

Base de Dados

Introdução

António Gonçalves

Separação entre dados e programas

A separação entre dados e programas é um princípio que garante que os **dados** (informações armazenadas) e os **programas** (aplicações que utilizam esses dados) sejam geridos de forma **independente**. Esta separação permite que alterações nos dados ou na sua estrutura não afetem diretamente o funcionamento das aplicações, e vice-versa.

A separação implica que os detalhes sobre como os dados são armazenados ou organizados estão **abstraídos** do programa. O programa interage com os dados através de uma **interface definida**, como consultas **SQL** num **SGBD** ou bibliotecas específicas para manipulação de ficheiros.

O Que é uma Base de Dados

1. Uma **base de dados** é uma coleção de **dados**.
 - 1.1. Um **dado** é um **facto** que deve ser armazenado (**persistido**) e possui um **significado implícito**.
 - 1.2. Um exemplo de **dado** é "João, 30 anos".
 - 1.3. **Persistido** significa que os **dados** são armazenados de forma **duradoura**, permanecendo disponíveis mesmo após o **desligamento do sistema**.
2. Diz respeito a algum **aspecto do mundo real** e é criada com foco num **propósito específico**.
 - 2.1. Exemplos de **propósitos** incluem **sistemas de gestão financeira**, **lojas online**, **hospitais** ou **sistemas educativos**.
3. Tem uma **estrutura lógica** que confere **significado** aos **dados**.
 - 3.1. Um exemplo de **estrutura lógica** é "Maria, 25 anos, Lisboa", organizado em **campos** como **nome**, **idade** e **cidade**.

Sistemas de gestão de bases de dados (SGBD)

O SGBD é o software que gere a base de dados, fornecendo uma interface para a criação, manutenção e manipulação dos dados.

Principais responsabilidades do SGBD:

- **Armazenamento:** Garante que os dados sejam gravados e mantidos de forma persistente.
- **Recuperação:** Facilita a consulta e recuperação dos dados armazenados.
- **Segurança:** Controla o acesso aos dados, permitindo apenas a utilizadores autorizados.
- **Integridade:** Garante que os dados estão consistentes e livres de erros.
- **Independência:** Permite que as aplicações acessem aos dados sem se preocuparem com os detalhes de armazenamento.

Sistemas de gestão de bases de dados (SGBD)

SGBD	Principais Características
MySQL	<ul style="list-style-type: none">- Gratuito (GPL) e disponível em versão comercial.- Suporte a SQL.- Amplamente utilizado em aplicações web.
PostgreSQL	<ul style="list-style-type: none">- Open source.- Suporte avançado a consultas complexas e JSON.- Foco em integridade e extensibilidade.
MariaDB	<ul style="list-style-type: none">- Derivado do MySQL (100% compatível).- Mais rápido para operações complexas.- Suporte a armazenamento distribuído.
SQL Server	<ul style="list-style-type: none">- Produto Microsoft.- Integração com aplicações Microsoft (e.g., .NET, Power BI).- Suporte a Big Data.
Oracle Database	<ul style="list-style-type: none">- Solução comercial de alta performance.- Suporte avançado a clustering e Big Data.- Segurança robusta.

Base de Dados VS SGBD

Aspecto	Base de Dados (BD)	Sistema de Gestão de Bases de Dados (SGBD)
Definição	Coleção organizada de dados armazenados de forma persistente.	Software que gere e controla o acesso à base de dados.
Função	Armazenar os dados.	Facilitar a manipulação, consulta e manutenção dos dados.
Responsabilidade	Guardar informações de forma estruturada ou não.	Gerir o armazenamento, segurança, integridade e recuperação.
Interface	Não possui interface própria para acesso direto.	Oferece interfaces (como SQL) para interação com os dados.
Segurança	Não controla o acesso diretamente.	Garante acesso restrito aos dados por utilizadores autorizados.
Integridade	Apenas contém os dados armazenados.	Verifica e mantém a consistência e validade dos dados.
Exemplo	Uma tabela de nomes e idades.	MySQL, PostgreSQL, Oracle Database, MongoDB.

SGBD: Separação entre Dados e Programas

O SGBD utiliza o princípio de separação de dados e programas ao:

1. **Armazenar os dados:** Organiza-os em estruturas específicas, como tabelas, índices e ficheiros, geridos de forma independente da lógica de negócio.
2. **Delegar a lógica de aplicação:** Mantém a lógica de processamento e regras de negócio na camada de aplicação, que interage com o SGBD através de linguagens como SQL.
3. **Fornecer mecanismos de gestão:** Oferece ferramentas para consultas, transações, backups e controlo de acesso, enquanto a aplicação decide como processar e utilizar os dados recuperados.

