
Modelo Relacional

Sistemas de Informação I

- Estes slides contém informação adaptada de material pedagógico disponibilizado por:
 - Helder Pita
 - Walter Vieira

Requisitos da informação

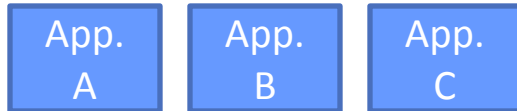
- **Actualidade:** O valor da informação depende da sua actualidade, pois só com base em informações actuais se podem tomar decisões acertadas. Os sistemas devem fornecer informação actual e conter mecanismos de actualização eficiente da informação;
- **Correcção:** Não chega a informação ser actual, é necessário que esta seja correcta. Só com informação rigorosa se podem tomar decisões correctas;

Requisitos da informação (cont.)

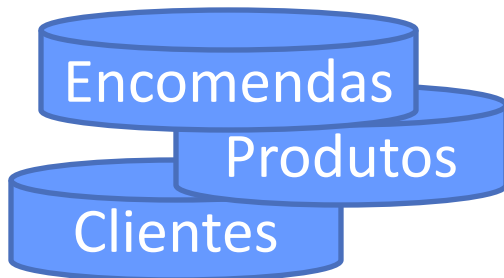
- **Relevância:** A informação deve ser filtrada de modo a apresentar apenas aquela que é relevante. Grandes quantidades de informação tornam mais lentas as tomadas de decisão;
- **Disponibilidade:** Se a informação não está disponível quando é necessária, torna-se inútil. Cada vez mais as tomadas de decisão são quase instantâneas;
- **Legibilidade:** A informação deve ser apresentada num formato que seja legível e facilmente interpretada pelo utilizador.

Motivação para utilização de um SGBD

Depart. de Processamento de Encomendas



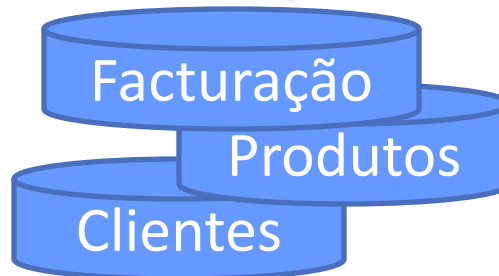
Sist. Gestão Encomendas



Depart. de contabilidade



Sist. De Facturação



Depart. de tesouraria



Sist. de Facturação



Armazenamento baseado em ficheiros

- **Factos:**

- Aplicações informáticas acedem e manipulam dados
- Aplicações distribuídas
- Partilha de dados entre diversas aplicações
- Garantia de consistência nos dados

- **Problemas:**

- Forte dependência entre dados e programas: um programa precisa de ter a estrutura dos dados para os poder manipular;
- Duplicação de dados;
- Neste modelo cada aplicação possui os seus próprios ficheiros. A partilha é difícil. A manutenção do sistema é complicada e exige muito esforço;



Questão: Como resolver o problema da partilha de dados ?

Solução

Separar aplicações dos dados !

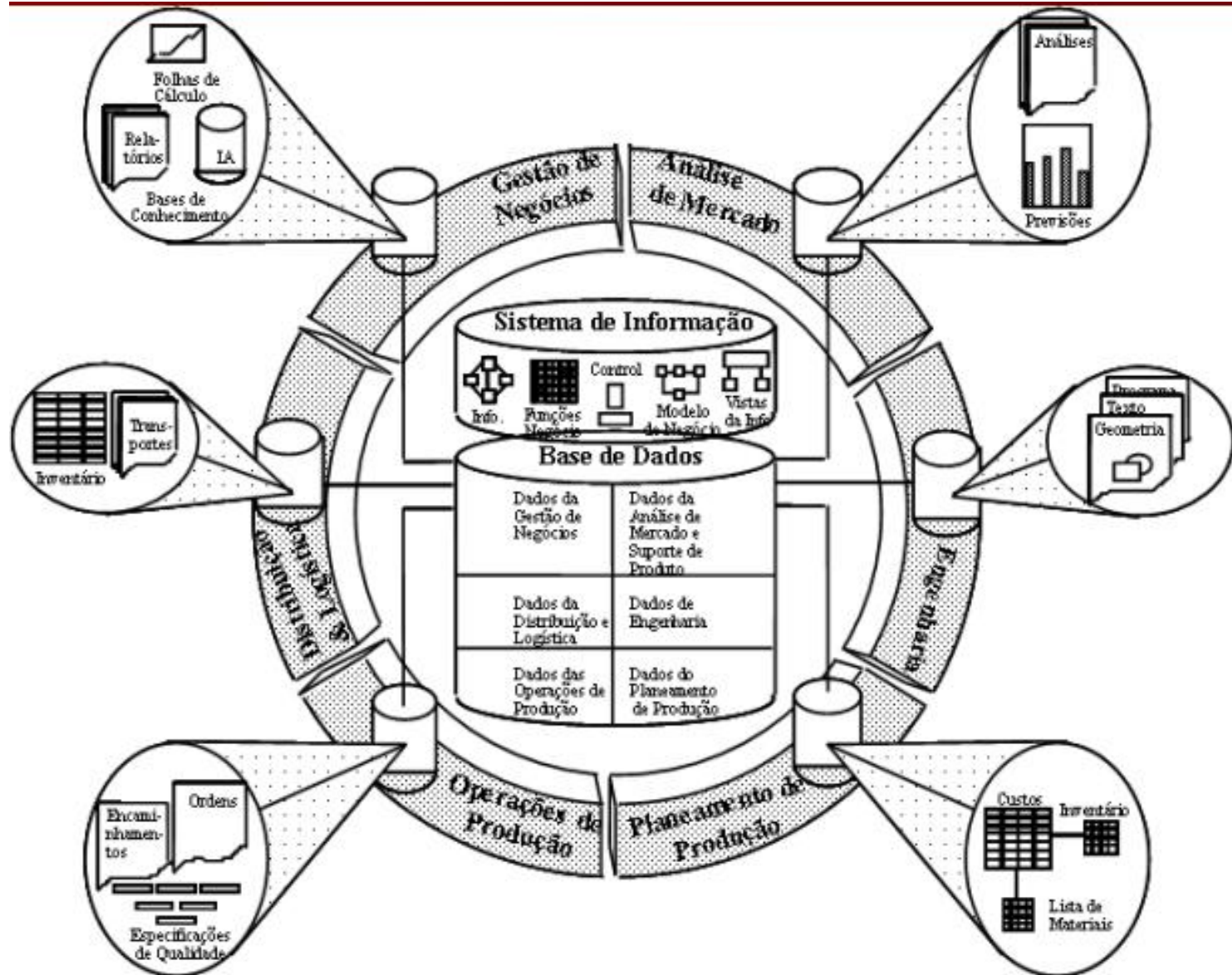
- Delegar a gestão dos acessos aos dados;
- Delegar a responsabilidade de armazenamento dos dados;
- Delegar (alguma!) a responsabilidade de consistência dos dados.



