

Tecnologias de Base de Dados**1º Trabalho Prático**

Versão 1.0 (2-Out-2019)

Tema: Controlo de Transações**1. Introdução**

No desenvolvimento de sistemas de informação é frequente lidar com cenários onde várias aplicações acedem, de modo concorrente a dados partilhados. O Sistema de Gestão de Bases de Dados (SGBD) é a entidade responsável pelo controlo e gestão destes acessos concorrentes. Para desenvolver competências nesta área foi considerada a seguinte situação:

A Autoridade Nacional de Segurança Rodoviária (ANSR) é um serviço central da administração direta do Estado que tem por missão o planeamento e coordenação da segurança rodoviária em Portugal. Para esse fim disponibiliza relatórios com dados agregados sobre acidentes e também listagens dos acidentes com mortos e ou feridos graves que ocorreram em Portugal Continental ao longo dos anos. Esta informação é disponibilizada no site da ANSR em ficheiros no formato pdf (<http://www.ansr.pt/Estatisticas/RelatoriosDeSinistralidade/Pages/default.aspx>).

Parte desta informação encontra-se também disponível no formato csv na Central de Dados (<http://centraldedados.pt/>), um repositório aberto com múltiplos datasets de informação pública. Os datasets em csv com os dados brutos relativos aos acidentes com mortos e ou feridos graves podem ser descarregados do projeto da central de dados sobre sinistralidade no GitHub (<https://github.com/centraldedados/sinistralidade.git>).

Com base na informação da ANSR pretende-se criar um sistema de informação que permita gerir em ‘tempo real’ esta informação através de uma base de dados (servidor) e uma aplicação (cliente). Toda a informação deve estar disponível no sistema de modo que permita responder às questões principais referidas nos relatórios da ANSR. Por exemplo, onde estão os pontos negros no ano 2018. Mas fundamentalmente pretende-se que a base de dados possa gerir múltiplos acessos de leitura e escrita sem comprometer a integridade e fiabilidade da informação armazenada. Para esse fim devem apresentar testes com acessos concorrentes que garantam a consistência e integridade da informação para todos os utilizadores em simultâneo da base de dados. Na Fig. 1 mostra-se um esquema com vários tipos de utilizadores que podem aceder à base de dados, com níveis de acesso diferenciados e usando credenciais de acesso (user/password).

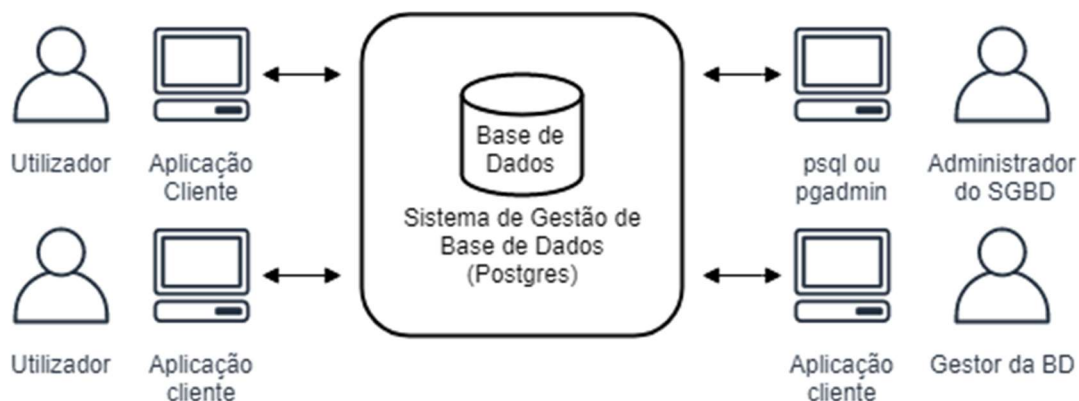


Figura 1 Exemplo de diferentes tipos de utilizadores/clientes que podem ter acessos concorrentes à base de dados implementada no SGBD.

Na Fig. 2 são apresentados diferentes tipos de operações realizadas de diferentes localizações e por vários tipos de utilizadores. Como estas operações podem gerar acessos concorrentes, na implementação pretende-se que a base de dados consiga manter a consistência da informação.

