Introdução às Bases de Dados

Tópicos de Gestão de Bases de Dados

FCUL, Departamento de Informática Ano Letivo 2021/2022

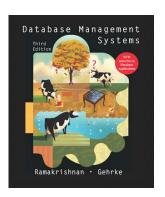
Ana Paula Afonso

Sumário e Referências

Sumário

- Bases de dados e SGBDs
 - Razões de utilização, vantagens e desvantagens
- Componentes de um SGBD e Utilização de SGBD
 - Conceção de BD e desenvolvimento de aplicações
 - Análise de dados
 - Concorrência e robustez
 - Eficiência e escalabilidade

- Referências
 - R. Ramakrishnan (capítulo 1)



Razões de Uso de SGBDs

- Suportam conceitos úteis para a gestão de BDs
 - Linguagem para criar, alterar, e consultar/analisar dados
 - Restrições para manter a integridade dos dados (ex. idade > 0)
 - Utilizadores e privilégios de acesso aos dados
 - Transações: programas que transformam o estado da BD
- Oferecem **mecanismos automáticos** de gestão de BDs
 - Aplicação de restrições de integridade aquando de alterações nos dados
 - Controlo de acesso à BD por vários utilizadores em simultâneo
 - Recuperação de faltas (ex. falta de energia elétrica)
 - Otimização do acesso aos dados, para respostas rápidas
- Suportam grandes quantidades de dados e de transações
 - Mecanismos desenhados para permitir expansão de capacidade

Vantagens dos SGBD (1)

Independência dos dados

- Aplicações não estão expostas aos detalhes de como os dados estão representados e armazenados
- SGBD disponibiliza uma visão abstrata dos dados

Acesso eficiente aos dados

 O SGBD utiliza uma variedade de técnicas sofisticadas para armazenar e recolher dados de uma forma eficiente

Vantagens dos SGBD (2)

Integridade dos dados e segurança

 O SGBD pode aplicar restrições de integridade durante o acesso aos dados

Administração dos dados

Profissionais experientes podem organizar
a representação dos dados,
por forma a minimizar a redundância
e melhorar o armazenamento e recolha dos dados

Vantagens dos SGBD (3)

Acesso concorrente e recuperação de falhas

- Acesso aos dados como se fosse acedido por um utilizador de cada vez
- Minimiza os efeitos de falhas no sistema

Redução do tempo de desenvolvimento de aplicações

- Disponibiliza funções de acesso comuns
- Interface de alto nível para os dados
- Mais robusto:

Tarefas executadas pelo SGBD não precisam de ser verificadas

SGBD Desvantagens

- Aplicações complexas de software
- Desempenho inaceitável para algumas aplicações
 - Aplicações de tempo-real
- Não disponibiliza análise flexível de dados texto
- Nem sempre os benefícios dos SGBD são necessários

Quando **Não** usar um SGBD

- Custos iniciais demasiado elevados
 - Investimentos em hardware, software e formação
 - Necessidade de definição e aplicação de políticas de segurança
- Bases de dados simples ou imutáveis (*)
- Ausência de acessos concorrentes (*)
- Requisitos de tempo-real incontornáveis
 - SGBDs não garantem execução num intervalo de tempo fixo
- Custo inicial do software pode ser reduzido
 - Open source: MySQL, MariaDB, PostgreSQL, ...
 - Versões grátis de produtos: Oracle Express, SQL Server Express, ...

(*) uso de ficheiros - pode ser suficiente

Bases de Dados noSQL

- NoSQL é uma abordagem recente para criar bases de dados
 - o termo NoSQL é usado para referir bases de dados não relacionais
 - são BD não tabulares e armazenam os dados de maneira diferente das tabelas relacionais
 - as BD NoSQL surgem em vários tipos com base no seu modelo de dados
 p. e.: documento, chave-valor (key-value), graph,...
 - desenhadas para serem escaláveis para grandes quantidades de dados grandes quantidades de utilizadores
- Existem vários produtos NoSQL
 - exs: MongoDB, Neo4j, Apache Cassandra, Google Cloud BigTable

Ranking dos SGBD

Dec 2021	Rank Nov 2021	Dec 2020	DBMS	Database Model
1.	1.	1.	Oracle 🖽	Relational, Multi-model 👔
2.	2.	2.	MySQL 🚹	Relational, Multi-model 👔
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model 👔
4.	4.	4.	PostgreSQL □ ●	Relational, Multi-model 👔
5.	5.	5.	MongoDB 🔠	Document, Multi-model 🔞
6.	6.	1 7.	Redis 🚹	Key-value, Multi-model 👔
7.	7.	4 6.	IBM Db2	Relational, Multi-model 👔
8.	8.	8.	Elasticsearch	Search engine, Multi-model 👔
9.	9.	9.	SQLite #	Relational
10.	1 11.	1 11.	Microsoft Access	Relational

https://db-engines.com/en/ranking

Utilização de SGBDs - questões essenciais

- Conceção de Bases de Dados e Desenvolvimento de Aplicações
- 2. Análise de Dados
- 3. Concorrência e Robustez
- 4. Eficiência e Escalabilidade

Concepção de Bases de Dados e Análise dos Dados

Conceção de BDs

- Como se pode descrever um caso real
 - (ex. uma universidade) em termos de dados a guardar num SGBD?
- Que fatores devem ser considerados na decisão de como organizar os dados armazenados?
- Como desenvolver aplicações que recorrem a um SGBD?