Esta separação garante flexibilidade, manutenção simplificada e reutilização dos dados em

Aplicações que NÃO utilizam um SGBD

- **Descrição:**

Nesses sistemas, as estruturas de dados (como tabelas ou ficheiros) são integradas diretamente no código da aplicação. Isso significa que:

- O **programador é responsável** por implementar as funcionalidades de armazenamento e recuperação de dados.
- O controlo de acesso aos dados (quem pode ler, escrever ou alterar) também é gerido diretamente no código da aplicação.

- **Impactos:**

- **Manutenção complexa:** Se a estrutura dos dados mudar (por exemplo, acrescentar um novo campo), é necessário alterar o código da aplicação, o que aumenta os custos e o tempo de manutenção.
- **Pouca flexibilidade:** Cada aplicação precisa ser personalizada para gerir os dados, o que dificulta a reutilização e integração com outros sistemas.

EXEMPLO: Aplicações que NÃO utilizam um SGBD

Uma aplicação simples que armazena os dados dos clientes de uma loja num ficheiro de texto (`clientes.txt`). Cada linha do ficheiro contém informações sobre um cliente, como nome, contacto e endereço, separadas por vírgulas.

Exemplo de conteúdo do ficheiro:

 Copy code

```
João Silva, 912345678, Rua das Flores, Lisboa  
Maria Santos, 915678123, Avenida Central, Porto
```

- **Funcionamento:**
 - A aplicação lê o ficheiro para consultar os dados.
 - Quando um novo cliente é registado, a aplicação abre o ficheiro em modo de escrita e adiciona a nova linha.
 - Para editar ou eliminar um cliente, o programa precisa reescrever o ficheiro completo, excluindo ou alterando os registos necessários.

Aplicações que utilizam um SGBD

- **Descrição:**

Nesses sistemas, o SGBD atua como intermediário entre a aplicação e os dados. A aplicação apenas utiliza o SGBD para aceder aos dados, sem precisar preocupar-se com os detalhes do armazenamento ou controlo de acesso.

- **Benefícios:**

- **Armazenamento automatizado:** O SGBD gere onde e como os dados são armazenados (em tabelas, índices, ficheiros, etc.).
- **Controlo de acesso centralizado:** O SGBD implementa políticas de segurança para definir quem pode aceder e modificar os dados.
- **Facilidade de manutenção:** Alterações na estrutura dos dados (como adicionar um novo campo) não requerem mudanças no código da aplicação, pois o SGBD abstrai esses detalhes.

EXEMPLO: Aplicações que utilizam um SGRD

Uma loja online, como um e-commerce, utiliza um **SGBD** para gerir e armazenar informações sobre produtos, clientes, encomendas e pagamentos.

- **Estrutura dos Dados no SGBD:**

- **Tabela "Produtos":** Contém informações sobre os produtos disponíveis, como nome, descrição, preço e stock.
- **Tabela "Clientes":** Armazena dados dos clientes, como nome, email, morada e histórico de compras.
- **Tabela "Encomendas":** Regista as compras realizadas, incluindo os produtos comprados, quantidade, valor total e data.

- **Operações Realizadas pela Aplicação:**

- **Consulta:** Quando um cliente pesquisa por produtos, a aplicação utiliza o SGBD para consultar a tabela "Produtos".
 - Exemplo de SQL: `SELECT * FROM Produtos WHERE categoria = 'Eletrónica';`
- **Inserção:** Quando um cliente se regista, os seus dados são inseridos na tabela "Clientes".
 - Exemplo de SQL: `INSERT INTO Clientes (nome, email, morada) VALUES ('João Silva', 'joao@email.com', 'Lisboa');`

Comparação

Aspeto	Sem SGBD	Com SGBD
Armazenamento	Dados geridos diretamente pelo programa, armazenados em ficheiros simples, como <code>clientes.txt</code> .	SGBD gere o armazenamento, utilizando tabelas em sistemas como MySQL ou PostgreSQL.
Controlo de Acesso	Implementado manualmente no programa, com maior risco de falhas.	Gerido centralmente pelo SGBD, como permissões configuradas no PostgreSQL.
Manutenção	Alterações na estrutura de dados obrigam a modificar o código do programa.	Alterações nos dados são feitas diretamente no SGBD, sem impacto no programa.
Independência	Dados estão acoplados ao programa, dificultando integração e reutilização.	Dados são independentes do programa, facilitando integração com outras aplicações.
Escalabilidade	Limitada, com dificuldades para grandes volumes ou acessos simultâneos.	Altamente escalável, adequado para grandes volumes e múltiplos utilizadores simultâneos.
Exemplo	Um programa que guarda registos de clientes num ficheiro de texto (<code>clientes.txt</code>).	Uma aplicação de loja online que utiliza MySQL para gerir clientes e produtos.