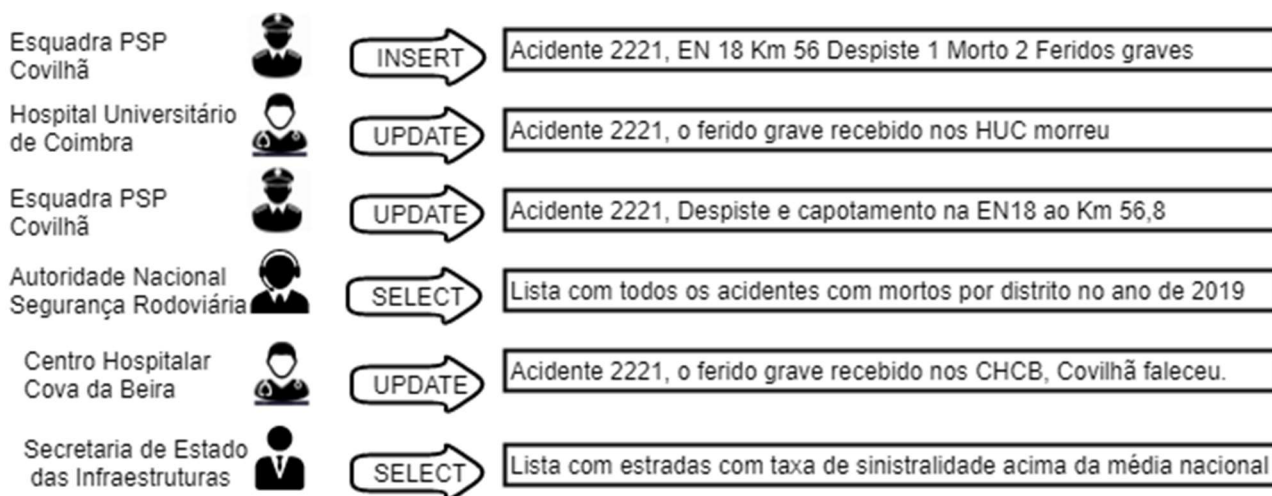


Figura 2 Exemplo de situação com múltiplos acessos a um registo para ilustrar eventuais situações de concorrência.

Na Fig. 3 é representada uma situação em que duas aplicações cliente pretendem alterar/atualizar/eliminar um registo. Por exemplo, se as duas operações de update do Hospital Universitário de Coimbra e da Esquadra da PSP da Covilhã descritas na Fig. 2 forem em simultâneo, como é gerida a situação?

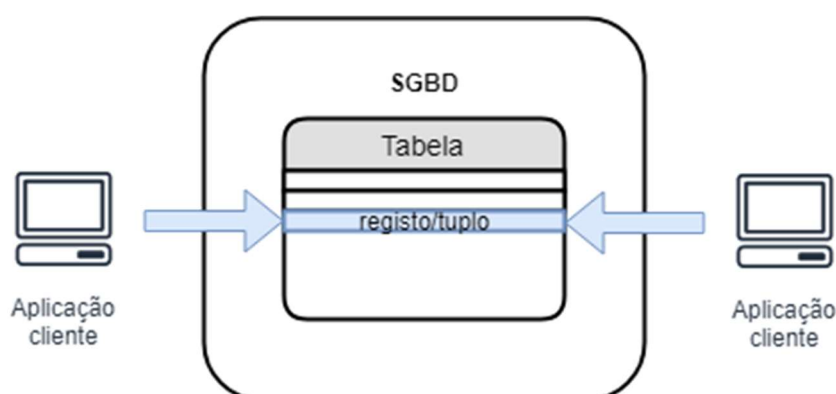


Figura 3 Esquema que mostra o acesso concorrente a uma tabela da base de dados.

2. Objetivos do trabalho

1. Criar uma base de dados que inclua a informação disponibilizada pela ANSR, preparada para operar com acessos concorrentes e que satisfaça as operações mais relevantes para os diferentes tipos de utilizadores.
2. Aprender a efetuar o controlo de transações no SGBD/BD/Aplicação:
 - a. Compreender o que são transações (<https://www.postgresql.org/docs/12/tutorial-transactions.html>).
 - b. Nível de isolamento (<https://www.postgresql.org/docs/current/transaction-iso.html>).
 - c. Tipos de trincos (<https://www.postgresql.org/docs/current/explicit-locking.html>).
 - d. WAL escrita à cabeça (<https://www.postgresql.org/docs/12/wal-intro.html>).
3. Desenvolver uma aplicação para testar a gestão das transações.
4. Apresentar a solução em ambiente operacional de teste.
5. Escrever um relatório sobre o trabalho realizado.

3. Competências a adquirir

1. Desenvolver a capacidade de análise e de implementação de bases de dados.
2. Desenvolver as competências necessárias para gerir SGBD.

