



Fundamentos de Bases de Dados

Terminologia e Elementos das Bases de Dados

Célia Duarte

UFCD 10797

Ficheiros, Bases de Dados e RDBMS

- > Porquê a necessidade de uma Base Dados?
- > Os primeiros sistemas de informação, baseavam-se em ficheiros "File-Processing Systems", que apresentavam algumas limitações:
 - > Dados duplicados, separados e isolados,
 - > Falta de integridade dos dados,
 - > Aplicações dependentes do formato dos ficheiros,
 - > Tipos de ficheiros incompatíveis,
 - > Dificuldade em retratar a realidade com ficheiros,
 - > Díficil representação dos dados na perspetiva do utilizador.

Ficheiros, Bases de Dados e RDBMS (cont.)

> **Ficheiros**

- > Flat files são mais fáceis de entender;
- > Redundância da informação;
- > Necessário mais espaço em disco;
- > Dificuldade na manutenção da integridade dos dados.

> **Base Dados Relacionais**

- > Divisão dos dados em várias entidades relacionadas entre si;
- > Eliminação de informação duplicada - redundância;
- > Facilidade na pesquisa de dados;
- > Manipulação de dados simplificada;
- > Otimização do espaço em disco;
- > BDs relacionais são mais difíceis de implementar, mas são altamente escaláveis

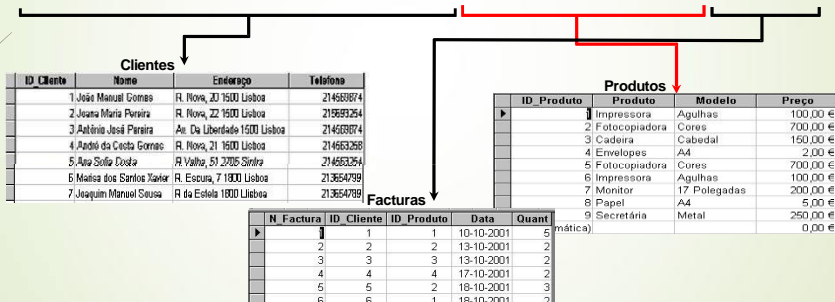
Ficheiros, Bases de Dados e RDBMS (cont.)

> **Vantagens das Bases de Dados**

- > Redução da duplicação de dados,
- > Redução das inconsistências e aumento da integridade dos dados,
- > Formato do repositório dos dados único,
- > Estrutura lógica guardada na Base de Dados juntamente com os dados,
- > Facilidade em retratar/transportar os objetos/entidades da vida real para o Sistema,
- > Aplicações independentes do formato dos dados - os dados são acedidos pelo DBMS, libertando o programador da preocupação de saber como os dados estão guardados fisicamente e protegendo os dados de acessos não autorizados.

Ficheiros vs Bases de Dados

Cliente	Endereço	Telefone	Produto	Modelo	Preço	Data	Qt
João Manuel Gomes	R. Nova, 20 1500 Lisboa	214569874	Impressora	Agulhas	100,00 €	18-10-2001	5
Joana Maria Pereira	R. Nova, 22 1500 Lisboa	215693254	Fotocopiadora	Cores	700,00 €	13-10-2001	2
António José Pereira	Av. Da Liberdade 1500 Lisboa	214569874	Cadeira	Cabedal	150,00 €	13-10-2001	2
André da Costa Gomes	R. Nova, 21 1500 Lisboa	214563258	Envelopes	A4	2,50 €	17-10-2001	2
Ana Sofia Costa	R. Velha, 51 2705 Sintra	214563254	Fotocopiadora	Cores	700,00 €	18-10-2001	3
Marisa dos Santos Xavier	R. Escura, 7 1800 Lisboa	213654799	Impressora	Agulhas	100,00 €	18-10-2001	2



O Modelo Relacional

- A história do modelo relacional começa em 1970 com a publicação por Edgar Frank Codd, do artigo "A Relational Model of Data for Large Shared Data Banks" no ACM Journal.
- > As primeiras implementações dos estudos de E.F.Codd, começaram nos anos 70 com o System R na IBM e INGRES na Universidade de Berkeley na Califórnia.
- > A IBM, com o System R, desenvolveu uma série de linguagens de query (consulta), nomeadamente o SEQUEL, que mais tarde veio a dar origem à Structured Query Language (SQL).
- > A primeira implementação comercial de SQL foi realizada pela Relational Software Inc., hoje conhecida por Oracle.
- > Em 1986, o ANSI (American National Standards Institute) publica um standard SQL; "Systems Application Architecture Database Interface (SAA SQL)".
- > O último standard SQL é de 2019 (https://en.wikipedia.org/wiki/SQL#Current_standard)