Análise dos Dados

 Como o utilizador pode obter resposta às suas questões através de interrogações aos dados no SGBD?

Concorrência e Robustez / Eficiência e Escalabilidade

Concorrência e Robustez

- Como é que um SGBD permite o acesso concorrente aos dados?
- Como é que este protege os dados num caso de uma falha do sistema? (ex. falta eletricidade)

Eficiência e Escalabilidade

- Como é que um SGBD armazena grandes colecções de dados e executa interrogações nesses dados de uma forma eficiente? (ex. índices)
- Como lida com grande quantidade de utilizadores e acessos?

SQL em Aplicações

- Os comandos SQL podem ser chamados através de uma aplicação informática
 - Escrita numa qualquer linguagem host
 - Java, C, PHP, Perl, Python, ...

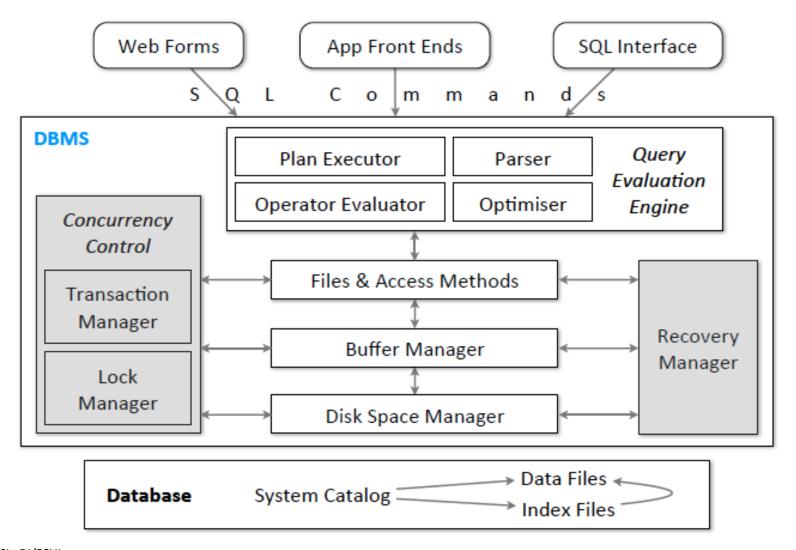
Um exemplo: PHP

```
<?php
           // Estabelece uma ligação com a base de dados usando o programa openconn.php
           // A variável $conn é inicializada com a ligação estabelecida
           include "openconn.php";
           $query = "INSERT INTO pessoa VALUES ('O seu nome', 50)";
           $res = mysqli query($conn, $query);
           if ($res) {
                       echo "Um novo registo inserido com sucesso";
           } else {
                      echo "Erro: insert failed" . $query . "<br>" . mysqli error($conn);
           // Termina a ligação com a base de dados
           mysqli_close($conn);
?>
```

Em exemplo: PHP

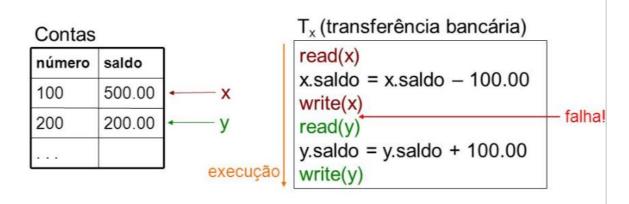
```
<?php
          $dbhost = "appserver-01.alunos.di.fc.ul.pt";
          $dbuser = "ibd000";
          $dbpass = "XXXXXXX";
          $dbname = "ibd000";
          $conn = mysqli_connect($dbhost, $dbuser, $dbpass, $dbname);
          if (mysqli_connect_error()) {
                     die("Database connection failed:".mysqli_connect_error());
?>
```

Componentes de um SGBD



Gestão de Transacções

- Exemplos:
 - Transferência bancária de dinheiro com falha do sistema
 - Compra de produtos com acesso concorrente
- Uma transação é uma qualquer execução de uma tarefa unitária no SGBD