Para

- **Aumentar a produtividade no desenvolvimento das aplicações;**
- **Melhorar a manutenção de aplicações;**
- **Gerir melhor os acessos aos dados e a sua coerência;**
- **Permitir um melhor acesso aos dados das organizações.**

Visão geral de um sistema de informação



Dados vs informação vs conhecimento



Evolução dos SGBDs

- 1ª Geração – Modelos Hierárquicos e em Rede
 - Evolução dos sistemas de ficheiros
- 2ª Geração – Modelo Relacional
 - Evolui de um conceito teórico – Teoria dos conjuntos
- 3ª Geração – Novos Modelo
 - Evolução dos SGBDs de 2ª geração, para suportar os novos requisitos aplicacionais
 - Nomeadamente Bases de dados Object-Relacional e Object-oriented

Modelo Hierárquico

1960s

- Os primeiros SGBDs comerciais aparecem no mercado, através da IBM, com o seu sistema IMS e a linguagem DL/1
- Um sistema hierárquico armazena a informação como uma colecção de árvores onde cada registo tem uma e uma só raiz
- Existem dois tipos de dados: registos que contêm dados e registo que contêm associações pai-fillho, de 1-N
- O acesso aos dados é feito percorrendo a árvore desde a raiz até ao registo pretendido. Esta navegação é da responsabilidade do programador. Assim, o programador tem de conhecer a estrutura física dos dados
- Um dos principais problemas desta representação é a dificuldade de armazenar dados não hierárquicos.

Modelo em rede

1960s

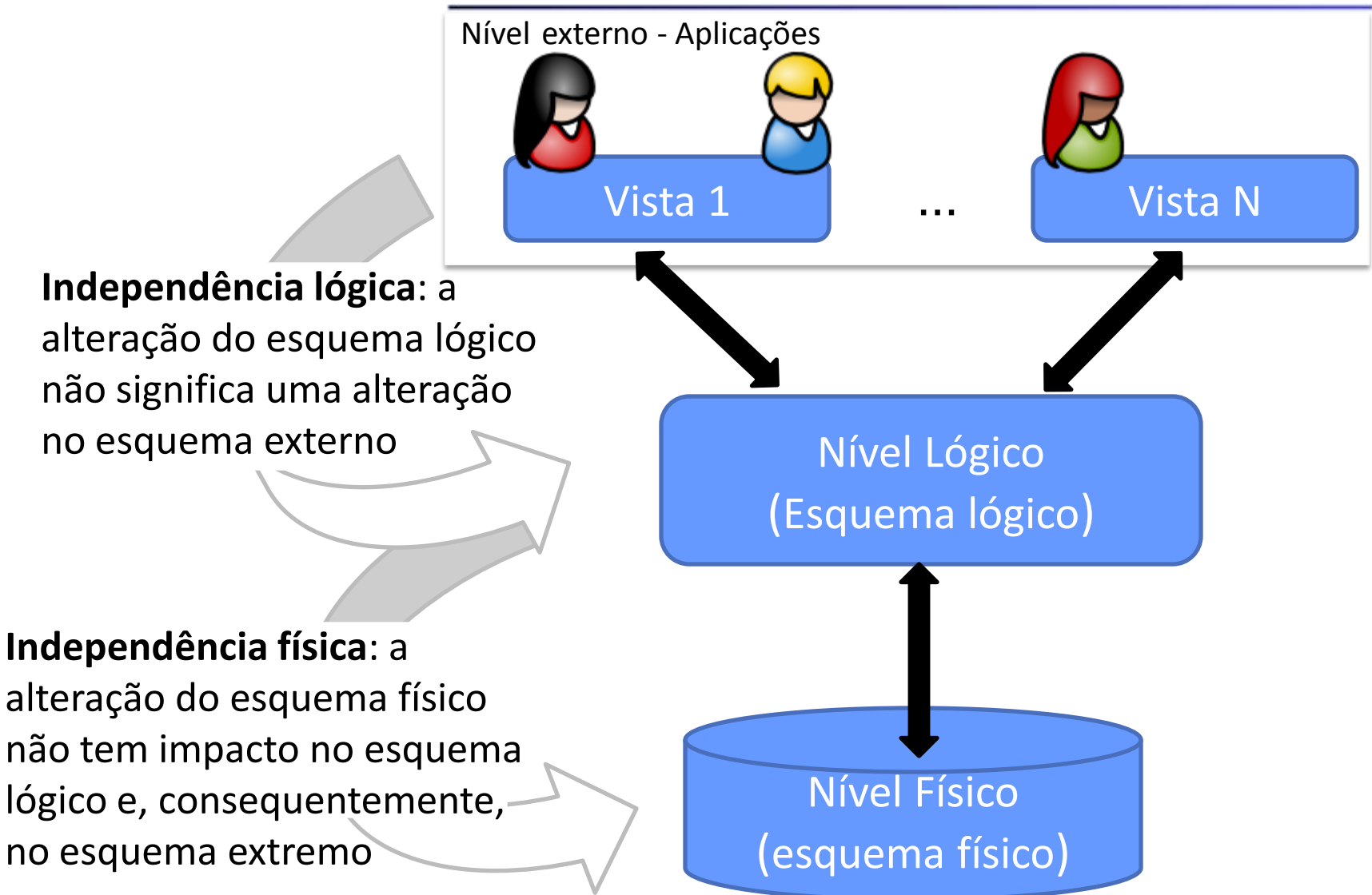
- É uma extensão do modelo hierárquico, onde a representação deixa de ser feita em árvore e passa a ser feitas através de grafos
- É suportado por uma estrutura base – o Conjunto, que não é mais do que um contentor de apontadores para outros conjuntos ou para registos de tipos simples (inteiros, floats, strings e caracteres)
- No entanto, alguns dos problemas subsistiam, nomeadamente a necessidade da navegação se ser feita de forma imperativa e que obrigava o programador a conhecer a estrutura física do dados
- C.W. Bachman desenvolveu o primeiro sistema comercial que utilizava um modelo em rede, o IDS