Exemplo Base de Dados Faculdade

- Neste texto, vamos utilizar uma base de dados universitária para ilustrar todos os conceitos
- Os dados consistem em informações sobre:
 - Alunos
 - Inscrições
 - Cadeiras
- Exemplos de programas de aplicação:
 - Adicionar novos alunos, Inscrições e cursos
 - Registrar alunos em cursos e gerar listas de turmas
 - Atribuir notas aos alunos, calcular as médias das notas (GPA) e gerar

Universo em Base de Dados

Definição de Universo em Base de Dados

O **universo** de uma base de dados refere-se ao **contexto ou domínio total** que a base de dados foi projetada para representar. Ele define o **mini-mundo real** que está a ser modelado, incluindo todas as entidades, atributos e relações relevantes.

Exemplo: Base de Dados de uma Faculdade

Universo:

O universo é uma **faculdade**, e a base de dados representa informações sobre **alunos**, **disciplinas** e **inscrições**.

Tabelas:

1. CADEIRA:

- Regista as disciplinas oferecidas.
- Campos: Código da Cadeira (**CodCad**) e Nome da Disciplina (**Nome**).

2. ALUNO:

- Contém os dados dos alunos.
- Campos: Número Mecânico (**NumMec**), Nome do Aluno (**Nome**) e Curso em que estão matriculados (**Curso**).

3. INSCRIÇÃO:

- Liga os alunos às disciplinas em que estão inscritos.
- Campos: Número Mecânico do Aluno (**NumMec**) e Código da Cadeira (**CodCad**).

Base de Dados: Faculdade

CADEIRA	
CodCad	Nome
12347	Bases de Dados
34248	Álgebra
32439	Introdução aos Computadores

ALUNO		
NumMec	Nome	Curso
798764544	João Pinto	LCC
345673451	Carlos Semedo	MIERSI
487563546	Maria Silva	LBIO
452212348	Pedro Costa	LMAT

INSCRIÇÃO	
NumMec	CodCad
798764544	12347
345673451	12347
798764544	34248
452212348	32439

- **Universo** = uma faculdade
- **Dados**: alunos, cadeiras, inscrições em cadeiras.

Base de Dados: Faculdade

Tabela

ALUNO		
NumMec	Nome	Curso
798764544	João Pinto	LCC
345673451	Carlos Semedo	MIERSI
487563546	Maria Silva	LBIO
452212348	Pedro Costa	LMAT

Registo
(linha)

Campo (coluna)

- Numa base de dados relacional, a informação é organizada em **tabelas**, onde cada tabela contém um conjunto de **registos** (ou **linhas**), e cada registo é subdividido em múltiplos campos (ou **colunas**), que representam os **atributos** das entidades armazenadas.

Inter-relacionamento e consistência

CADEIRA	
CodCad	Nome
<u>12347</u>	Bases de Dados
<u>34248</u>	Álgebra
<u>32439</u>	Introdução aos Computadores

ALUNO		
NumMec	Nome	Curso
<u>798764544</u>	João Pinto	LCC
<u>345673451</u>	Carlos Semedo	MIERSI
<u>487563546</u>	Maria Silva	LBIO
<u>452212348</u>	Pedro Costa	LMAT

INSCRIÇÃO	
NumMec	CodCad
798764544	12347
345673451	12347
798764544	34248
452212348	32439

- Os dados numa BD relacionam-se entre si de acordo com o universo modelado e de forma **consistente**. No exemplo não existem dois alunos com o mesmo n° mecanográfico, e uma inscrição refere-se a um alunos e uma cadeira que existem

consistência

CADEIRA	
CodCad	Nome
<u>12347</u>	Bases de Dados
<u>34248</u>	Álgebra
<u>32439</u>	Introdução aos Computadores

Consistência dos Dados na Tabela "CADEIRA"

A **consistência dos dados** refere-se à garantia de que as informações armazenadas na tabela "CADEIRA" seguem as **regras de integridade** e são coerentes com o universo modelado (neste caso, as cadeiras de uma faculdade). Para assegurar consistência, o sistema de gestão de bases de dados (SGBD) aplica restrições e validações específicas.

Regras de Integridade em Base de Dados

As **regras de integridade** são **restrições** ou **condições** impostas a uma base de dados para garantir que os dados sejam consistentes, válidos e coerentes com o universo modelado. Estas regras ajudam a manter a fiabilidade e a qualidade das informações armazenadas, evitando duplicações, inconsistências ou referências inválidas.

consistência

CADEIRA	
CodCad	Nome
<u>12347</u>	Bases de Dados
<u>34248</u>	Álgebra
<u>32439</u>	Introdução aos Computadores

1. Chave Primária (CodCad):

- O campo **CodCad** (Código da Cadeira) é definido como **único** na tabela, garantindo que cada disciplina tem um código exclusivo.
- **Consistência:** Não pode haver duas cadeiras com o mesmo **CodCad**.
 - **Exemplo correto:** "12347" refere-se apenas a "Bases de Dados".
 - **Exemplo incorreto:** Inserir outra cadeira com o mesmo código "12347" seria inválido.

