melhorar as relações entre as tabelas de se relacionam com Repositories.

Escolhemos essa resolução para tabela Repository pois ela participará de muitos relacionamentos posteriormente.

Relação Star



Mapeado para:

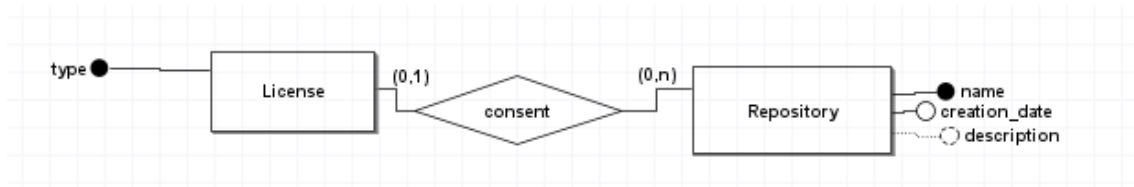
Stars (**idContrib**, **idRepo**, date)

references Contributor (idContrib),

references Repository (idRepo);

Relacionamento N-M, mapeamento feito seguindo a padronização feita segundo aula, pois há somente uma alternativa para tal. Por queremos representar uma timeline e esse relacionamento participa dela, essa é a justificativa para modelarmos dessa forma.

Entidade License e relação com Repository



Mapeado para:

License (**idLicense**, type);

Repositories (**idRepo**, idUser, name, description*, creationDate, idLicense)

references Users (idUser),

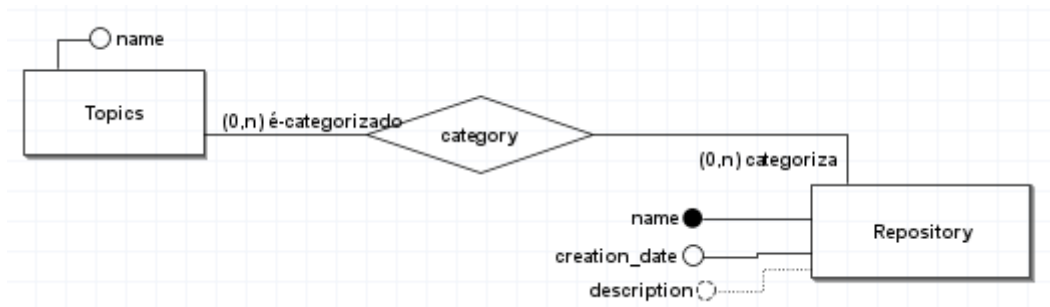
references Licence (idLicense);

Alterações feitas:

- Repository recebeu novo atributo: IdLicense
- Repository recebeu novo atributo: description

Decidimos por esse mapeamento pois o Relacionamento é 1-N, portanto, representando assim, o desejado.

Entidade Topics e relação com Repository



Criado uma tabela para o relacionamento por se tratar de um Relacionamento N-M:

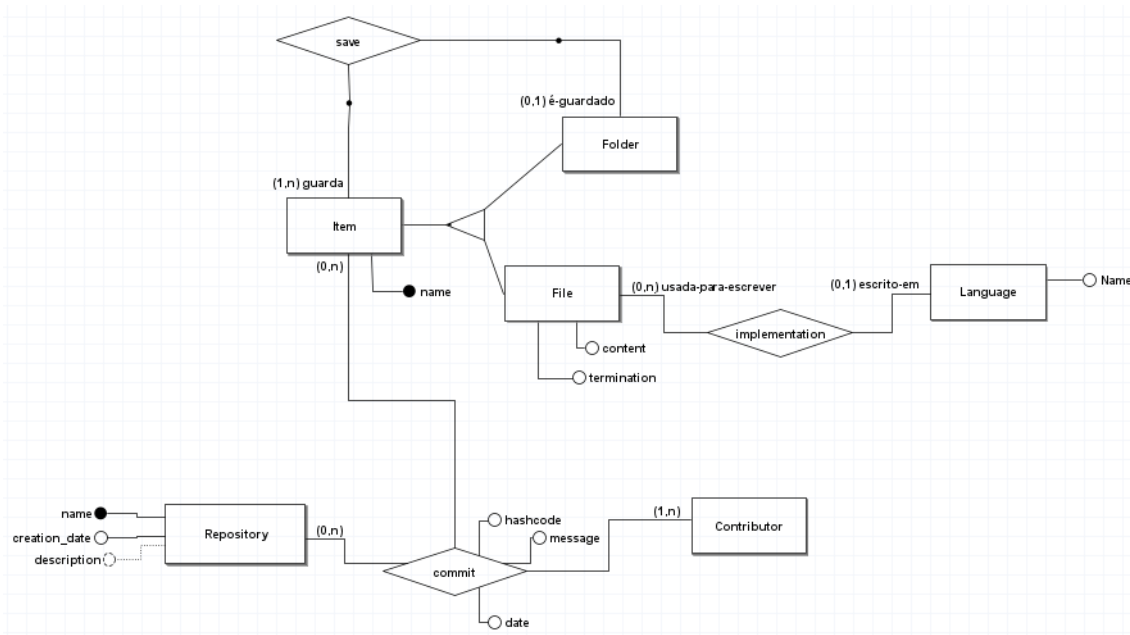
Topics (**idTopic**, name)