Sistema de Gestão de Informação

1970s

- Inicia-se uma nova era para os SGBDs
- Hoje sabemos que um “bom” sistema de gestão de informação deve:
 - Permitir um bom grau de **independência** da informação
 - **Reduzir** a **redundância** da informação armazenada
 - Garantir a **integridade** da informação e evitar a sua inconsistência
 - Facilitar a **partilha** de informação pelos vários utilizadores
 - Possibilitar uma forma uniforme de garantir a **segurança** e a privacidade da informação
 - **Facilitar** a definição da estrutura da informação e do seu **manuseamento**

Arquitetura a três níveis



Notas prévias - O Particular e o Geral

Particular

Colias philodice

Plutão

Audi A4

José

Euchloe olympia

Terra

Fiat Punto

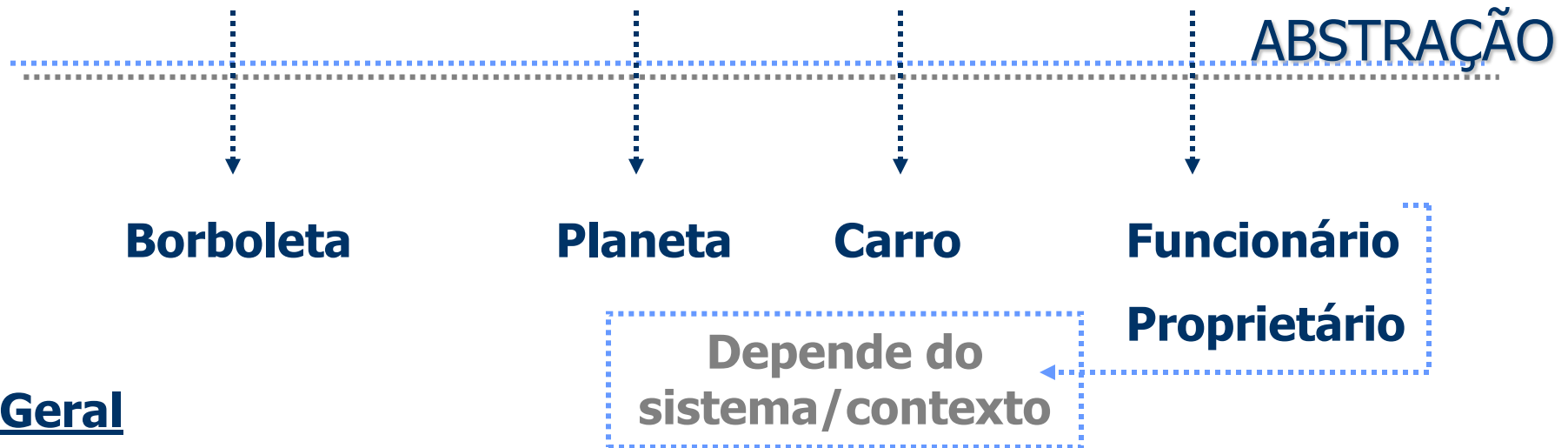
Manuel

Calephelis virginensis

Venús

BMW 530d

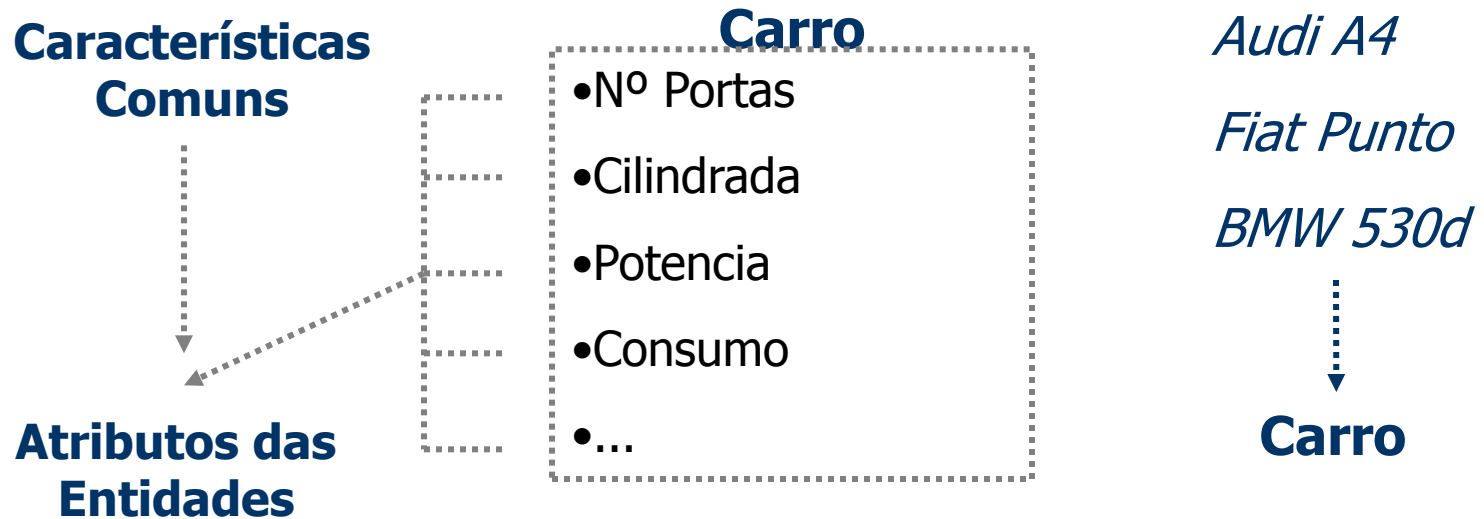
Ana



Geral

Notas prévias - Entidade

- Concretização da abstracção de um conjunto de conceitos/objectos que partilham um conjunto de características



Notas prévias – Entidade (cont.)

- Dependendo do problema, as entidades podem ter diferentes atributos (apenas aqueles que são relevantes)

Carro

- Matricula
- Cilindrada
- Cor
- Proprietário
- ...

Para a DGV

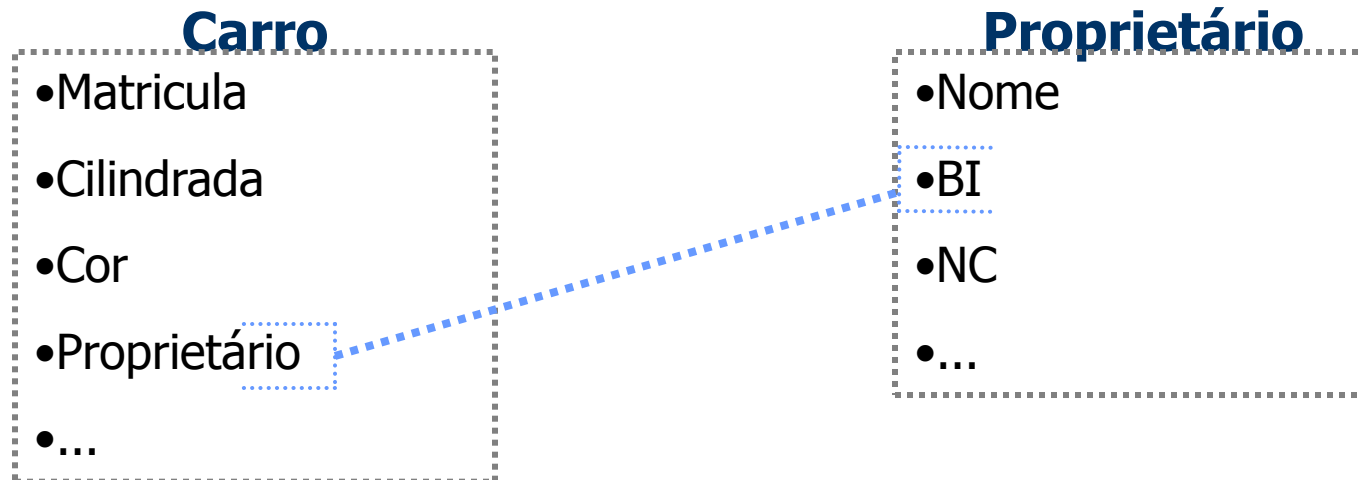
Carro

- Cilindrada
- Potencia
- Consumo
- Marca
- ...

Para uma revista auto

Notas prévias - Associação