2. Integridade dos Dados:

- Os valores inseridos em **CodCad** e **Nome** devem ser válidos e completos.
- **Consistência:**
 - **CodCad** deve ter um formato válido (ex.: numérico).
 - **Nome** da cadeira não pode ser nulo (obrigatório).

consistência

CADEIRA	
CodCad	Nome
<u>12347</u>	Bases de Dados
<u>34248</u>	Álgebra
<u>32439</u>	Introdução aos Computadores

Consistência dos Dados na Tabela "CADEIRA"

A **consistência dos dados** refere-se à garantia de que as informações armazenadas na tabela "CADEIRA" seguem as **regras de integridade** e são coerentes com o universo modelado (neste caso, as cadeiras de uma faculdade). Para assegurar consistência, o sistema de gestão de bases de dados (SGBD) aplica restrições e validações específicas.

Dados não consistência

ALUNO		
NumMec	Nome	Curso
<u>798764544</u>	João Pinto	NULL
<u>345673451</u>	Carlos Samedo	12345
<u>798764544</u>	NULL	LBIO
<u>452212348</u>	Pedro Costa	LMAT

No exemplo existem várias **inconsistência**.

Inconsistências ocorrem quando os dados armazenados violam regras de **integridade**, estão em conflito ou não refletem corretamente a realidade que deveriam representar

Inter-relacionamento

CADEIRA		
CodCad	Nome	
<u>12347</u>	Bases de Dados	
<u>34248</u>	Álgebra	
<u>32439</u>	Introdução aos Computadores	

ALUNO		
NumMec	Nome	Curso
<u>798764544</u>	João Pinto	LCC
<u>345673451</u>	Carlos Semedo	MIERSI
<u>487563546</u>	Maria Silva	LBIO
<u>452212348</u>	Pedro Costa	LMAT

INSCRIÇÃO	
NumMec	CodCad
798764544	12347
345673451	12347
798764544	34248
452212348	32439

O inter-relacionamento de tabelas é um conceito que define como as tabelas de uma base de dados estão ligadas entre si para representar as relações existentes no universo modelado. Esse inter-relacionamento é essencial para organizar os dados de forma estruturada, coerente e eficiente, permitindo a sua consulta e manipulação.

Inter-relacionamento: Dados não consistência

CADEIRA	
CodCad	Nome
<u>12347</u>	Bases de Dados
<u>34244</u>	Álgebra
<u>32439</u>	Introdução aos Computadores

ALUNO		
NumMec	Nome	Curso
<u>798764544</u>	João Pinto	NULL
<u>345673451</u>	Carlos Samedo	12345
<u>798764544</u>	NULL	LBIO
<u>452212348</u>	Pedro Costa	LMAT

INSCRIÇÃO	
NumMec	CodCad
798764544	12347
345673451	12347
798764545	34248
452212346	32439

Aplicações de BD – modelos típicos

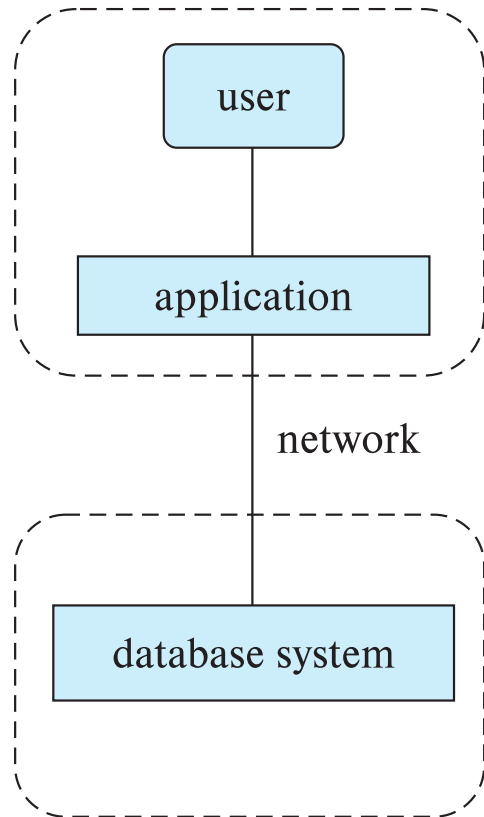
- **Arquitetura de duas camadas:**

- a aplicação reside na máquina do cliente, onde invoca a funcionalidade do sistema de base de dados na máquina do servidor

- **Arquitetura de três camadas:**

- a máquina cliente funciona como um front end e não contém quaisquer chamadas diretas à base de dados.
- A extremidade do cliente comunica com um servidor de aplicações, normalmente através de uma interface de formulários.
- O servidor de aplicações, por sua vez, comunica com um sistema de base de dados para aceder aos dados.

