

Universidade da Beira Interior

Departamento de Informática

Tecnologias de Base de Dados



Professor: Rui Cardoso
Engenharia Informática

João Brito nº M9984
Luís Pereira nº M10156
Carlos Esteves nº E10304

1. Introdução

Este projeto foi realizado no âmbito da unidade curricular de Tecnologias de Base de Dados com o objetivo de criar uma base de dados preparada para operar com acessos concorrentes e que satisfaça várias operações relevantes a cada tipo de utilizador, também é objetivo a criação de uma aplicação de modo a testar a gestão das transações.

Ao longo deste trabalho foram desenvolvidas várias aplicações que permitem comunicar com a base de dados através de uma arquitetura cliente-servidor.

Este trabalho coloca em prática os conteúdos lecionados tanto nas aulas práticas assim como nas aulas teóricas, aprofundando deste modo os conhecimentos adquiridos.

2. Modelo de dados

Os dados utilizados foram obtidos do site “<https://github.com/centraldedados/sinistralidade.git>” e foram previamente processados com scripts de modo a facilitar o preenchimento da base de dados.

O modelo de dados utilizado consiste em quatro tabelas: Distritos, Concelhos, Vias, Acidentes. Cada uma destas tem os seus atributos como se pode observar na seguinte figura:

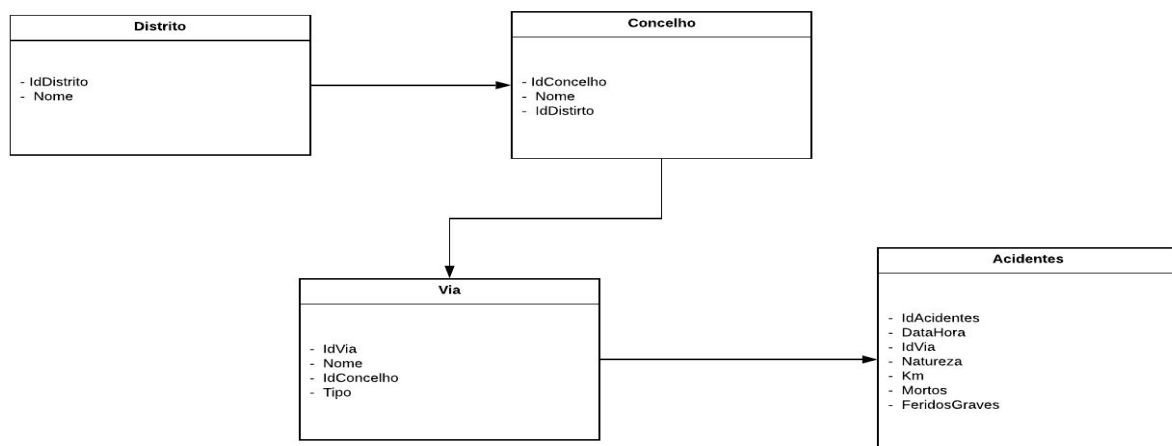


Figura 1 - Modelo de dados.

A tabela “Distrito” possui IdDistrito como chave primária e ainda Nome. Esta relaciona-se com a tabela “Concelhos” através do IdDistrito. Na tabela “Concelhos” temos os campo de IdConcelho como chave primária, o Nome e ainda IdDistrito (chave estrangeira). A tabela “Concelhos” relaciona-se com a das “Vias” através do IdConcelho, sendo chave estrangeira da tabela das vias. Para além deste atributo, também apresenta IdVia como chave primária, Nome e Tipo. Por último a tabela Acidentes possui IdAcidentes (chave primária), DataHora, IdVia (chave estrangeira), Natureza, Km, Mortos, FeridosGraves.

Este foi o modelo de dados utilizado pelo nosso grupo e encontra-se preenchido com dados de 2013 a 2017.

3. SGBD e Base de Dados

Para criar e carregar os dados para a base de dados foi desenvolvido o script “create_db.sql” que garante que a base de dados não existe já no sistema e depois a cria e carrega dados de 2013 a 2017 de acordo com o nosso modelo de dados para as respetivas tabelas. Este também cria e garante as permissões aos diferentes utilizadores.

A base de dados opera segundo o seguinte esquema relacional:

