



iscte

INSTITUTO
UNIVERSITÁRIO
DE LISBOA



emprego
digital



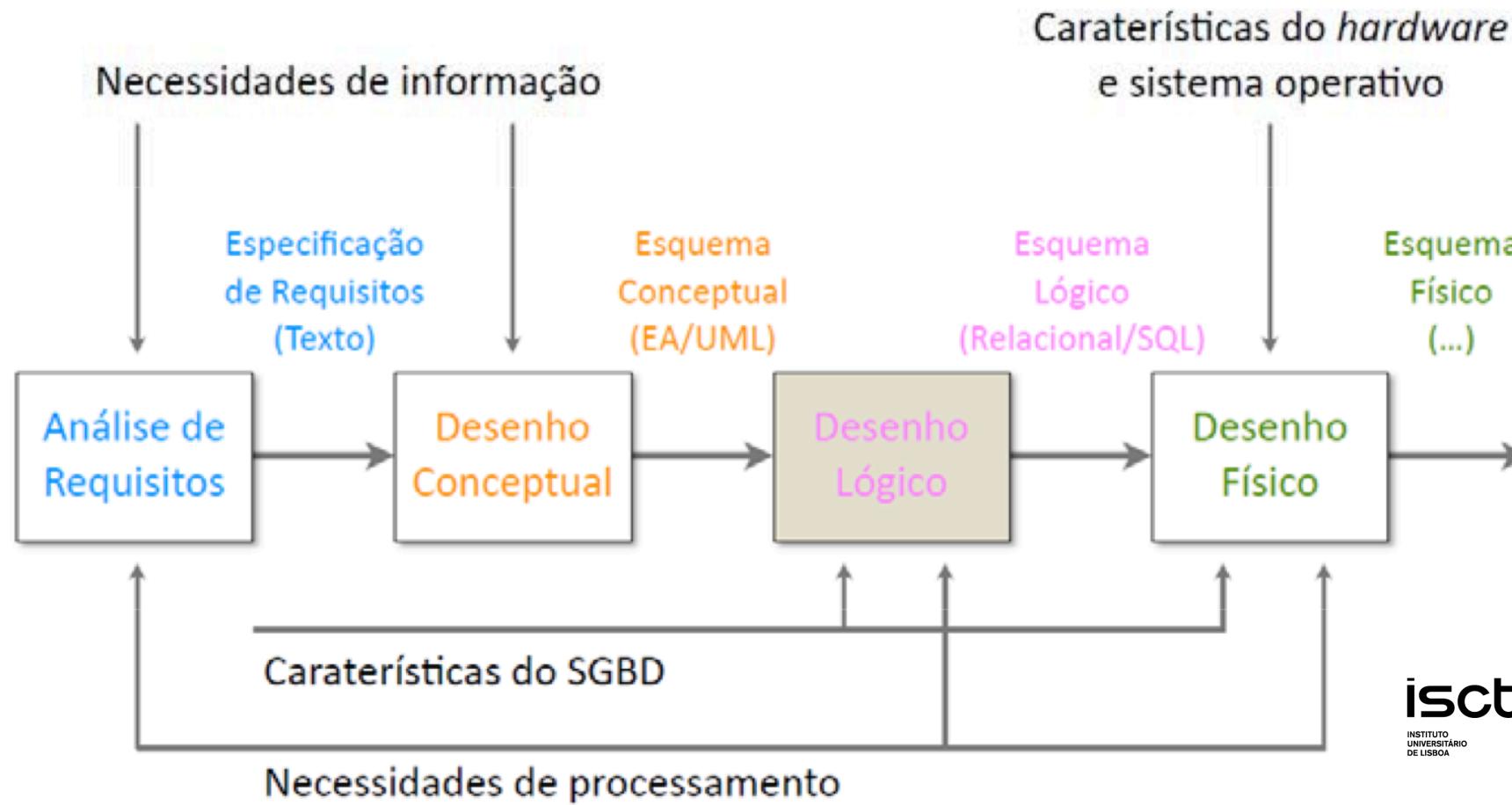
Bases de Dados

Modelo Relacional – I

Sumário e Referências

- Sumário
 - Enquadramento no processo de desenho de BD
 - Vertentes do modelo relacional
 - Estruturas de dados
 - Operadores relacionais
 - Restrições de integridade
 - Propriedades das relações
 - Comandos SQL
 - Criação de tabelas
 - Inserção, atualização e remoção de dados
- Referências
 - R. Ramakrishnan (**capítulo 2**)

Processo de desenho de BD



Modelo Relacional: História

- Criado em 1970 por Edgar Codd
 - Muito simples e elegante
 - Estrutura de dados, operadores relacionais e regras de integridade
 - Modelo formal com forte **base matemática** a suportá-lo
 - Utilizado pela esmagadora **maioria** dos **SGBD**
 - 10 comerciais: Oracle; Outros: MS SQL Server, IBM DB2, MySQL, ...
- Não foi o primeiro modelo de dados
 - Nos anos 1960 já existia o modelo hierárquico e o modelo em rede
 - BD baseados em modelos de dados orientados a objetos
 - alternativa, ainda problemas de concretização