Linguagem SQL - Exemplos

Data Definition Language (DDL)

Create Database Teste2

Drop Database Teste2

Data Manipulation Language (DML)

Insert into Clientes values (1,'Célia Duarte','Lisboa')

Select *

From Cliente

Where Codigo_Cliente < 1000

O Que é uma Base de Dados Relacional?

> **Uma Base de Dados Relacional inclui:**

- > Tabelas que contêm um único sujeito/entidade;
- > Tabelas são unidas por relações;
- > Tabelas tem atributos/campos, que podem ser campos chave
- > Primary key – chave primária
- > Foreign key – chave estrangeira ou externa
- > Tipos de Relações (Relationships)
 - > One-to-one
 - > One-to-many
 - > Many-to-many (só indirectamente)

Modelação e Normalização de Dados

> **Passos para a Modelação e normalização de dados:**

- > Conceber o Entity Relationship Diagram (ERD)
- > Identificação de entidades e objetos;
- > Identificação de campos únicos (chaves);
- > Identificação dos atributos (campos/colunas) e respetivo datatype (tipo de dados: int, char, entre outros) relativos a cada entidade;
- > Identificação das relações entre tabelas e criação das chaves primarias (pk) e secundárias (fk).
- > Normalização não é por si o desenho da Base de Dados, mas um processo útil e necessário para o desenho otimizado da mesma.

Normalização

- > Normalização é o processo de remover todas as redundâncias existentes nos dados armazenados.
- > Quando o desenho da Base de Dados está completamente normalizado, não há repetição dos dados através das várias junções realizadas para reunir os dados das tabelas.
- > As vantagens da normalização são em termos de espaço em disco e redução de custos, e também a eficiência com que os dados podem ser mantidos e alterados.
- > A otimização do Sistema por vezes pode levar à desnormalização de parte da Base de Dados, para reduzir o número de **joins** necessários para obter toda a informação, já que este é um dos processos mais consumidores de recursos da base de dados.

Desnormalização

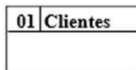
- > Aumento dos dados redundantes e respetivo aumento do espaço em disco necessário;
 - > Tabelas desnormalizadas podem implicar falhas lógicas, que têm de ser compensadas com código extra;
 - > Mais difícil manter a integridade dos dados;
 - > Aumento da performance de tabelas frequentemente utilizadas em queries, tabelas de validação ou reporting;
 - > Tabelas usadas para as operações de Insert, Update e Delete (registos de dados em OLTP – Online Transaction Processing), são pouco susceptíveis de serem desnormalizadas.
- > Ver <https://www.stitchdata.com/resources/oltp-vs-olap/>

Nomenclatura usada em Bases de Dados Relacionais

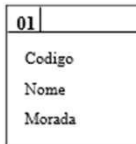
Termo	Significado	Exemplo	Como é Criado
Entidade	Tópico ou Objecto a descrever	Cliente, Encomenda, Produto	Entidades em Bases de Dados são representadas por tabelas
Linha ou Registo	Exemplo de um tópico ou objecto	Todos os dados referentes a um cliente	Adicionando linhas a uma tabela, criando registos únicos na tabela
Coluna ou Campo	Atributo de uma Entidade	Nome do Cliente	Atributos são os campos da tabela
Chave	Identificador unívoco	Código Cliente	Atributos identificados como Chaves.
Relações	Entidades podem estar associadas a outras entidades através de relações	Cliente - Encomenda	A relação cria-se através de DDL ou a nível aplicacional, sempre relacionando chaves de ambas as entidades.

