

- **SGBD**

## Sistema de Gestão de Base de Dados



- Permite guardar dados e informação dos mais variados tipos;
- Permite um rápido e fácil acesso aos dados;
- Acelera os processos de manuseamento da informação, como, por exemplo, consultas ou alterações de dados;
- Pode economizar muito papel e espaço para o armazenar.

- **Base de Dados**

Conjunto integrado de dados relacionados a um tópico ou propósito específico, que serão partilhados e utilizados concorrentemente para múltiplos objectivos por múltiplos tipos de utilizadores.

- **SGBD**

Genericamente um SGBD (**Sistema de Gestão de Base de Dados**) permite recolher e controlar dados sobre um qualquer tema ou assunto, como por exemplo, itens de um inventário. Assim, podemos armazenar, recuperar e manipular dados ou informação.

- **MS Access**

É um sistema de gestão de base de dados relacional (SGBD)

Como sabemos os **dados** por si só não têm qualquer valor a não ser que estejam relacionados de forma a obtermos informação. Deste modo, existem alguns requisitos ou condições que estes devem garantir, nomeadamente:

- **Actualidade;**
- **Correcção;**
- **Relevância;**
- **Disponibilidade;**
- **Legibilidade.**

## Do SGBD

- **Independência dos dados;** relativamente aos programas e procedimentos
- **Redundância controlada;** excepcionalmente para acesso mais rápido
- **Integridade dos dados;** assegurada por procedimentos de validação
- **Abstracção dos dados;** por parte do utilizador relativamente ao armazenamento
- **Acesso simultâneo aos dados;** vários utilizadores acedem simultaneamente
- **Facilidade de obtenção de informação actualizada;**
- **Diferentes vistas da base de dados;** para diferentes utilizadores

## 1. Baseados em objectos (Modelo conceptual);

Pretendem representar a realidade através de objectos. Por outro lado, entenda-se objectos como entidades que tanto podem caracterizar pessoas, organizações, coisas, etc.

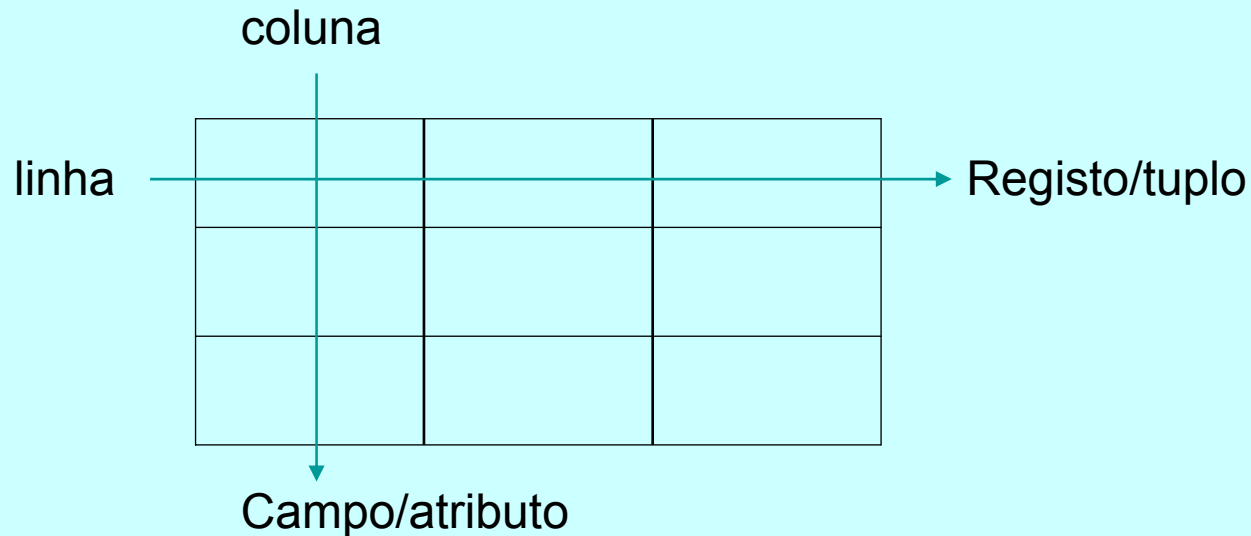
## 2. Baseados em registos (Modelo de implementação);