Gestão de Transações - Motivação

Produto

<u>pid</u>	pnome	preço	stock
1	P1	20	25
2	P2	12	10
3	Р3	15	10
4	P4	20	10

Cenário

9:00:00 - uma aplicação cliente (C1) efetua uma pesquisa sobre a quantidade de produtos "P1" em stock

SELECT stock FROM Produto WHERE pnome='P1'

9:00:01 - outra aplicação (C2) efetua uma pesquisa semelhante

O SGBD responde a ambos "25"

C1 reserva 20 unidades, pelo que efetua uma dedução da respetiva quantidade à BD

C2 efetua uma operação semelhante

Transação - Definição

- Uma transação num SGBD é uma abstração de um procedimento
 - Uma sequência de operações de leitura e/ou escrita SQL ...
 - executada de forma atómica pelo SGBD

```
Início da transação
    Instrução 1
    Instrução 2
    ...
    Instrução N
```

Fim da transação -- OK (**COMMIT**) ou Erro (**ROLLBACK**)

- O mecanismo de transações é essencial
 - sempre que a BD servir vários utilizadores simultaneamente
 - recuperação de falhas

Concorrência e Consistência

- Requisitos de um sistema transacional
 - Gestão de múltiplas transações em simultâneo
 - Gestão das regras de integridade
- Concorrência
 - Múltiplos utilizadores e respetivos pedidos em simultâneo
- Consistência (Integridade)
 - A BD está num estado consistente quando cumpre ...
 - ... as regras de integridade
- Um SGBD é um sistema transacional

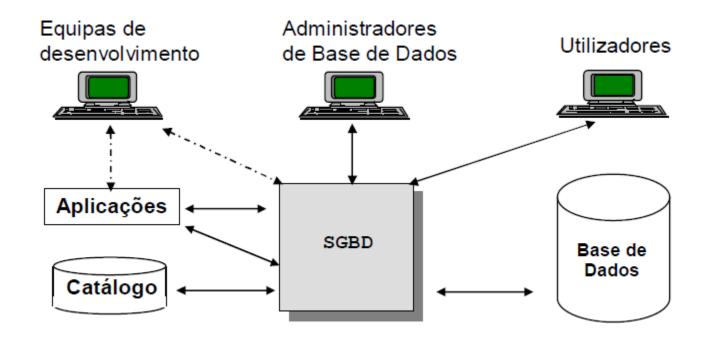
Execução Concorrente de Transacções

- Um protocolo de locking é um conjunto de regras que devem ser seguidas por cada transacção (garantido pelo SGBD)
 - Apesar das operações das transacções serem executadas intercaladamente
 - O resultado é o mesmo de executar as transacções numa determinada ordem
- Dois tipos de *locks*:
 - Partilhados por múltiplas transacções (leitura)
 - Exclusivos a uma transacção (escrita e leitura)

Transacções Incompletas

- Alterações feitas por transacções incompletas devem ser anuladas
- O SGBD mantém um log de todas as operações de escrita:
 - Para anular transacções incompletas
 - Ou refazer transacções completas depois de uma falha
 Checkpoints (forçam escrita em disco) para minimizar o tempo de recuperação (dim do log)

Desenvolvimento e Uso de SGBDs



Catálogo: contém a definição da BD. Ex: num SGBD relacional, contém definições das tabelas, vistas, regras de integridade, etc.

Intervenientes

- Empresas que desenvolvem os SGBDs (DB implementors)
- Programadores (DB application programmers)
 - Criam pacotes de software que permitem aos utilizadores aceder aos dados
- Utilizadores de aplicações (end users)
 - Indústrias, serviços, fora ou dentro da Web
 - Fazem entrar dados na organização e analisam dados
- Administradores de bases de dados (DB Administrator DBAs)
 - Manipulam esquemas físico e lógico (ex. para afinação)
 - Criam utilizadores e definem privilégios de acesso aos dados
 - Garantem disponibilidade e recuperação em caso de faltas

Tarefas Críticas do Administrador BD

- Concepção do modelo lógico e físico
 - Interage com os utilizadores para decidir que relações (tabelas) armazenar e como
- Fiabilidade
 - Faz backups periódicos
 - e mantém logs da actividade do sistema
- Afinação
 - Adequa o desempenho às alterações de requisitos

Tarefas Críticas do Administrador BD

Segurança e autorização

- Atribui diferentes permissões de acesso
- 4 tipos de acesso aos recursos (table/view): CRUD
 Create, Read, Update, Delete
- Utilização diferenciada para diferentes roles
 Um utilizador pode ter vários roles
 p.e., app_programador, app_read, app_write
 Cada role está associado a privilégios

```
CREATE ROLE 'app_programador', 'app_read', 'app_write'
GRANT ALL ON appdb.* TO 'app_programador';
GRANT SELECT ON appdb.* TO 'app_read';
GRANT INSERT, UPDATE, DELETE ON appdb.* TO 'app_write';
```