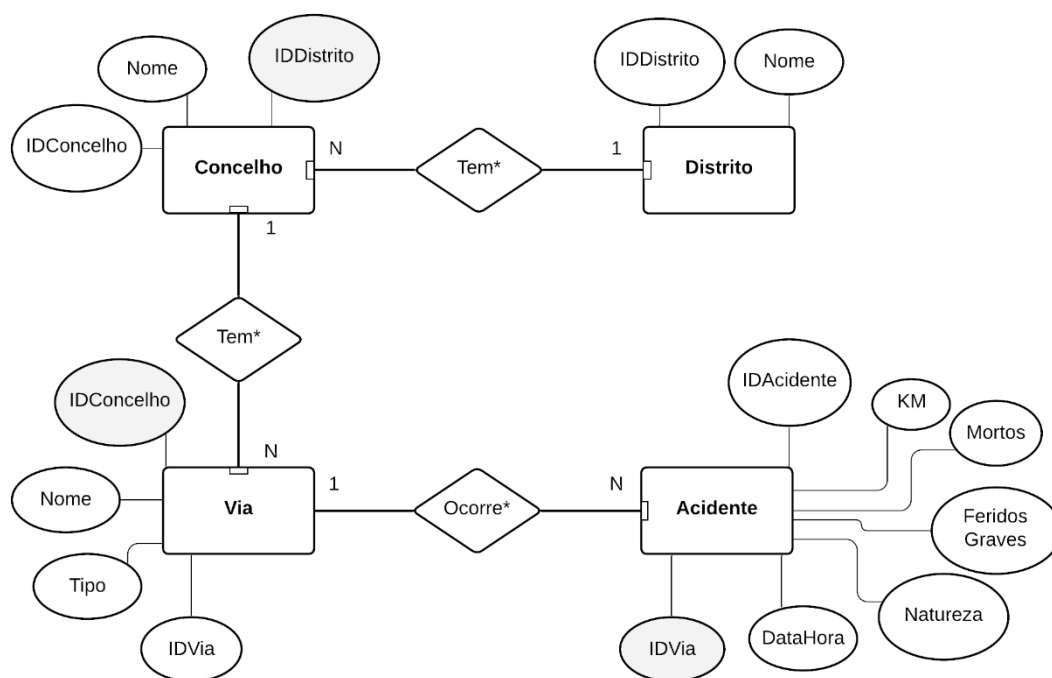


Figura 2 - Esquema relacional.

Os acidentes estão relacionados com as vias de modo a que numa via ocorrem vários acidentes, mas um acidente ocorre apenas numa via. Da mesma forma, uma via apenas tem um concelho, no entanto um concelho tem várias vias. Por último um concelho apenas tem um distrito, mas um distrito tem vários concelhos.

De modo a lidar com a concorrência e gerir as interrogações à base de dados foram implementados o “server.py” e “cliente.py”. O servidor opera através de sockets e pode ser dividido em três partes, na primeira este estabelece a conexão com o cliente, de seguida adiciona o pedido do cliente a uma fila assim como os seus dados (utilizador e password) e por fim faz o pedido à base de dados, retornando o resultado ao cliente. No fim deste processo, servidor e cliente desconectam-se.

Como o servidor opera com recurso a uma fila, garante que nenhum pedido em simultâneo é executado na base de dados, evitando assim que os mesmos dados estejam, por exemplo, a ser editados por clientes diferentes.

4. Aplicação

A aplicação foi desenvolvida em Python utilizando a biblioteca Kivy, que permite um rápido desenvolvimento de aplicações e que garante interfaces inovadoras. Foram então desenvolvidas interfaces para os três tipos de utilizadores: polícia, hospital e utilizadores apenas de consulta (i.e. outras entidades).

Ao utilizador polícia é-lhe permitido criar e inserir na base de dados um novo acidente, ao hospital é permitido pesquisar acidentes e editar os campos de mortos e feridos graves, o terceiro tipo de utilizador apenas consegue consultar dados, mas não tem permissões para os editar.

Todos os requests por parte destes utilizadores à base de dados iniciam um cliente que faz o pedido ao servidor. Assim, o servidor serve como “middleware” e garante que não há dados a ser editados ao mesmo tempo.

5. Testes

Ao longo desta secção iremos apresentar alguns dos possíveis teste para comprovar a funcionalidade do sistema. Uma das formas de garantir que o acesso a base de dados é feito de forma isolada, ligamos o servidor e, de seguida, preparamos os clientes e fazemos diversos pedidos à base de dados.

O que comprovamos com este teste é que os vários pedidos realizados à base de dados não entram em conflito devido à hierarquia do sistema. A concorrência também é testada e verifica-se que não há qualquer problema em acessos simultâneos.

O próximo teste tem como alvo a consistência da base de dados. Para tal, iremos tentar eliminar algum campo da tabela “Distritos”, o que iria provocar com que todos os dados de outras tabelas com referência a este distrito ficassem inconsistentes. Este teste não se consegue realizar pois nenhum utilizador tem permissões para eliminar dados.

Um outro teste seria adulterar os dados da base de dados. De forma a tornar a base de dados novamente íntegra, é possível aceder aos “logs” e reverter as alterações a esta por ordem cronológica.

6. Conclusão

Em suma foi conseguido criar e importar dados csv para uma base de dados, também foi desenvolvida uma interface para os vários tipos de utilizadores e um servidor-cliente que gere as transações das interfaces com a base de dados. É então possível consultar dados sobre acidentes, alterar dados e listar dados de acidentes no concelho.

Relativamente às propriedades básicas da base de dados , estas foram cumpridas. O trabalho futuro consiste em otimizar os programas implementados, encontrar possíveis falhas e acrescentar mais funcionalidades de estatísticas relativamente aos dados da base de dados.