Procuram representar a realidade através de registos onde a informação está estruturada sob a forma de campos. Dentro deste modelo temos vários submodelos:

- Hierárquico.....**colecção de registos relacionados através de ligações.**
- Rede.....**idêntico ao anterior mas a estrutura em vez de hierárquica é em forma de rede.**
- Relacional.....**constituído por relações, onde cada relação é uma tabela.**

- **Elemento fundamental**

Uma relação é uma estrutura fundamental, representada por uma tabela organizada em linhas e colunas. Os atributos traduzem o tipo de dados a armazenar e a cada atributo está associado um domínio ou gama de valores possíveis.



- **Propriedades das tabelas**

- Não pode haver duas colunas com o mesmo nome; cada coluna deve ser identificada de um modo exclusivo;
- Não deve haver campos vazios;
- O domínio de cada atributo deve ser formado por valores atômicos, ou seja, não é permitido incluir mais do que um valor em cada campo por registo;
- Cada linha da tabela representa uma entidade ou ocorrência única, deste modo, não deve haver registos duplicados.

- As **Tabelas** são constituídas por **Registos**, que contêm todos os dados acerca de um determinado assunto.

Os Registos, são por sua vez, formados por **Campos**.

The diagram illustrates a table structure. At the top, a large box labeled "Tabela: agenda" contains a header row with five columns: "Nome", "Morada", "Localidade", "Cod. postal", and "Telefone". Below this header are two rows of data. The first row contains "1", "João Maria", "Travessa das Flores", "Funchal", "2345 Funchal", and "345142514". The second row contains "2", "Abel Fonseca", "Rua do Vinho", "Azeitão", "4586 Azeitão", and "245689531". To the left of the table, the text "Nome do campo" is connected to the header row by an arrow. Below it, "registos" is connected to the data rows by a bracket. At the bottom, a horizontal line with vertical tick marks is labeled "campos", indicating the field structure.