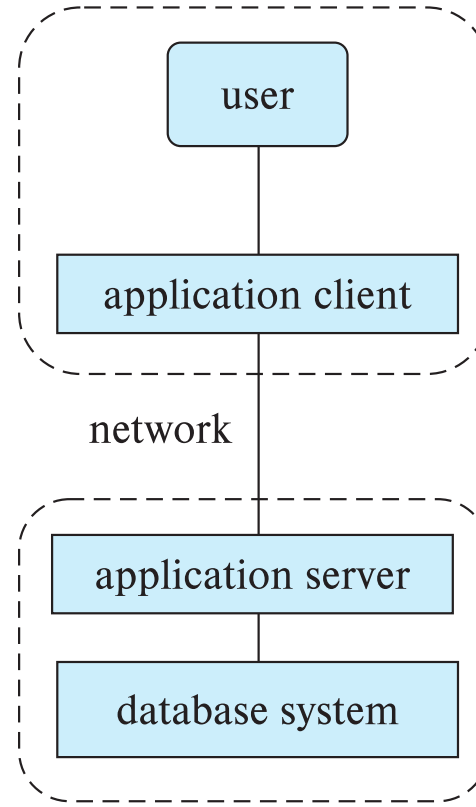
Aplicações de BD – modelos típicos



(a) Two-tier architecture

client

server



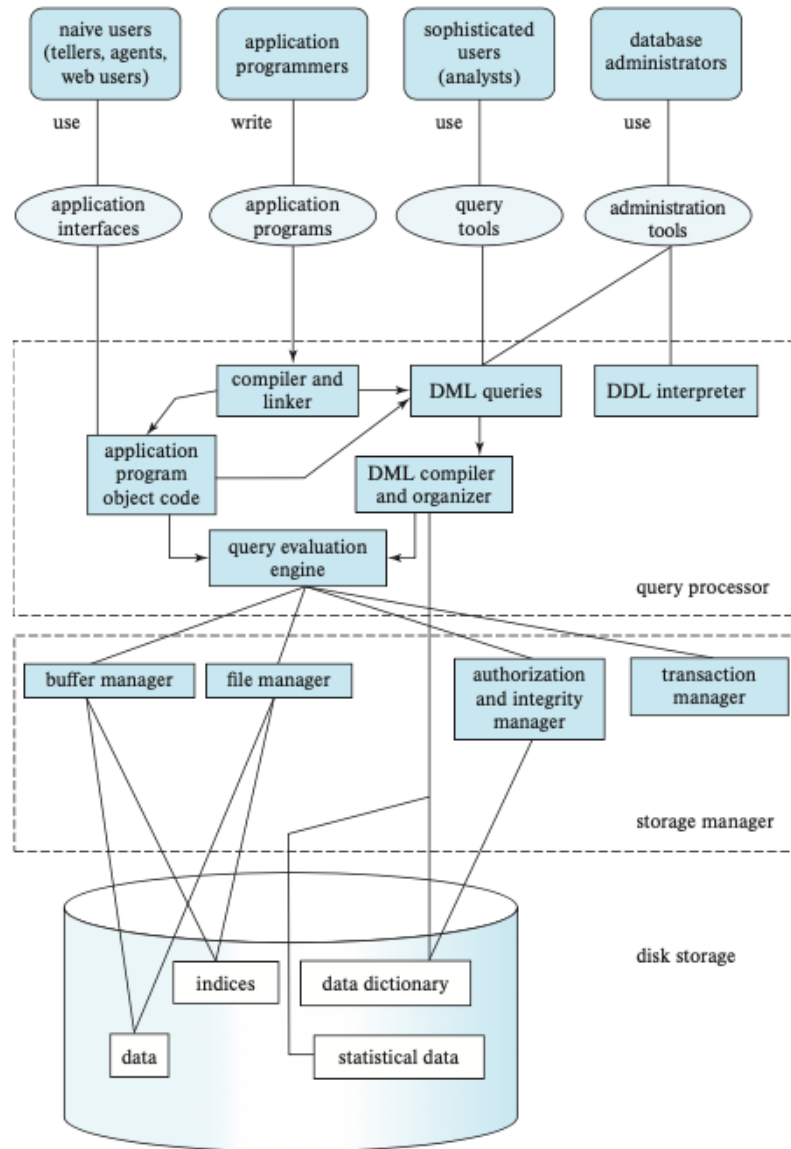
(b) Three-tier architecture

apresentação dos dados

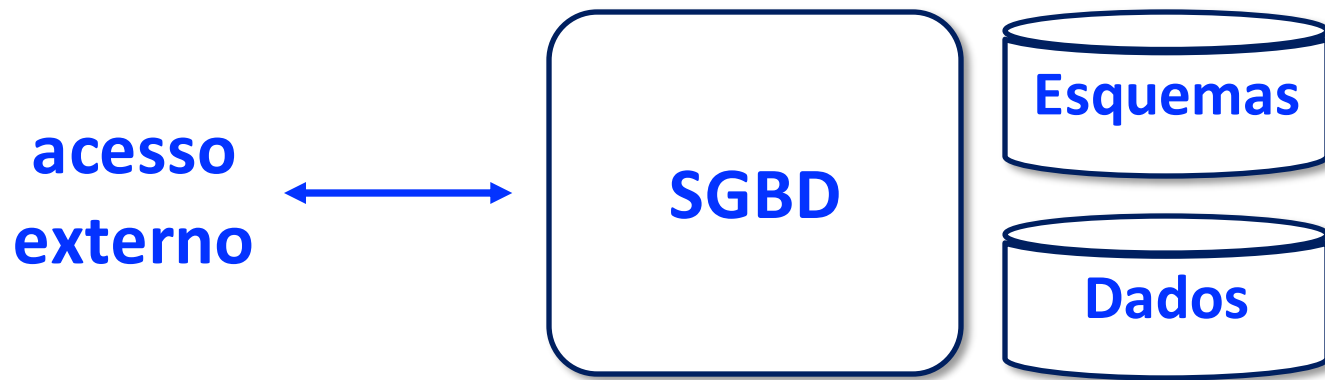
lógica aplicacional

Dados

Utilizadores da base de dados



SGBD: esquemas e dados



- Um SGDB permite a definição e manipulação de:
 - **Esquemas de base de dados**, que descrevem como uma base de dados está estruturada, também chamados de meta-dados.
 - **Bases de dados = instância de esquemas.**
- Um SGBD pode suportar BDs distintas com o mesmo esquema e a existência de vários esquemas distintos.

SGBD: esquemas e dados

Esquema

O **esquema** define a estrutura da tabela, ou seja, as colunas (atributos) e os respetivos tipos de dados:

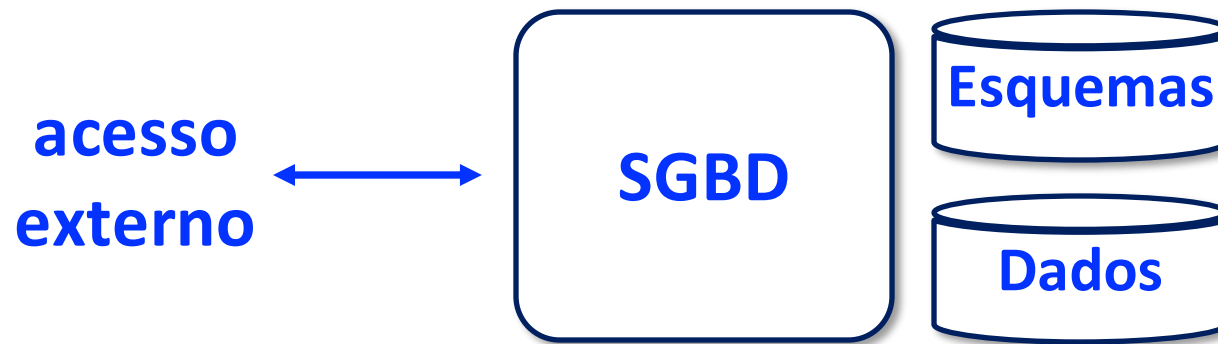
Nome da Coluna	Tipo de Dados	Descrição
ID	INT	Identificador único do cliente
Nome	VARCHAR(50)	Nome do cliente
Endereco	VARCHAR(100)	Endereço do cliente
Telefone	VARCHAR(15)	Número de telefone do cliente

Instância

A **instância** é o conteúdo atual da tabela, ou seja, os dados armazenados no momento:

ID	Nome	Endereco	Telefone
1	João Silva	Rua A, Lisboa	912345678
2	Maria Costa	Avenida B, Porto	913456789