- As entidades podem estar associadas entre si



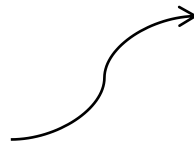
Modelo Relacional I

- Proposto por E.F. Codd em 1970
- Forte fundamento teórico – Teoria dos conjuntos
- Baseia-se na utilização de conceitos abstractos – **Relações** – e nas manipulações entre estes
- Define as operações que podem ser efectuadas sobre relações
- Modelo abstracto dos dados independente do modelo físico

Elementos principais do modelo relacional

- Os principais elementos do modelo relacional são
 - Atributo
 - Domínio
 - Esquema de relação
 - Relação
- Para exemplificar estes conceitos, considere-se a seguinte informação sobre Veículos, representada na forma tabular

Relação



Matricula	Cilindrada	Cor	Proprietario
10-10-AA	1493	Preto	1
20-55-TQ	1980	Branco	2
XX-34-01	1190	Cinzeno	3

Domínio

- O domínio é o conjunto possível de valores que cada atributo pode tomar
- O Domínio tem não só associado um conjunto de valores possíveis mas também o tipo desses valores (String, Inteiro, etc.)

Para o exemplo

- Conjunto de todas as matrículas (alfanumérico)
- Conjunto de todas as cilindradas (inteiro)
- Conjunto de todas as cores (cadeia de caracteres)
- Conjunto de todos os códigos de proprietários

Matricula	Cilindrada	Cor	Proprietario
10-10-AA	1493	Preto	1
20-55-TQ	1980	Branco	2
XX-34-01	1190	Cinzeno	3

Valores possíveis: [Branco, Preto, Cinzeno]

←..... Domínio

Domínio (cont.)