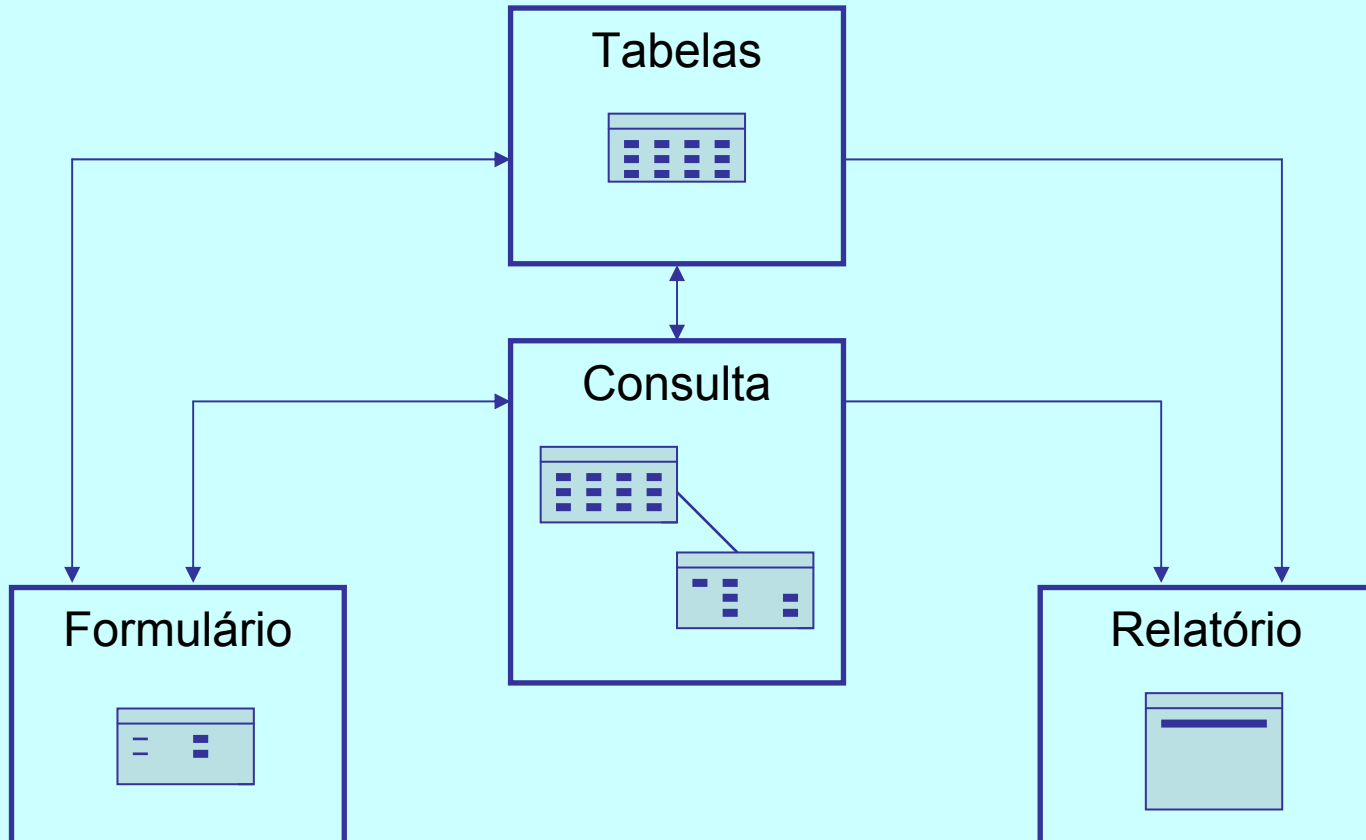
| Tabela: agenda |              |                     |            |              |           |
|----------------|--------------|---------------------|------------|--------------|-----------|
|                | Nome         | Morada              | Localidade | Cod. postal  | Telefone  |
| 1              | João Maria   | Travessa das Flores | Funchal    | 2345 Funchal | 345142514 |
| 2              | Abel Fonseca | Rua do Vinho        | Azeitão    | 4586 Azeitão | 245689531 |

Nome do campo →

registos {

campos

- **Tabelas** (tables)  
Servem para armazenar dados numa estrutura.
- **Inquéritos/Consultas** (querys)  
Servem para localizar e reunir dados armazenados dentro de um certo critério de procura.
- **Formulários** (forms)  
Servem para introduzir, visualizar, modificar ou apagar dados numa certa estrutura de apresentação.
- **Relatórios** (reports)  
Servem para visualizar e compor dados no ecrã ou no papel.



## 1ª Tarefa

Criar uma base de dados elementar, por exemplo, uma agenda.

### Objectivos

- Elaboração de um esboço de uma base de dados
- Contacto com o Access
- Criação de uma tabela
- Introdução dos campos
- Escolha do tipo de dados mais apropriados para os campos
- Definição das propriedades para os campos
- Introdução de dados

## Escolha de tipo de dados

Quando criamos um campo, por defeito, ele é assumido como texto

Os tipos de dados determinam que géneros de valores são possíveis de armazenar num determinado campo. Por exemplo, se o tipo de dados for número, o campo só vai permitir que os dados armazenados sejam numéricos.

A escolha é importante porque causa da veracidade dos dados e, por outro lado, é importante evitar desperdício de espaço armazenado na base de dados.

Por exemplo, se um campo for o BI, não podemos permitir a introdução de dados que não numéricos!

