Universidade do Estado de Santa Catarina - UDESC
Centro de Ciências Tecnológicas - Departamento de Ciência da Computação
Professora: Dra. Rebeca Schroeder Freitas
Disciplina: Banco de Dados I - 2022/2

Projeto de Banco de Dados Relacional: Sistema de Finanças Pessoais

Ana Paula Chiarelli de Souza

Baseado no projeto anterior, foram escolhidas as entidades Grupo, Usuário e o relacionamento Integrante para desenvolvimento do projeto em sua fase final.

Notação simplificada

Usuário(#cod_usuario, email, senha, nome)

Grupo(#cod_grupo, nome, &cod_usuario)

Integrante(#&cod_grupo, #&cod_usuario)

Criação do banco de dados no PostgreSQL

```
CREATE TABLE usuarios (
	cod_usuario SERIAL PRIMARY KEY NOT NULL,
	email VARCHAR(100) UNIQUE NOT NULL,
	senha VARCHAR(100) NOT NULL,
	nome VARCHAR(100) NOT NULL
);

CREATE TABLE grupos (
	cod_grupo SERIAL PRIMARY KEY NOT NULL,
	nome VARCHAR(100) NOT NULL,
	proprietario int NOT NULL,
	FOREIGN KEY (proprietario) REFERENCES usuarios (cod_usuario)
);

CREATE TABLE integrante (
	cod_usuario int NOT NULL,
	cod_grupo int NOT NULL,
	PRIMARY KEY (cod_usuario, cod_grupo),
	FOREIGN KEY (cod_usuario) REFERENCES usuarios (cod_usuario),
	FOREIGN KEY (cod_grupo) REFERENCES grupos (cod_grupo)
);
```

Escopo do projeto:

Uma opção para inserir novas tuplas da primeira tabela
"INSERT INTO usuarios (email, senha, nome) VALUES (?,?,?)"

Uma opção para inserir novas tuplas da segunda tabela (grupos)
"INSERT INTO grupos (nome, proprietario) VALUES (?,?) returning cod_grupo"

Returning foi utilizado para recuperar o valor do código do grupo (por ser serial) e setá-lo como o último grupo inserido, porque o usuário proprietário precisa ser inserido automaticamente como integrante do grupo.

Uma opção para listar todas as tuplas da primeira tabela
"SELECT cod_usuario, email, senha, nome FROM usuarios"

Uma opção para listar todas as tuplas da segunda tabela
"SELECT cod_grupo, nome, proprietario FROM grupos"

Uma opção para listar o resultado de uma consulta que envolva uma junção entre as duas tabelas
"select cod_usuario, email, senha, usuarios.nome, grupos.nome, proprietario, cod_grupo from grupos join usuarios on grupos.proprietario = usuarios.cod_usuario"

A função retorna todos os grupos e o nome de seus respectivos proprietários.

Uma opção para listar o resultado de uma consulta que envolva subconsulta(s) e uma ou mais funções de agregação.
O usuário irá informar o grupo e a consulta retornará a quantidade de grupos que o usuário proprietário é integrante. Para isso precisei criar a função que insere usuários em grupos (sem ser o administrador, que é adicionado automaticamente durante a criação de um grupo).

"insert into integrante (cod_usuario, cod_grupo) values (?, ?)"

Com essa função eu pude inserir usuários nos grupos e utilizar a consulta abaixo para verificar a quantidade de grupos que o proprietário participa:

select count(*) from integrante where cod_usuario = (select proprietario from grupos where cod_grupo = ?)

Na aplicação, precisei quebrar esta subconsulta em duas, para verificar se o grupo que o usuário informou existe e utilizar o resultado da subconsulta como input da consulta mais externa para contar o número de grupos daquele usuário.