

BASE DE DADOS

ANTÓNIO GONÇALVES

# ÍNDICE

Introdução as Base de Dados .....	1
-----------------------------------	---

# INTRODUÇÃO AS BASE DE DADOS

## Parte 1: Conceitos Fundamentais

1. Quais são duas desvantagens importantes associadas ao uso de sistemas de bases de dados em comparação com sistemas de ficheiros tradicionais?
2. Quais são cinco diferenças principais entre a linguagem de definição de dados (DDL) usada em bases de dados e a declaração de tipos em linguagens de programação como Java ou C++?

## Parte 2: Configuração e Planeamento

1. Quais são os seis passos essenciais para configurar um sistema de bases de dados desde o início?
2. Quais problemas poderiam surgir ao armazenar metadados e dados principais exclusivamente em ficheiros simples?

## Parte 3: Consultas e Integração

1. Quais são as principais diferenças entre consultas feitas em motores de busca e consultas realizadas em bases de dados relacionais?
2. Desenvolva um esquema para um sistema de gestão de biblioteca, identificando tabelas para livros, membros e empréstimos, e explique como as tabelas devem estar relacionadas.

## Parte 4: Base de Dados - Aplicações e Modelos

1. O que é um sistema de gestão de bases de dados (SGBD) e quais são as suas principais funções?
2. Cite pelo menos três exemplos de aplicações reais que utilizam sistemas de bases de dados e explique brevemente a sua utilidade.
3. Qual é o principal objetivo dos sistemas de bases de dados em relação à redundância de dados? Explique como eles ajudam a resolver problemas de redundância e inconsistência.
4. Descreva o conceito de atomicidade das transações e dê um exemplo de como este princípio é aplicado num sistema de bases de dados.
5. O que é o modelo relacional de bases de dados e como ele organiza os dados?
6. Explique os três níveis de abstração de dados (físico, lógico e de visualização) e a importância de cada um.
7. Diferencie esquemas físicos, esquemas lógicos e instâncias numa base de dados.

8. O que é a linguagem de definição de dados (DDL)? Dê um exemplo de como criar uma tabela usando DDL.
9. O que é a independência de dados físicos e como ela beneficia o uso de bases de dados?
10. Quais foram os avanços mais significativos na história dos sistemas de bases de dados desde os anos 1950 até os anos 2010?

## Respostas

**Parte 1: Conceitos Fundamentais** 1. Custos iniciais elevados devido à necessidade de software, hardware e formação. Complexidade de implementação e manutenção, exigindo conhecimentos especializados.

1. A DDL cria estruturas reais na base de dados; Java/C criam abstrações para o programa. A DDL permite restrições de integridade, como chaves primárias, enquanto Java/C não. A DDL suporta autorizações para controlo de acesso, algo ausente em Java/C. Java/C têm um sistema de tipos mais rico; a DDL é mais limitada. A DDL foca-se em tabelas e relações; Java/C++ trabalham com objetos.

**Parte 2: Configuração e Planeamento** 3. Identificar os requisitos do sistema. Criar um modelo de dados com tabelas e relações. Definir restrições de integridade para garantir consistência. Planejar a estrutura física de armazenamento. Desenvolver aplicações para interação dos utilizadores. Inicializar o sistema com dados.

1. Redundância e inconsistência de dados. Dificuldade de acesso e pesquisa eficiente. Isolamento de dados, dificultando a integração. Limitações de segurança e controlo de acesso.

**Parte 3: Consultas e Integração** 5. Motores de busca aceitam palavras-chave simples; bases de dados usam linguagem estruturada (ex.: SQL). Motores de busca retornam listas de URLs; bases de dados retornam tabelas organizadas.

1. Tabela Livros: ISBN (chave primária), Título, Autor, Ano. Tabela Membros: ID\_Membro (chave primária), Nome, Contacto. Tabela Empréstimos: ID\_Empréstimo (chave primária), ISBN (chave estrangeira), ID\_Membro (chave estrangeira), Data\_Empréstimo, Data\_Devolução. Relações: Cada empréstimo está ligado a um livro e a um membro.

**Parte 4: Base de Dados - Aplicações e Modelos** 7. É um software que gerencia dados, permitindo armazenamento, consulta, modificação e remoção. Oferece controlo de acesso, integridade e recuperação de dados.

1. Banca: Gerenciamento de contas e transações. Universidades: Gestão de inscrições, notas e turmas. Vendas: Controle de inventário e histórico de compras.
2. Objetivo: Reduzir a duplicação de dados. Como resolvem: Centralizando o armazenamento e aplicando restrições de integridade.
3. Atomicidade garante que uma transação ocorra completamente ou não ocorra. Exemplo: Transferência bancária; o débito e o crédito devem ser realizados juntos ou nenhum ocorre.

4. Modelo que organiza dados em tabelas (relações), com colunas representando atributos e linhas representando registros.
5. Físico: Como os dados são armazenados. Lógico: Quais dados existem e como estão relacionados. Visualização: O que os utilizadores podem ver, ocultando detalhes desnecessários.
6. Esquema físico: Organização real no disco. Esquema lógico: Estrutura geral e relações. Instância: Conjunto de dados atual na base de dados.
7. DDL define estruturas de dados e restrições. Exemplo:

```
CREATE TABLE Instrutor (  
    ID CHAR(5),  
    Nome VARCHAR(20),  
    Nome_Departamento VARCHAR(20),  
    Salario NUMERIC(8,2)  
);
```

1. Capacidade de modificar o esquema físico sem alterar o lógico. Benefício: Aplicações continuam a funcionar mesmo com mudanças no armazenamento.
2. 1950s-1960s: Processamento sequencial com fitas magnéticas. 1970s: Introdução do modelo relacional (Ted Codd). 1980s: Surgimento de sistemas comerciais e adoção de SQL. 1990s: Armazéns de dados e suporte a decisões. 2000s: NoSQL e análise de grandes volumes de dados. 2010s: Sistemas paralelos e integração de SQL com tecnologias modernas.