Category (**idRepo**, **idTopic**)

references Repository (idRepo),

references Topics (idTopic);

Estruturação do repositório



Entidade Item: especialização e relação com Repository

Mapeado para:

Items (**idItem**, *idRepo*)

references Repository (idRepo);

Folders (**idFolder**, *nameFolder*, *idRepo*)

references Items (idFolder, idRepo);

Files (**idFile**, *idRepo*, *nameFile*, *content*, *termination*)

references Item (idFile, idRepo);

Optamos por essa alternativa considerando o relacionamento de Item com Repository. Desse modo, também, a consulta de quantos arquivos um repositório tem ou quantas pastas ele possui fica trivial. Ainda, a entidade File tem muitas características suas que, em uma tabela Items “mista” poderia resultar em vários campos nulos quando o Item fosse um Folder.

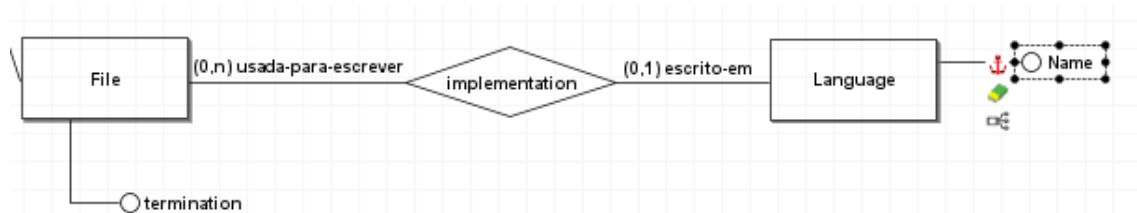
Saves (*idFolder**, **idItem**)

references Items (idItem),

references Folders (idFolder);

O Relacionamento 1-N (Opcional do “lado 1” / Opcional do “lado N”) save foi modelado dessa forma para facilitar a consulta posteriormente para obter quantos arquivos uma pasta contém. Assim sendo, fica fácil realizar tal consulta.

Entidade Language:



Mapeado para:

Languages (**idLang**, name)

Implementations (**idFile**, idLang)

references Files (idFile),

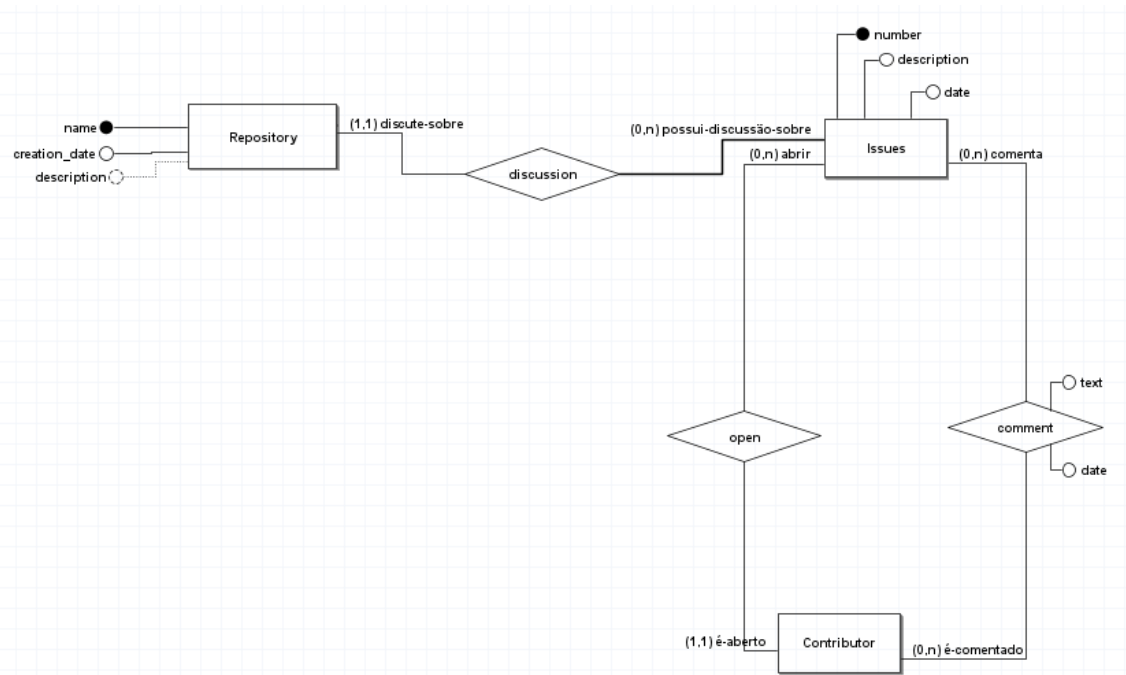
references Languages (idLang);