Modelo Relacional: Vertentes

- Estrutura de dados
 - Base de dados é uma **coleção de relações**
 - **Relação** é uma **tabela** com **linhas** e **colunas**
 - Cada coluna tem valores de um **domínio** de dados
- Operadores relacionais
 - Para gestão de tabelas e outras estruturas
 - Para inserção, remoção e pesquisa de dados
- Regras de Integridade
 - Para garantia da coerência dos dados
 - Ex. um aluno para se inscrever em disciplinas tem de estar matriculado

SQL – Structured Query Language

- Linguagem universal/norma para o modelo relacional
 - 1986, norma do American National Standards Institute (ANSI)
 - 1987, também pela International Standard Organization (ISO)
- Apesar de ser uma norma (standard) os SGBD podem introduzir características específicas
- Componente nuclear estável, mas em evolução
 - 1986: primeira formalização do SQL standard (SQL-86)
 - 7 revisões:
 - SQL-89; SQL-92
 - SQL-99 uma grande extensão ao SQL-92
 - SQL2003, SQL2006, SQL2008, SQL2011

Relações

- Relações são **tabelas** com linhas e colunas
 - colunas = atributos = campos
 - linhas = registos = entidades = tuplos
- Esquema de relação inclui
 - Nome da relação, Nome e domínio de dados de cada coluna
 - Ex: Students (sid: integer, name: string, login: string, age: integer, gpa: real)
- Instância da relação
 - Conjunto de tuplos

Propriedades das Relações

- **Restrições de domínio**
 - Especificam o tipo de dados de cada atributo
 - Todas as colunas têm de ter um domínio
 - Ex. string, integer, real
 - SGBD oferecem domínios específicos
 - Ex. $char(n)$, int , $smallint$, $number(n,m)$
- O **grau** da relação = número de colunas
- A **cardinalidade** de uma relação = número de linhas

Base de Dados

- **Base de dados (BD) relacional**
 - coleção de relações com nomes distintos
- **Esquema relacional da BD**
 - coleção dos esquemas de todas as relações
- **Instância da BD relacional**
 - coleção das instâncias de todas as relações

Síntese de Comandos SQL

- **SQL-DDL:** Data Definition Language
 - operações sobre a estrutura das tabelas e gestão de restrições de integridade
 - CREATE TABLE
 - DROP TABLE
 - ALTER TABLE
- **SQL-DML:** Data Manipulation Language
 - operações sobre os dados das tabelas
 - INSERT INTO
 - DELETE FROM
 - UPDATE
 - SELECT

CREATE TABLE: Criação de Tabelas

- Dado um esquema de relação
 - *Students (sid: integer, name: string, login: string, age: integer, gpa: real)*
- O comando CREATE TABLE cria uma nova tabela sem dados

```
CREATE TABLE Students (sid integer,  
                      name char(30),  
                      login char(30),  
                      age integer,  
                      gpa real);
```

INSERT: Inserção de Linhas

<i>sid</i>	<i>name</i>	<i>login</i>	<i>age</i>	<i>gpa</i>
53831	Madayan	madayan@music	11	1.8
53832	Guldu	guldu@music	12	2.0
53650	Smith	smith@math	19	3.8
53666	Jones	jones@cs	18	3.4
50000	Dave	dave@cs	19	3.3

INSERT

```
INTO Students (sid, name, login, age, gpa)
VALUES (53688, 'Smith', 'smith@ee', 18, 3.2)
```

<i>sid</i>	<i>name</i>	<i>login</i>	<i>age</i>	<i>gpa</i>
53831	Madayan	madayan@music	11	1.8
53832	Guldu	guldu@music	12	2.0
53688	Smith	smith@ee	18	3.2
53650	Smith	smith@math	19	3.8
53666	Jones	jones@cs	18	3.4
50000	Dave	dave@cs	19	3.3

UPDATE: Atualização de valores

```
UPDATE Students S  
SET S.age = s.age + 1, s.gpa = s.gpa + 1  
WHERE S.sid = 53688
```

Exemplo: UPDATE

<i>sid</i>	<i>name</i>	<i>login</i>	<i>age</i>	<i>gpa</i>
53831	Madayan	madayan@music	11	1.8
53832	Guldu	guldu@music	12	2.0
53688	Smith	smith@ee	18	3.2
53650	Smith	smith@math	19	3.8
53666	Jones	jones@cs	18	3.4
50000	Dave	dave@cs	19	3.3

```
UPDATE Students S  
SET S.gpa = S.gpa - 0.1  
WHERE S.gpa >= 3.3
```

<i>sid</i>	<i>name</i>	<i>login</i>	<i>age</i>	<i>gpa</i>
50000	Dave	dave@cs	19	3.2
53666	Jones	jones@cs	18	3.3
53688	Smith	smith@ee	18	3.2
53650	Smith	smith@math	19	3.7
53831	Madayan	madayan@music	11	1.8
53832	Guldu	guldu@music	12	2.0

DELETE: Apagar Linhas

```
DELETE  
FROM Student S  
WHERE S.name = 'Smith'
```



O futuro profissional começa aqui