- Cada um dos valores do domínio é atómico, i.e. indivisível;
- A indivisibilidade de cada atributo depende do contexto do problema que se quer tratar. No entanto, um atributo é sempre constituído por um elemento atómico dentro do domínio em causa;
- Existe um valor comum, possível em todos os domínios – NULL – que indica a ausência de valor.

Matricula	Cilindrada	Cor	Proprietario
10-10-AA, 11-10-AA	1493	Preto	1
20-55-TQ	1980	Branco	2
XX-34-01	1190	cinzento	3

← **Errado**

Correcto

Matricula	Cilindrada	Cor	Proprietario
10-10-AA	1493	Preto	1
11-10-AA	1493	Preto	1
20-55-TQ	1980	Branco	2
XX-34-01	1190	Cinzento	3

Esquema de Relação

- Um esquema de relação $R(A_1, A_2, \dots, A_n)$ é constituído por:
 - O nome da relação: R
 - Uma lista ordenada de atributos: A_1, A_2, \dots, A_n
- Cada atributo A_i representa o papel que determinado domínio tem na relação
- Um esquema de relação é utilizado para **descrever** uma **Relação**

Exemplos:

CARRO(Matricula, Cilindrada, Cor, Proprietario)

PROPRIETARIO(Nome, BI, NC)

Relação

- Uma relação r de um esquema $R(A_1, A_2, \dots, A_n)$ é denotada da forma $r(R)$ e é constituída por:
 - um conjunto de tuplos onde $r = \{t_1, t_2, \dots, t_n\}$, onde cada tuplo t_i é uma lista ordenada $\langle v_1, v_2, \dots, v_n \rangle$
 - Cada valor v_i , com $1 \leq i \leq n$, tem um valor que
$$\begin{cases} v_i \in \text{Domínio}(A_i) \\ v_i = \text{NULL} \end{cases}$$
 - n é a cardinalidade da relação (número de tuplos da relação)

A relação $r(\text{CARRO})$ consiste no conjunto:

$r = \{ \langle "10-10-AA", 1493, "Preto", 1 \rangle;$
 $\langle "20-55-TQ", 1980, "Branco", 2 \rangle;$
 $\langle "10-10-AA", 1190, "Cinzento", 3 \rangle \}$

Matricula	Cilindrada	Cor	Proprietario
10-10-AA	1493	Preto	1
20-55-TQ	1980	Branco	2
XX-34-01	1190	Cinzento	3

Tuplo

Relação e tuplos

- Como uma relação é um conjunto de tuplos, pela própria definição de conjunto, a sua ordem não é relevante

Matricula	Cilindrada	Cor	Proprietario
10-10-AA	1493	Preto	1
20-55-TQ	1980	Branco	2
XX-34-01	1190	Cinzento	3

Matricula	Cilindrada	Cor	Proprietario
XX-34-01	1190	Cinzento	3
10-10-AA	1493	Preto	1
20-55-TQ	1980	Branco	2

São equivalentes

Grau e Cardinalidade

- Designa-se de grau o número de atributos que um esquema de relação possui
- Designa-se de cardinalidade o número de tuplos que uma relação contém