Alterações feitas:

- Languages recebeu novo atributo: name

Utilizamos esse mapeamento pensando no futuro: posteriormente será implementado uma consulta importante com o objetivo de obter o percentual das linguagens utilizadas em um repositório. A relação entre Files e Languages modelado dessa forma, visa aumentar o desempenho dessa consulta em específico.

Entidade Issue e seus Relacionamentos



Relação Open e Discussion

Mapeado para:

Issues (**idIssue**, description, date, **idRepo**, idContrib)

references Repositories (idRepo),

references Contributor (idContrib);

Esse mapeamento foi feito considerando a resolução do Relacionamento 1-N e a Entidade Fraca. Dessa forma, cada Issue será única dentre as Issues do repositório.

Adicionado o atributo (idContrib) em Issues que representará o contribuidor que abriu o Issue, representando assim o relacionamento.

Relação Comment

Mapeado para:

Comments (**idContrib**, **idIssue**, **idRepo**, date, text)

references Contributors (idContrib),

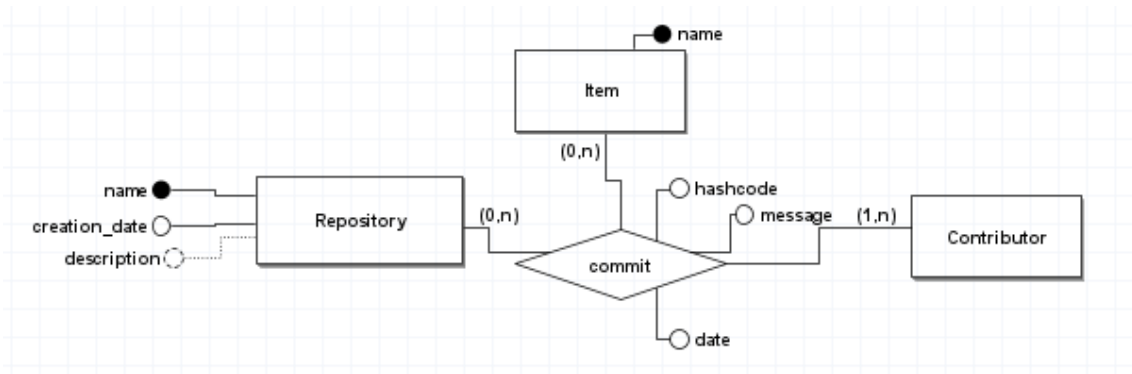
references Issues (numIssue)

references Repositories (idRepo);

Relacionamento N-M, mapeamento feito seguindo a padronização feita segundo aula, pois há somente uma alternativa para tal. Por queremos

representar uma timeline e esse relacionamento participa dela, essa pode ser uma justificativa para modelarmos dessa forma. É necessário guardar o nome do repositório em que o Issue se encontra, pois cada Issue tem um número único dentro de um repositório.

Relação Commit



Mapeado para:

```
****Commit (hashCode, idItem, idRepo, idContrib, message, date)
    references Item (idItem),
    references Repositories (idRepo),
    references Contributors (idContrib);
```

Mapeamento de entidade ternária (N:N:N), uma única alternativa. Foi criado uma tabela para o relacionamento Commit para identificá-la unicamente.