**Texto**..... alfanumérico, armazena até 255 caracteres

**Memo**.....alfanumérico utilizado para comentários, armazena até 65 535 caracteres

**Número**.....numérico que podem ser utilizados em cálculos. Podem ocupar 1, 2, 4 ou 8 bytes

**Data/hora**...armazena formatos de data e hora até ao ano 9999 e ocupa 8 bytes

**Moeda**.....armazena formatos de moeda que podem ser utilizados em cálculos. Ocupa 8 bytes

**Numeração automática**.. armazena números sequenciais e não podem ser alterados

**Sim/Não**..... armazena valores booleanos, 0 ou 1, ou do tipo F ou V. Ocupa 1 byte

**Objecto OLE**.....permite incorporar ou ligar objectos criados noutros programas até 1Gb

**Hiperligação**..... alfanumérico utilizado para referenciar ligações. Até 2048 caracteres

**Assistente de pesquisa**...armazena um valor de outra tabela ou de uma lista de valores

|    | Geral                     | Pesquisa      |
|----|---------------------------|---------------|
| 1  | Tamanho do campo          | 50            |
| 2  | Formatar                  |               |
| 3  | Máscara de introdução     |               |
| 4  | Legenda                   |               |
| 5  | Valor predefinido         |               |
| 6  | Regra de validação        |               |
| 7  | Texto de validação        |               |
| 8  | Necessário                | Não           |
| 9  | Permitir comprimento zero | Não           |
| 10 | Indexado                  | Não           |
| 11 | Compressão Unicode        | Não           |
| 12 | Modo IME                  | Sem controlos |
| 13 | Modo de frase IME         | Nenhum        |
| 14 | Etiquetas inteligentes    |               |

1. Define o número máximo de um campo
2. Especifica como devem ser apresentados os campos texto, memo, número, data/hora, moeda e sim/não.
3. Especifica regras para a introdução de dados
4. Define uma rótulo usado no formulário para a introdução de dados
5. Define um valor predefinido quando introduzimos um novo registo
6. Forma de validar os registos introduzidos através de uma expressão
7. Exibe uma mensagem quando os valores introduzidos não respeitam a máscara
8. Obriga ou não a introdução de dados no campo
9. Permite registos que não tenham dados
10. Cria um índice de forma a melhorar a procura e consultas
11. Permite a compressão nos campos texto, memo e hiperligação
12. Permite a conversão de caracteres para caracteres asiáticos
13. Indica que modo de frase IME
14. Permite a introdução de etiquetas aquando da introdução de dados

## Tamanho do campo *In TIC10, Areal Editores (livro adoptado)*

| Tipo de dados Numérico | Domínio de valores   | Casas decimais | Tamanho  |
|------------------------|--|----------------|----------|
| Byte                   | 0 a 255  | Nenhuma        | 1 byte   |
| Inteiro                | -32768 a 32767   | Nenhuma        | 2 bytes  |
| Inteiro longo          | -2147483648 a 2147483647   | Nenhuma        | 4 bytes  |
| Simple                 | -3.402823E38 a 3.402823E38   | Máximo de 7    | 4 bytes  |
| Duplo                  | -1.79769313486231E308 a 1.79769313486231E308   | Máximo de 15   | 8 bytes  |
| ID de replicação       | Código utilizado na criação de uma réplica, para cópias sincronizadas da base de dados | Não disponível | 16 bytes |
| Decimal                | $-10^{28-1}$ a $10^{28-1}$   | Máximo de 28   | 12 bytes |

## Máscara de Introdução (texto e data/hora)

| Carácter      | Significado   |
|---------------|---|
| 0             | Introdução obrigatória de um dígito {0, 1, 2, 3,..., 9}.  |
| 9             | Introdução não obrigatória de um dígito ou espaço.  |
| #             | Introdução não obrigatória de um dígito ou espaço, com a possibilidade de inserir o sinal mais (+) e o sinal menos (-).   |
| L             | Introdução obrigatória de uma letra {A, B, C, ..., Z}.  |
| ?             | Introdução opcional de uma letra.   |
| A             | Introdução obrigatória de uma letra ou dígito.  |
| a             | Introdução opcional de uma letra ou dígito.   |
| &             | Introdução obrigatória de um qualquer carácter ou espaço.   |
| C             | Introdução opcional de um qualquer carácter ou espaço.  |
| .,:;- /       | Introdução de um dos seguintes separadores: casas decimais, milhares, datas ou horas.   |
| ←             | Converte todos os caracteres para minúsculas.   |
| →             | Converte todos os caracteres para maiúsculas.   |
| !             | Obriga a que os caracteres na máscara sejam introduzidos da direita para a esquerda, ao contrário da introdução normal (da esquerda para a direita). Este carácter pode ser colocado em qualquer parte da combinação de caracteres. |
| \             | Permite escrever um destes caracteres utilizados como caracteres da máscara de introdução. Isto é, a utilização da barra (\) antes do carácter anula o seu efeito.  |
| Palavra-passe | Qualquer carácter escrito será guardado como um carácter; no entanto, serão apenas visualizados asteriscos (*).   |