Grau do esquema de relação $\text{CARRO} = 4$

Cardinalidade da relação $r(\text{CARRO}) = 3$

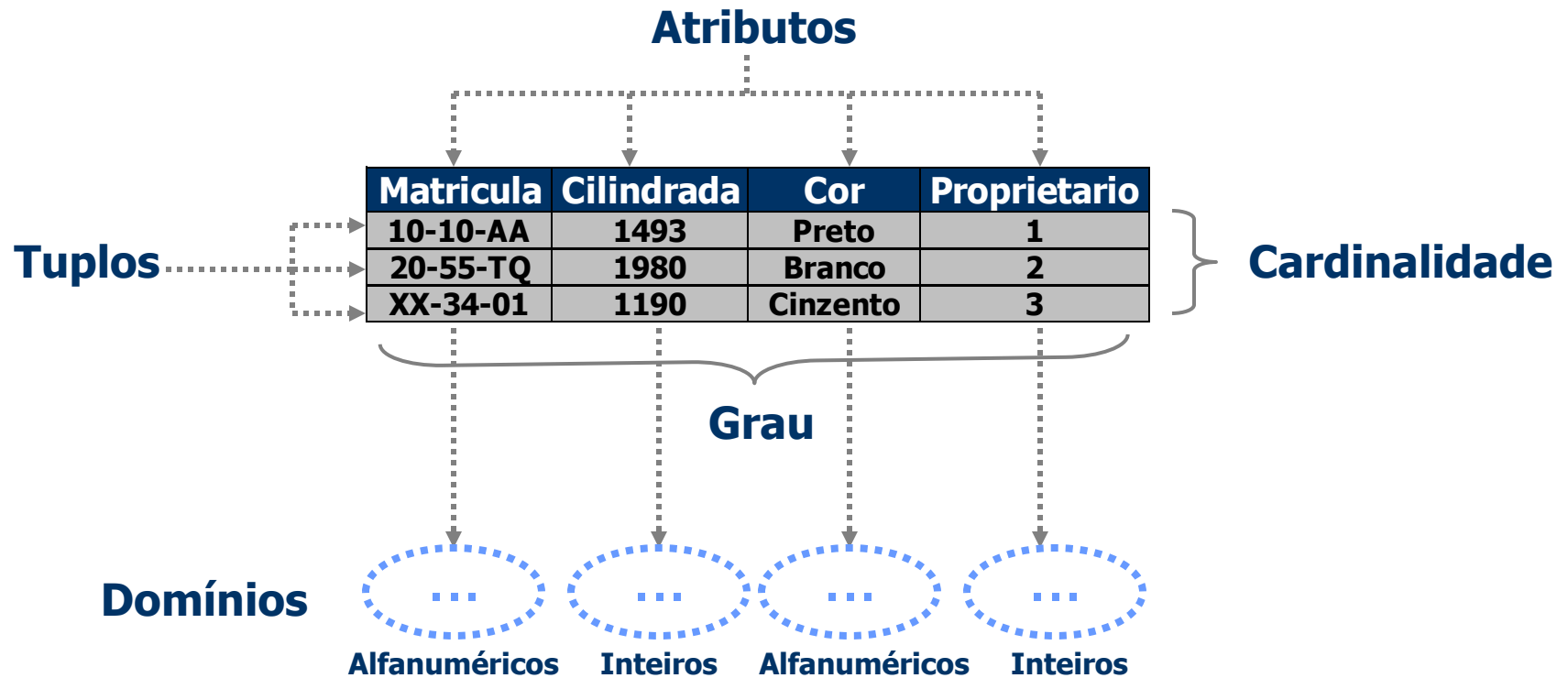
Grau

CARRO(Matricula, Cilindrada, Cor, Proprietario)

Matricula	Cilindrada	Cor	Proprietario
10-10-AA	1493	Preto	1
20-55-TQ	1980	Branco	2
XX-34-01	1190	Cinzeno	3

Cardinalidade

Resumo



Superchave

- Num conjunto todos os elementos são distintos uns dos outros
- Uma relação é um conjunto constituído por tuplos
- Logo, os tuplos têm de ser distintos uns dos outros

SUPERCHAVE

Associação de um ou mais atributos, que juntos, identificam univocamente um tuplo na relação

- No limite, a associação de todos os atributos de uma relação constituem uma superchave
- A superchave pode ter atributos redundantes

Chave

Superchave do esquema CARRO =

{Matricula, Cilindrada, Cor, Proprietario}

Superchave do esquema PROPRIETARIO =

{Nome, BI, NC}

CHAVE

Retirando os atributos redundantes da superchave temos aquilo a que se designa chave de um esquema de relação

Chave esquema CARRO = **{Matricula}**

Chave do esquema PROPRIETARIO = **{BI},{NC}**



As chaves dependem fortemente do domínio de aplicação

Chave Candidata

Considerando o esquema CARRO definido da seguinte forma:

CARRO(Matricula, NumeroMotor, Cilindrada, Cor, Proprietario)

E sabendo que cada para cada carro

- Existe uma e só uma matricula
- Existe um e só um número de motor

Existem duas chaves candidatas:

•{*Matricula*}

•{*NumeroMotor*}

Ambas identificam
univocamente um carro

CHAVE CANDIDATA

Cada uma das chaves de um esquema de relação é uma chave candidata

Chave primária

- Consiste na chave escolhida entres as chaves candidatas e que passará a identificar os tuplos da relação
- Quando existem mais que uma chave candidata, é escolhida uma delas recorrendo a algumas regras práticas, sendo escolhida a que:
 - Tiver um maior significado na sistema
 - Tiver menos atributos

No esquema CARRO

Se fosse para ser aplicado a um sistema gestão de um parque automóvel, a escolha da chave primária recaia sobre o atributo

Matricula



A escolha da chave primária depende fortemente do domínio de aplicação, mas também de outros requisitos, nomeadamente de desempenho

Chaves – convenção para representação

- No esquema de relação, é prática comum sublinhar os atributos que pertencem à chave primária

Exemplos

CARRO(Matricula, NumeroMotor, Cilindrada, Cor, Proprietario)

PROPRIETARIO(Nome, BI, NC)


- Pode também colocar-se o sufixo [AK] nas chaves candidatas, exceptuando a chave primária

Exemplos

CARRO(Matricula, NumeroMotor[AK], Cilindrada, Cor, Proprietario)

PROPRIETARIO(Nome , BI ,NC[AK])

Restrições de integridade

- Consistem em condições impostas ao Esquema Relacional
 - Restringem os dados que podem existir nas instâncias da base de Dados
 - São especificadas quando o esquema relacional é definido
 - São verificadas sempre que qualquer relação é alterada
 - **Integridade de Entidades**
 - **Integridade Referencial**
 - **Integridade de Domínio**
 - Integridade de Coluna
 - Integridade de Valor não nulo
 - Integridade de Utilizador
- 
- Restrições a considerar**

Integridade de Entidade

- A restrição de **integridade de entidade** garante que a relação de integridade de entidade possui uma chave única composta por um ou mais atributos e que nenhum atributo pertencente à chave tem o valor NULL
- O SGBD deve garantir sempre a Integridade de Entidade, verificando sempre que é adicionado um novo tuplo se os valores correspondentes à chave primária e aos atributos que foram declarados como únicos (outras chaves candidatas) são de factos únicos diferentes de NULL (*Primary Key* e *Unique*)

Exemplos que violam a integridade de Entidade

Nome	BI	NC
José	123456789	987654321
Mário	123456789	123654778

Nome	BI	NC
José	123456789	987654321
Mário	223344556	987654321

← →
Tuplos a inserir...

Integridade Referencial

- É uma restrição que envolve duas relações associadas;
- É usada para manter coerência entre os valores dessas relações;
- Informalmente, a condição imposta pela restrição de integridade referencial é: **tuplo numa relação apenas pode “referir” outro tuplo que realmente já exista noutra relação**

Matricula	Cilindrada	Cor	Proprietario
10-10-AA	1493	Preto	123456789
20-55-TQ	1980	Branco	223344556
XX-34-01	1190	Cinzentos	123456789

Nome	BI	NC
José	123456789	987654321
Mário	223344556	123654778

CARRO(Matricula, NumeroMotor, Cilindrada, Cor, Proprietario)

PROPRIETARIO(Nome, BI, NC)

Integridade Referencial (cont.)