3. Resolver situações de concorrência em base de dados.
4. Desenvolver aplicações cliente/servidor ligadas a SGBD.

4. Datas Importantes

Data de divulgação: 25 de setembro de 2019 (2ª aula teórica), 9 de outubro de 2019 (enunciado disponibilizado).

Formação dos grupos: Os grupos têm de estar formados e enviados ao docente até 12 de outubro de 2019.

Data de entrega: Dom, 3 de novembro de 2019 (cada dia de atraso implica menos um valor na nota).

Data da defesa: 5ªf, 7 de novembro de 2019 ou 6ªf 8 de novembro de 2019 (a efetuar no turno prático ou teórico).

5. Grupos

O trabalho é realizado por grupos de 3 alunos do mesmo turno prático, caso não seja possível e excecionalmente podem ser considerados elementos de outro turno.

6. Avaliação

O trabalho está cotado para 5 valores.

Os componentes avaliados são: o modelo de dados implementado, a importação dos dados em csv, a aplicação desenvolvida, as operações disponíveis de acesso à base de dados, os mecanismos desenvolvidos para gerir as transações, o relatório e a apresentação.

O trabalho será defendido pelo grupo na 8ª semana de aulas no turno prático respetivo ou no turno teórico (sujeito a confirmação quando for entregue o trabalho). Alunos que não participem na apresentação do trabalho terão zero valores. Cada aluno pode ter uma classificação diferente da dos colegas de grupo, refletindo deste modo o seu desempenho no trabalho e na discussão do mesmo. Todos os trabalhos serão demonstrados e defendidos perante o docente da disciplina em sessões de defesa do trabalho específicas para cada grupo. As defesas dos trabalhos têm duração aproximada de 15 a 20 minutos. Na apresentação será fornecido pelo docente o trabalho recebido de cada grupo. Ficará a cargo dos alunos a sua instalação de modo a terem o trabalho pronto a demonstrar na sala e na hora marcada para a defesa. Para evitar eventuais percalços os grupos devem usar pelo menos um computador da sala com a base de dados implementada no SGBD Postgres e os outros dois podem ser outros computadores na sala ou portáteis dos alunos, com a aplicação desenvolvida instalada para testar a gestão da concorrência. Se usarem os portáteis é necessário que forneçam o MAC address de cada um atempadamente ao gestor de sistemas para que seja possível comunicar na rede da sala no dia da defesa.

7. Desenvolvimento

Software

O sistema de gestão de base de dados a usar para a realização do trabalho é o Postgres. A aplicação a desenvolver pode utilizar uma linguagem de desenvolvimento à escolha do grupo.

SGBD

- Postgres 12, Manual em A4 (<https://www.postgresql.org/files/documentation/pdf/12/postgresql-12-A4.pdf>).
- Postgres, documentação oficial do SGBD com exemplos (<https://www.postgres.com>).
- Postgres, instalação Win,Mac, Linux (<https://www.postgresql.org/download/>)
- pgAdmin, aplicação para gerir o SGBD Postgres (<https://www.pgadmin.org/download/>).

A base de dados deve ser implementada através de um script SQL.

Deve criar um backup da base de dados com o modelo de dados e os dados já inseridos, que permita depois reconstruir a base de dados em qualquer máquina com o Postgres.

APLICAÇÃO

Relativamente à aplicação cliente a desenvolver podem usar a ferramenta que considerarem mais adequada. Em baixo está uma lista com várias alternativas de ferramentas e exemplos de aplicações clientes desenvolvidas em Java, FreePascal, C#, Python e Radzen para SQL Server e Postgres (no trabalho só se pode utilizar o Postgres como SGBD).

- Exemplo aplicação JBuilder/SQL Server http://www.di.ubi.pt/~pprata/bd/BD0405-SQLServer_JBuilder.pdf
- Exemplo aplicação JBuilder/SQL Server <http://www.di.ubi.pt/~pprata/bd/Projecto.rar>
- Exemplo aplicação C#/Postgres (<https://www.codeproject.com/Articles/30989/Using-PostgreSQL-in-your-C-NET-application-An-intr>).
- Exemplo aplicação Lazarus/SQL Server (<http://webx.ubi.pt/~rcardoso/tbd1920/software/Ex2.zip>)
- Biblioteca para aplicação Lazarus/Postgres <https://www.devart.com/pgdac/pgdacfpc.zip>
- Instalar o Lazarus Win (<https://www.lazarus-ide.org/index.php?page=downloads>).
- Instalar o Radzen Win,Mac,Linux (<https://download.radzen.com/download/>)
- Instalar Radzen/Postgres (Rapid Application Development for the Web) (<https://www.radzen.com>)
- Ligação App/Web Radzen/Postgres (<https://www.radzen.com/documentation/postgresql/>).
- Instalar Radzen/Postgres Win,Mac,Linux (<https://download.radzen.com/>).
- Exemplo aplicação Python/Postgres (<http://webx.ubi.pt/~rcardoso/tbd1920/software/query2db.tar>)