*In TIC10, Areal Editores (livro adoptado)*

## Regras de validação *In TIC10, Areal Editores (livro adoptado)*


### Exemplos

| Regra de validação                          | Significado  |
|---|--|
| $\neq 0$                                    | O valor introduzido tem que ser diferente ( $\neq$ ) de zero (0).  |
| $> 1000$ Ou 0                               | O valor introduzido tem que ser superior ( $>$ ) a 1000 ou igual a zero (0).                                   |
| Como "A????"                                | O valor introduzido tem de ter 4 caracteres e iniciado pela letra A.   |
| $\geq \#1/1/96\#$ E $\leftarrow \#1/1/97\#$ | A data introduzida tem que ser superior ou igual ( $\geq$ ) a 1/1/1996 e inferior ( $\leftarrow$ ) a 1/1/1997. |

| Nome   | Morada    | Local  |
|--------|-----------|--------|
| Maria  | R. Poços  | Lisboa |
| Regina | Gil Eanes | Oeiras |
| Maria  | Huelva    | Oeiras |

A tabela ao lado está construída segundo o modelo relacional. Não temos registos duplicados mas, temos dados repetidos o que não nos permite identificar de modo unívoco cada linha da tabela ou registo.

Um atributo ou conjunto de atributos designa-se por **chave** quando nos permite identificar de modo **único** cada entidade ou registo da tabela

| <u>Codpessoal</u>  | Nome   | Morada    | Local  |
|---|--------|-----------|--------|
| 1   | Maria  | R. Poços  | Lisboa |
| 2   | Regina | Gil Eanes | Oeiras |
| 3   | Maria  | Huelva    | Oeiras |


Para ultrapassar o problema anterior e podermos ter uma chave, basta acrescentar um campo que traduza **registos unívocos**.

- Cada tabela deve possuir uma chave. Pode ser constituída por apenas um atributo (**chave simples**) ou por mais do que um atributo (**chave composta ou concatenada**)
- **Chave candidata**  
são todos os atributos ou conjuntos de atributos passíveis de identificar univocamente os registos de uma tabela.
- **Chave primária**  
Entre as chaves candidatas numa tabela, uma delas será escolhida para identificar cada tuplo de forma única. A seleccionada tomará o nome de chave primária.
- **Chave estrangeira**  
É um conjunto de um ou mais atributos que são chave primária numa outra tabela.

- **Chave primária**

Entre as chaves candidatas numa tabela, uma delas será escolhida para identificar cada tuplo de forma única. A seleccionada tomará o nome de chave primária.

Na tabela em baixo a chave primária será no atributo Codpessoal porque é o único passível de identificar univocamente os registos da tabela.

| <u>Codpessoal</u>  | Nome   | Morada    | Local  |
|---|--------|-----------|--------|
| 1   | Maria  | R. Poços  | Lisboa |
| 2   | Regina | Gil Eanes | Oeiras |
| 3   | Maria  | Huelva    | Oeiras |

- **Chave estrangeira ou externa**

É um conjunto de um ou mais atributos que são chave primária numa outra tabela.

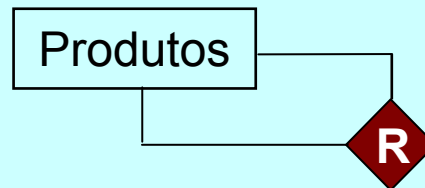


- **Características da chave primária**

- **Ser unívoca:** o(s) atributo(s) que desempenham o papel de chave primária têm que ter um valor único para cada registo;
- **Não pode conter valores nulos:** nenhum dos atributos poderá conter um valor nulo;
- **Não redundante;** no caso de ser composta, só devem ser incluídos os atributos necessários para identificar de modo unívoco os registos.