- O que acontece aos tuplos de **r(CARRO)** se um dos tuplos de **r(PROPRIETÁRIO)** for apagado?
- Existem três possibilidades:
 1. Apagar automaticamente os tuplos “correspondentes” em **r(CARRO)**
 2. Inserir NULL nos campos de **r(CARRO)** que “referem” as ocorrências de tuplos de **r(PROPRIETARIO)**
 3. Não permitir apagar qualquer tuplo de **r(PROPRIETARIO)** enquanto os tuplos correspondentes em **r(CARRO)** não forem apagados

Chave Estrangeira

- A definição mais formal da restrição de integridade referencial leva ao conceito de Chave Estrangeira;
- Uma chave estrangeira **especifica** uma restrição de integridade entre dois esquemas de relação, R_1 e R_2 ;
- Assim, um conjunto de atributos FK no esquema de relação R_1 é uma chave estrangeira de R_2 que referencia R_2 se forem garantidas as duas condições:
 1. Os atributos de FK têm o **mesmo domínio** que os da chave primária PK do esquema de relação R_2
 2. O valor de FK num tuplo t_1 de R_1 , ou tem nos atributos de FK o valor de PK para algum tuplo t_2 de R_2 ou é NULL
- **Note-se que uma Chave Estrangeira pode referir o seu próprio esquema de relação.**

Integridade Domínio

- Na integridade de domínio é imposto que cada atributo seja atômico, no domínio desse atributo
- Os valores dependem do tipo de dados que estão associados aos domínios
- A definição do domínio engloba, normalmente, a definição de um nome, do seu tipo, da sua dimensão, dos valores e gama permitidos;

- **Numéricos** (Decimal, Real, etc...)
- **Cadeias de caracteres** (char ,varchar)
- **Cadeia de bits** (binary, varbinary)
- **Data/Hora** (Datetime)

Alguns exemplos de tipos de dados

Integridade de Coluna

- A integridade de coluna é um refinamento da integridade de domínio
- Define-se a gama admissível de valores para as colunas alvo, a fim de se eliminar a inserção de tuplos inválidos para o domínio do problema

Do exemplo anterior

CARRO(Matricula, NumeroMotor, Cilindrada, Cor, Proprietario)

- Os atributos cilindrada e matrícula têm as seguintes restrições de integridade de coluna:
 - **Cilindrada** – Valor inteiro positivo
 - **Matrícula** – cadeia de caracteres com um dos seguintes padrões: **DD-DD-LL**, **DD-LL-DD** ou **LL-DD-DD**

Outros tipos de integridade

Integridade de de valor não nulo

- Elimina a hipótese de o valor nulo ser atribuído a determinados atributos

Integridade de Utilizador

- A integridade de utilizador é uma qualquer regra que, não estando englobada nas anteriores, é necessária para manter a coerência do sistema
- São restrições que saiem dos mecanismos definidos no modelo relacional e que são implementados pelos SGBD

Exemplos:

- Um carro não pode mudar de proprietário
- Um proprietário não pode ter mais que 3 carros



Estas integridades estão sempre relacionadas com o sistema em causa

Exemplo

(1ª parte) Fornecedores de filmes de video

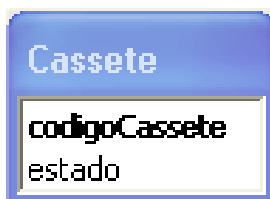
Definição do sistema

- O sistema permitirá a uma empresa de aluguer de vídeos, gerir os alugueres das sua cassetes
- Sobre os filmes que a empresa possui, pretende-se que seja possível saber:
 - o código do filme (único para cada um)
 - o título do filme
 - o ano de lançamento
- Sobre as cassetes que possui, deve ser possível saber:
 - o código da cassette (único para cada uma)
 - indicação do filme contido na cassette
 - o estado da cassette (disponível, alugado, perdido, danificado)

Exemplo (cont.)

Entidades

- Filme
- Cassete



Atributos

- Código de filme, titulo, ano lançamento
- Código da cassete, estado

Associações

- Para se saber qual o filme que cada cassete tem gravado é necessário existir um associação entre os esquemas de relação Filme e Cassete

Como se representa essa associação?

Exemplo (cont.)

Associação

- Por causa da associação entre CASSETE e FILME, houve a necessidade de incluir um atributo extra no esquema de relação CASSETE

Esquemas de Relação

FILME(codigoFilme, titulo, anoLancamento)

CASSETE(codigoCassete, estado, filme [FK])

Chave estrangeira

- Existe uma chave estrangeira em CASSETE que referencia a chave primária de FILME. Através dessa chave é garantida a restrição de Integridade Referencial

Exemplo (cont.)

Tipos dos Atributos

- codigoFilme – carácter(5)
- titulo – carácter(255)
- anoLancamento – inteiro
- codigoCassete – carácter(5)
- estado – carácter(10)

Outras restrições

- O atributo anoLancamento tem de ser superior a 1900
- O atributo estado só pode tomar um dos seguintes valores:
 - alugado, disponível, perdido, estragado

Exemplo (cont.)

r(FILME)

codigoFilme	titulo	anoLancamento
F0001	Mad Max	2000
F0002	007 - Goldeneye	1999
F0003	Tubarão	2001

r(CASSETTE)

codigoCassete	estado	filme
C0001	alugado	F0001
C0002	disponivel	F0001
C0003	estragado	F0002
C0004	disponivel	F0003
C0005	disponivel	F0001

Exemplo (cont.)

- Quais os códigos das cassetes que contêm o filme Mad Max e se encontram disponíveis?

codigoCassete	
C0002	
C0005	

Como resultado obtêm-se uma nova relação



Como se determinam estes códigos?