A aplicação deve permitir efetuar consultas, inserções e remoções de informação para vários tipos de utilizadores. Devem ser também desenvolvidos testes para os diversos níveis de isolamento do Postgres e todas as transações devem ser registadas na base de dados.

8. Tarefas a efetuar

Elaborar o diagrama entidade relacionamento e o correspondente esquema relacional. Tenha presente que o modelo deve estar normalizado, assumindo para as situações não especificadas as soluções que pareçam mais plausíveis. No desenvolvimento deve também indicar explicitamente as escolhas efetuadas.

Produzir scripts em SQL para: criar a base de dados, efetuar o upload dos ficheiros csv e efetuar backup do esquema e dos dados que permita reconstruir a base de dados em qualquer SGBD Postgres.

No desenvolvimento da base de dados implementar procedimentos armazenados e funções definidas pelo utilizador (UDFs User Defined Functions) (<https://www.postgresql.org/docs/current/xfunc.html>), criando os gatilhos (triggers) (<https://www.postgresql.org/docs/12/triggers.html>) necessários para salvaguardar a operacionalização da base de dados.

Construir uma aplicação (na linguagem escolhida) que permita efetuar as operações necessárias aos vários tipos de utilizadores e que possa demonstrar como é efetuada a gestão da concorrência.

Elaborar um relatório descrevendo o trabalho realizado.

9. Documentação a entregar

Código em SQL para reconstruir a base de dados com dados no SGBD, o código da aplicação desenvolvida e o relatório.

Submeter através da página da disciplina no Moodle três ficheiros **x_y_z_bd.zip** com a base de dados, **x_y_z_app.zip** com a aplicação e **x_y_z_rel.zip** com o relatório (substituir x, y e z pelos números dos elementos do grupo e submeter dentro do prazo estabelecido).

10. Relatório

Deve ser elaborado um relatório detalhado abordando, pelo menos, os seguintes tópicos:

1. Introdução - Apresentação do trabalho desenvolvido.
2. Modelo de dados - Apresentação e justificação do modelo de dados desenvolvido, incluindo as opções tomadas para as situações não especificadas.

3. SGBD e Base de Dados – Apresentar o modelo relacional, as funcionalidades implementadas para gerir as interrogações à base de dados e a gestão da concorrência com exemplos.
4. Aplicação – Apresentar a aplicação e as funcionalidades implementadas.
5. Testes – Apresentar os testes que efetuou que validem a gestão da concorrência. Caracterizar as transacções (gestão, controlo, níveis de isolamento e recuperação), os tipo de trincos (*locks*) e descrever também os resultados obtidos.
6. Conclusão - Indicar o que foi conseguido. Indicar o que não foi conseguido. Indicar a(s) razão(ões).

11. Referências

SQL Server 2000 Para Profissionais, Orlando Belo, FCA ISBN 972-722-505-5

SQL - Structured Query Language, Luís Manuel Dias Damas, FCA ISBN 972-722-443-1

Gestão de transações Chapter 22 Transaction Management pp 667 – 724 Database Systems: A Practical Approach to Design, Implementation, and Management 6th edition, Thomas Connolly and Carolyn Begg.

Postgres SQL Prático (https://pt.wikibooks.org/wiki/Categoria:Livro/PostgreSQL_Pr%C3%A1tico).

Postgres SQL Prático Transações (https://pt.wikibooks.org/wiki/PostgreSQL_Pr%C3%A1tico/Transa%C3%A7%C3%B5es).

Postgres Transações (<https://www.postgresql.org/docs/12/tutorial-transactions.html>).

Postgres Nível de isolamento (<https://www.postgresql.org/docs/current/transaction-iso.html>).

Postgres Tipos de trincos (<https://www.postgresql.org/docs/current/explicit-locking.html>).

Postgres WAL escrita à cabeça (<https://www.postgresql.org/docs/12/wal-intro.html>).