3	Carlos Nogueira	Praça C, Coimbra	914567890

Como se “fala” com uma base de dados?



- **SQL (Structured Query Language)** é a linguagem padrão usada para interagir com bancos de dados relacionais.
- Abrange vários tipos de comandos agrupados em categorias:
 - **DDL - linguagem de definição de dados:** para definir e manipular esquemas;
 - **DML - linguagem de manipulação de dados:** para manipular (alterar) dados.
 - **DCL - Linguagem de Controle de Dados:** para gerenciar permissões e segurança

Como se “fala” com uma base de dados?

Resumo das Categorias e Principais Siglas

Categoria	Principais Comandos
DDL	CREATE , ALTER , DROP , TRUNCATE , RENAME
DML	SELECT , INSERT , UPDATE , DELETE , MERGE
TCL	COMMIT , ROLLBACK , SAVEPOINT
DCL	GRANT , REVOKE
QL	SELECT , WHERE , GROUP BY , HAVING , ORDER BY

DDL - linguagem de definição de dados

- Linguagem para definir o modelo de dados
- **Criação de Estruturas:** Cria tabelas, índices e outros objetos no banco.
- **Alteração de Estruturas:** Modifica tabelas ou objetos existentes.
- **Exclusão de Estruturas:** Remove objetos como tabelas e índices.
- **Definição de Restrições:** Configura regras como chaves primárias e estrangeiras.
- **Manipulação de Esquemas:** Organiza objetos em diferentes contextos no banco.