Exemplo (cont.)

- Existem vários caminhos possíveis...
- Um deles passa por efectuar uma **selecção** sobre a relação r(FILME) para escolher o tuplo que representa o filme desejado

codigoFilme	titulo	anoLancamento
F0001	Mad Max	2000
F0002	007 - Goldeneye	1999
F0003	Tubarão	2001

codigoFilme	titulo	anoLancamento
F0001	Mad Max	2000



Exemplo (cont.)

- De seguida efectua-se um **produto cartesiano** entre a relação resultante da selecção e r(CASSETE)

codigoFilme	titulo	anoLancamento	codigoCassete	estado	filme
F0001	Mad Max	2000	C0001	alugado	F0001
F0001	Mad Max	2000	C0002	disponivel	F0001
F0001	Mad Max	2000	C0005	disponivel	F0001
F0001	Mad Max	2000	C0003	estragado	F0002
F0001	Mad Max	2000	C0004	disponivel	F0003

- Segue-se uma nova **selecção** sobre o atributo estado, só se considerando os tuplos que contenham o valor 'disponível' e verifiquem a condição 'códigoFilme==filme'

codigoFilme	titulo	anoLancamento	codigoCassete	estado	filme
F0001	Mad Max	2000	C0002	disponivel	F0001
F0001	Mad Max	2000	C0005	disponivel	F0001

- Finalmente é realizada uma operação que **projecta** as colunas pretendidas

codigoCassete
C0002
C0005

Exemplo (cont.)

- Resumindo

codigoFilme	titulo	anoLancamento
F0001	Mad Max	2000
F0002	007 - Goldeneye	1999
F0003	Tubarão	2001

1º

codigoCassete	estado	filme
C0001	alugado	F0001
C0002	disponivel	F0001
C0003	estragado	F0002
C0004	disponivel	F0003
C0005	disponivel	F0001

2º

codigoCassete	estado	filme
C0001	alugado	F0001
C0002	disponivel	F0001
C0005	disponivel	F0001

3º

codigoCassete
C0002
C0005

4º

Exemplo (cont.)

Exemplo (2ª parte)

Pretende-se adicionar funcionalidades ao modelo de dados existente, para que possa fornecer :

- Informação sobre os fornecedores da empresa. Essa informação inclui:
 - o código do fornecedor (único)
 - o número de contribuinte
 - o nome da empresa do fornecedor
- Informação sobre que filmes fornecidos por cada fornecedor

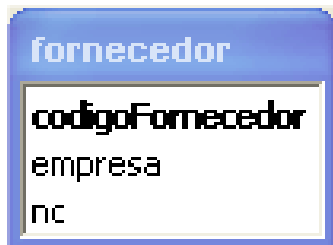
Atributos

- Código fornecedor, Empresa, número de contribuinte

Associações

- Para se saber qual o filme que cada fornecedor fornece é necessário existir uma associação entre os esquemas de relação FILME e FORNECEDOR

Como?



Exemplo (cont.)

Associação

- Por causa da associação entre Fornecedor e Filme, houve a necessidade de criar um esquema de relação adicional (associação M-N)

Esquemas de Relação

FORNECEDOR(codigoFornecedor, empresa, nc)

FORN_FILME(codigoFilme, codigoFornecedor)



A chave do esquema de relação associativo FORN_FILME é composta pelas chaves primárias das tabelas que estão associadas

- No esquema de relação FORNECEDOR o atributo nc tem de ser único. Por essa razão existem duas chaves candidatas: codigoFornecedor e nc.
- Dessas, uma é eleita para chave primária

Exemplo (cont.)

FORNECEDOR

codigoFornecedor	Empresa	NC
FN001	Filmes LDA	123456789
FN002	Só Filmes	987654321

FORN_FILME

codigoFornecedor	codigoFilme
FN001	F0001
FN001	F0002
FN002	F0003

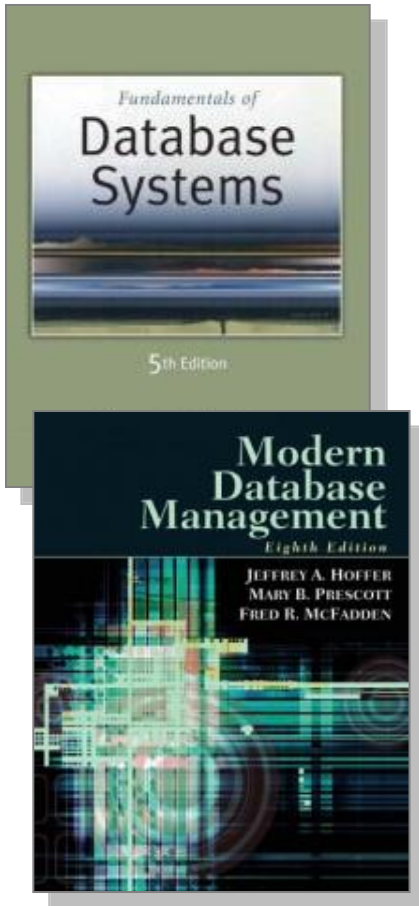
FILME

codigoFilme	titulo	anoLancamento
F0001	Mad Max	2000
F0002	007 - Goldeneye	1999
F0003	Tubarão	2001

Bibliografia

Folhas da unidade curricular de Introdução aos Sistemas de Informação

Prof. Walter Vieira, ISEL;



Fundamentals of Database System (5th Edition)

R. Elmasri, Shamkant B. Navathe

Addison Wesley, 2003

Modern Database Management, (8th Edition)

Jeffrey A. Hoffer, , Mary Prescott, Fred McFadden

Prentice Hall, 2006