Desenho de Bases de Dados Relacionais

Entidade Nível 0



Entidade Nível 1



Tipo de Relações entre Entidades

Um para Um (1:1)

Um para muitos (1:N)

Muitos para muitos (N:N)



Entidade Associativa

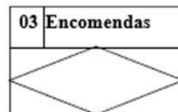
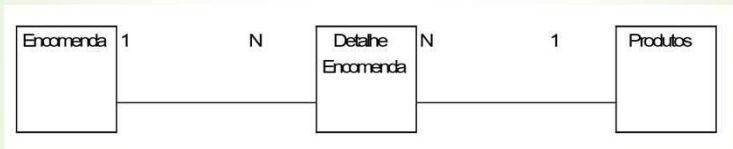


Diagrama de Entidades e Associações (DEA ou ERD)

- > Uma encomenda têm vários produtos e um produto pode estar em várias encomendas (N:N).
- > Uma encomenda têm várias entradas na entidade DetalheEncomenda (1:N).
- > Um produto está em vários detalhes (1:N)



- > Desta forma com a entidade associativa **Detalhe Encomenda**, podemos saber quais os produtos de uma encomenda e em que encomendas estão os produtos.

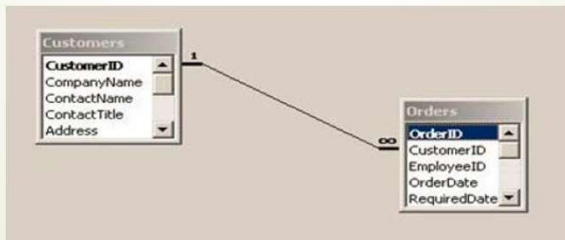
One-to-One Relationships

- > – Apenas um registo idêntico;
- > – É utilizada a Primary Key em ambas as tabelas;
- > – É usada para limitar/facilitar o acesso à informação.



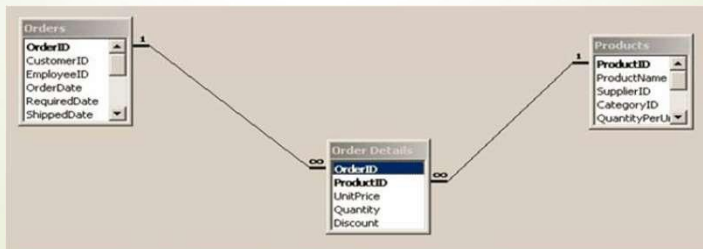
One-to-Many Relationships

- > – O tipo de relação mais comum
- > – Relacionado entre Primary Key e Foreign Key
- > – Pode ter muitos registos relacionados
- > – Integridade Referencial evita registos órfãos.



Many-to-Many Relationshps

- > – Uma encomenda, muitos produtos;
- > – Um produto, muitas encomendas;
- > – Relação não suportada diretamente entre tabelas pelos SGBDs;
- > – Necessário usar uma tabela de ligação (junction table) para relacionar ambas as tabelas.



Integridade Referencial

- > Regras para preservar relações entre tabelas
- > Previnem registos orfãos
 - ☐ Não permitem adicionar registos do lado N
 - ☐ Não permitem eliminar registos do lado 1
- > Cascade Update – permissão para alterar em tabelas relacionadas
- > Cascade Delete – permissão para apagar em tabelas relacionadas (muito perigoso)

Resumo - O que é uma BD Relacional

- Dados são armazenados em tabelas, que são constituídas por linhas e colunas e podem ser relacionadas entre si se tiverem colunas com o mesmo tipo, tamanho e dados (chaves).
- Campos Chave
 - > Chave Primária (Primary key) - PK
 - > Chave Externa (Foreign key) - FK
- Relações (Relationships)
 - > Um para Um (One-to-one) = PK <-> PK
 - > Um para muitos (One-to-many) = PK -> FK
 - > Muitos para muitos (Many-to-many) = Tabela Associativa (junction table) = PK -> FK & FK->PK
- Integridade Referencial
 - > Evita registos orfãos;
 - > Actualização ou eliminação em cascata (Cascade update/Cascade delete).

Diagrama Entidades/Associação - Exercício

- Objectivo Criar o Diagrama Entidades e Associações para retratar um processo de venda de um produto.
- A Base de Dados tem as seguintes entidades:
 - Morada, Telefone, E-mail, Moeda, País, Entidade (neste caso Cliente), ContaCorrente, CódigoPostal, Encomenda, FaturaCliente, DetalheEncomenda, DetalheFatura, Produto, CategoriaProduto.
 - Indique alguns dos atributos que considere importantes (nomeadamente as chaves e como as tabelas se relacionam entre si).