DDL - linguagem de definição de dados

ALUNO		
NumMec	Nome	Curso

```
CREATE TABLE ALUNO (  
  NumMec INTEGER PRIMARY KEY NOT NULL,  
  Nome VARCHAR(64) NOT NULL,  
  Curso VARCHAR(20) NOT NULL  
);
```

- SQL usada como DDL
 - criação de uma tabela via instrução **CREATE**

TABLE

DML - linguagem de manipulação de dados

- Linguagem para aceder e atualizar os dados organizados pelo modelo de dados adequado
- **Inserção de Dados:** Adiciona novos registros, como cadastrar um cliente com nome e e-mail.
- **Consulta de Dados:** Busca e visualiza dados armazenados, podendo filtrar por critérios específicos.
- **Atualização de Dados:** Altera informações existentes, como atualizar o endereço de um cliente.
- **Exclusão de Dados:** Remove registros, como apagar um pedido cancelado.
- **Manipulação em Massa:** Opera em vários registros ou tabelas, como combinar ou atualizar dados em lote.

DML - linguagem de manipulação de dados

ALUNO		
NumMec	Nome	Curso
798764544	João Pinto	LCC
345673451	Carlos Semedo	MIERSI
487563546	Maria Silva	LBIO
452212348	Pedro Costa	LMAT

```
INSERT INTO ALUNO  
(NumMec, Nome, Curso) VALUES  
(798764544, 'João Pinto', 'LCC');
```

- SQL usada como DML
 - Adiciona novos registros via instrução **INSERT INTO**

DML - linguagem de manipulação de dados

ALUNO		
NumMec	Nome	Curso
798764544	João Pinto	LCC
345673451	Carlos Semedo	MIERSI
487563546	Maria Silva	LBIO
452212348	Pedro Costa	LMAT

```
INSERT INTO ALUNO(NumMec, Nome, Curso)
VALUES(798764544, 'João Pinto', 'LCC'),
      (345673451, 'Carlos Semedo', 'MIERSI'),
      (487563546, 'Maria Silva', 'LBIO'),
      (452212348, 'Pedro Costa', 'LMAT');
```

- SQL usada como DML, neste caso para inserção de registos.

DML - linguagem de manipulação de dados

ALUNO		
NumMec	Nome	Curso
798764544	João Pinto	LCC
345673451	Carlos Semedo	MIERSI
487563546	Maria Menezes	LBIO
452212348	Pedro Costa	LMAT

```
SELECT Curso FROM ALUNO  
WHERE NumMec=798764544;
```

consulta

actualização

```
UPDATE ALUNO  
SET Nome='Maria Menezes'  
WHERE NumMec=487563546;
```

remoção

```
DELETE FROM ALUNO  
WHERE Curso='LMAT';
```

- SQL usada como **DML** (**SELECT** / consulta), e de novo como **DML** para actualização (**UPDATE**) ou remoção (**DELETE**) de registos.

DCL - Linguagem de Controle de Dados

- **Concessão de Permissões:** Autoriza utilizadores a realizar ações específicas, como leitura ou escrita.
- **Revogação de Permissões:** Remove permissões anteriormente concedidas.
- **Controlo de Segurança:** Define quem pode aceder ou modificar objetos na base de dados.
- **Permissões em Massa:** Gere permissões para grupos ou papéis de forma coletiva.
- **Gestão de Utilizadores:** Cria, gere e