Após termos identificado as entidades que vamos representar assim como os atributos que definem cada entidade, temos, também, que definir o modo como as entidades se relacionam entre si.

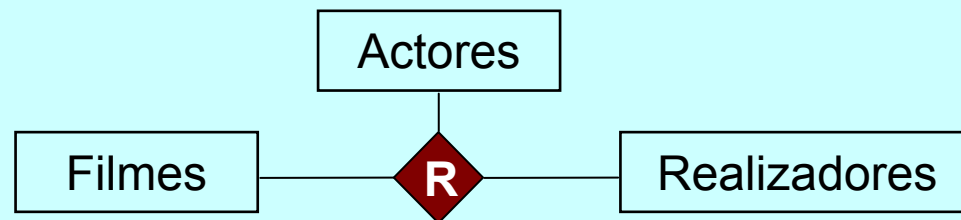
## Relações unárias



## Relações binárias (mais comuns)



## Relações ternárias



- As associações entre entidades (binárias) distinguem-se entre si pelo grau e caracterizam-se da seguinte maneira:

**- um-para-um                      1:1**

quando cada elemento de uma entidade A está associado, no máximo com um elemento da outra entidade B (aluno:turma, cidadão:BI)

**- um-para-muitos                1:N**

quando um elemento de A pode estar associado com vários elementos de B (cidade:cidadãos)

**- muitos-para-muitos        N:M**

quando a um elemento de A podem corresponder vários elementos de B e do mesmo modo, a cada elemento de B podem corresponder vários elementos de A (alunos:disciplinas – um aluno frequenta uma disciplina e uma disciplina é frequentada por vários alunos)

- A actualização de uma base de dados deve ser feita de forma a que a informação se mantenha consistente. Para tal deve ser assegurada dois tipos de integridade:

- **Integridade de entidade**

- os valores dos atributos que correspondem à chave primária de uma entidade não podem ter valores nulos.

- **Integridade referencial**

- um valor de uma chave externa tem obrigatoriamente de existir como elemento da chave primária da tabela relacionada com aquela chave externa

## Segundo as Formas Normais

**1ª FN** – Exige que todos os seus campos sejam atómicos e que não haja campos repetidos

Exemplo de uma tabela: Alunos (nºaluno, nome, morada, disciplinas)

| nºaluno | nome  | morada | disciplinas   |
|---------|-------|--------|---------------|
| 1       | Ana   | xsf    | Port.; Ingl.; |
| 2       | Carla | Xpto   | Mat.; Fis,    |
| 3       | Diogo | bns    | Port.; Fil.;  |

No campo **disciplinas** temos valores não atómicos, o que não deveria de acontecer. Logo, temos de alterar esta situação de forma a ter apenas uma disciplina só por registo.

# SGBD: Estratégias para concepção de BD TIC

| Nº aluno | nome  | morada | disciplinas |
|----------|-------|--------|-------------|
| 1        | Ana   | xf     | Port.       |
| 1        | Ana   | xf     | Ingl.       |
| 2        | Carla | Xpto   | Mat.        |
| 2        | Carla | Xpto   | Fis.        |

A modificação efectuada resolve o problema atómico dos registos mas, cria um de redundância, ou seja, ficamos com registos duplicados. Agora, cada aluno pode-se inscrever a várias disciplinas mas, só podemos diferenciar um registo de outro através dos atributos nºaluno e coddisciplina. Na tabela criada surge outro problema a que chamamos **dependência funcional** (atributos que são funcionalmente dependentes) Para solucionar esta situação temos de criar uma segunda tabela (2ªFN)



No cumprimento da **2ª FN** desagregamos a tabela inicial em duas tabelas, o que resultará no seguinte:



## Resumindo

Os campos só podem admitir valores atómicos, ou dito de outro modo, não podemos ter mais do que um valor elementar dentro de cada campo (**1ªFN**);

Cada atributo não-chave tem que ser funcionalmente dependente da chave elementar na sua totalidade e não apenas de uma parte dessa chave (**2ªFN**).

Por último, cada atributo não chave não pode depender funcionalmente de nenhum outro atributo que não seja a chave (**3ªFN**).