DCL - Linguagem de Controle de Dados

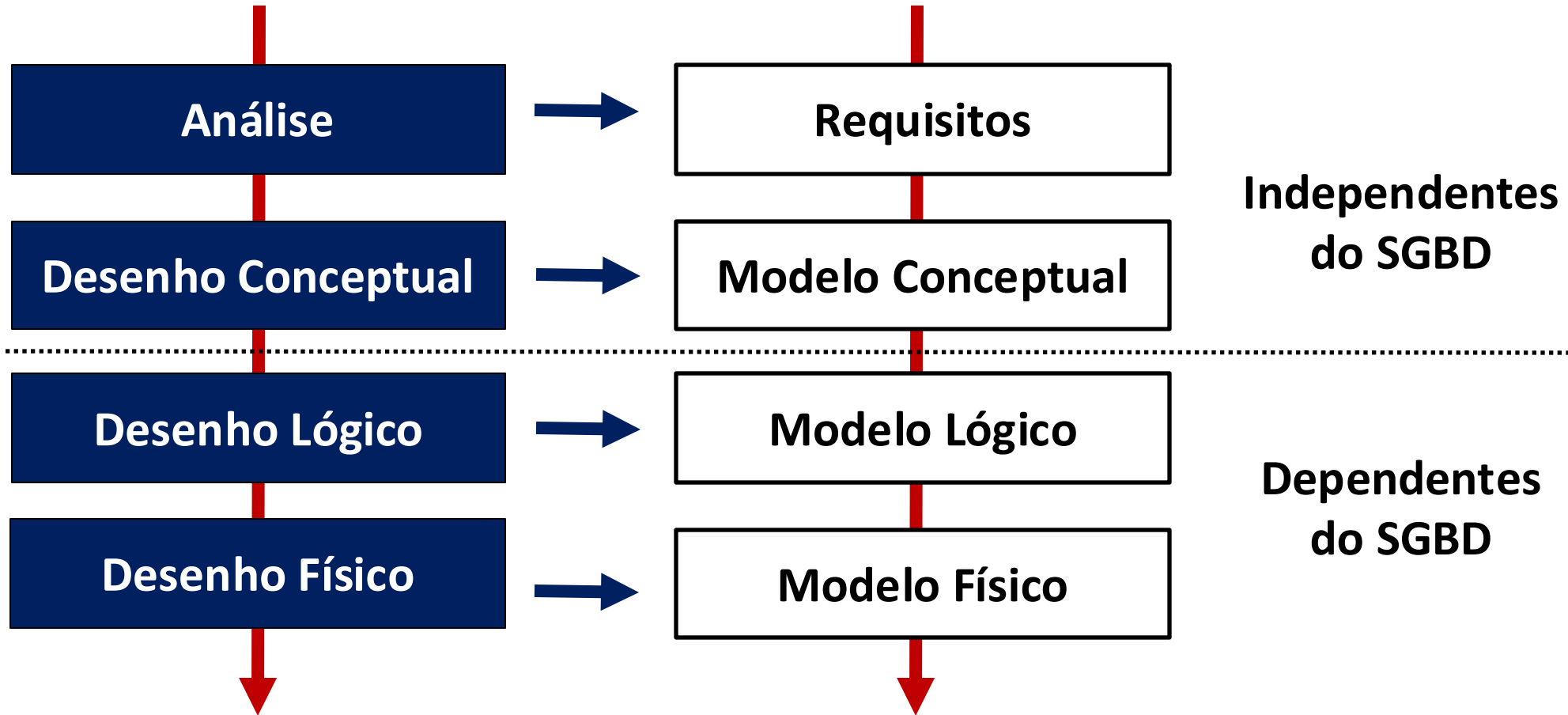
ALUNO		
NumMec	Nome	Curso
798764544	João Pinto	LCC
345673451	Carlos Semedo	MIERSI
487563546	Maria Silva	LBIO
452212348	Pedro Costa	LMAT

```
GRANT INSERT, UPDATE, DELETE  
ON ALUNO  
TO admin@localhost;
```

```
GRANT SELECT  
ON ALUNO  
TO guest@localhost;
```

- SQL como DCL - configuração de permissões de acesso a uma tabela para diferentes utilizadores via instrução **GRANT**.

Desenho e implementação de uma BD



- O desenho e implementação de uma BD compreende várias fases e correspondentes níveis de modelação.

Desenho e implementação de uma BD (cont.)

■ Modelo de Dados

- Conjunto de conceitos que descrevem a estrutura da BD em termos da relação entre dados, seu significado, e restrições.

■ Tipos de modelo:

- **Conceptual**: descreve os conceitos a representar em uma BD de forma independente do SGBD.
- **Lógico**: descreve de forma lógica a implementação de uma BD num SGBD.
- **Físico**: descreve a organização física interna dos dados no SGBD.

Desenho e implementação de uma BD (cont.)

■ **Análise**

- Compreensão do universo para a BD, resultando num conjunto de requisitos documentados.
- A derivação de requisitos pode passar por reuniões entre peritos sobre o universo em causa, entrevistas com potenciais utilizadores / clientes, etc.

■ **Desenho conceptual (ou modelação)**

- Definição de um modelo conceptual a partir dos requisitos.
- O modelo deve descrever as entidades da BD e a forma como se relacionam, de forma independente do SGBD.

Desenho e implementação de uma BD (cont.)

■ Desenho lógico (ou implementação)

- Mapeamento do modelo de dados conceptual num modelo de dados lógico concreto.
- Implementação da BD usando um SGBD.
- Na cadeira iremos considerar o **modelo relacional** e a sua implementação em **SQL**.

■ Desenho físico

- Mapeamento do modelo de dados lógico no modelo de dados físico interno ao SGBD, ex. em termos de parameterização do tipo de armazenamento a usar, optimização do seu uso tendo em conta padrões de acesso, operação em rede, redundância, ...

Base de Dados

Introdução

António Gonçalves

Baseado em:

- slides de Eduardo Marques, utilizado com autorização
- slides Database System Concepts