Para a base de dados ficar completa de forma a relacionar as tabelas, temos que criar uma terceira tabela

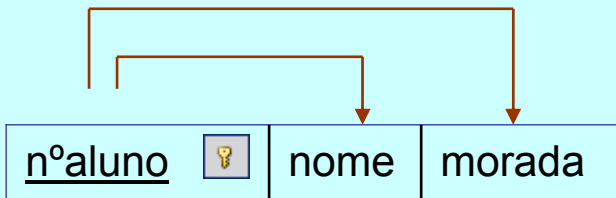


Tabela alunos

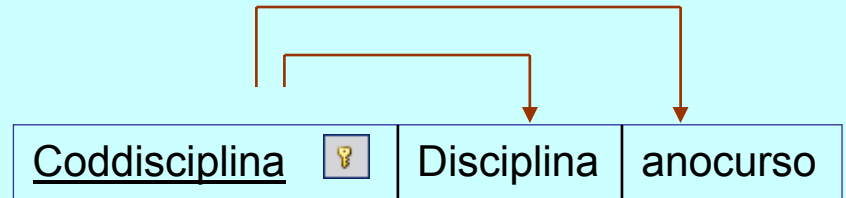


Tabela alunos\_discipl

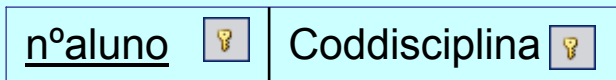


Tabela disciplina (com chaves estrangeiras)

- Estudo da situação para recolha de dados e informação relevante;
- Realização de um desenho da estrutura geral, evidenciando as entidades e fluxos de relações entre elas;
- Enumeração dos atributos e identificação dos atributos chave de cada entidade;
- Elaboração dos relacionamentos entre as várias entidades;
- Eventual revisão da estrutura de entidades, atributos e relacionamentos;
- Dedução das tabelas a partir da estrutura de entidades e relacionamentos;
- Desenvolvimento da base de dados: criação das tabelas, relacionamento entre tabelas;
- Criação dos programas de aplicação com interfaces gráficos de forma a facilitar a utilização

## 3ª e 4ª Tarefa - Criar bases de dados

### Propósitos

- Elaborar um esboço de base de dados
- Criar tabelas
- Introduzir campos
- Escolher dados mais apropriados para os campos
- Definir as propriedades para os campos
- Criar chaves e estabelecer relações entre tabelas
- Introduzir dados
- Criar formulários para introdução de dados

☺ **Consultar a sala de aula virtual**

## O que é uma consulta/query

É um objecto do MS Access e dos SGBD em geral que permite efectuar interpelações à base de dados, segundo determinados critérios e extrair a informação que os cumpra.

Os propósitos ao criar uma consulta poderão ser, nomeadamente, filtrar, analisar, actualizar, acrescentar ou eliminar dados de uma tabela. Com base em consultas podemos, também, elaborar relatórios.

## **De selecção**

Permitem obter dados de uma ou de várias tabelas que obedecem a determinados critérios. São os mais utilizados.

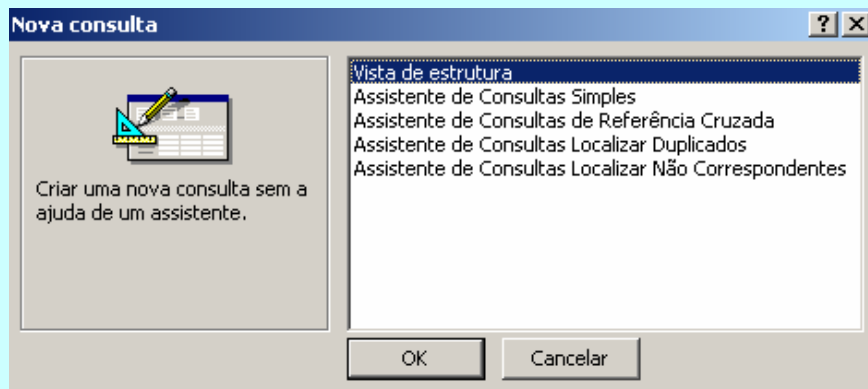
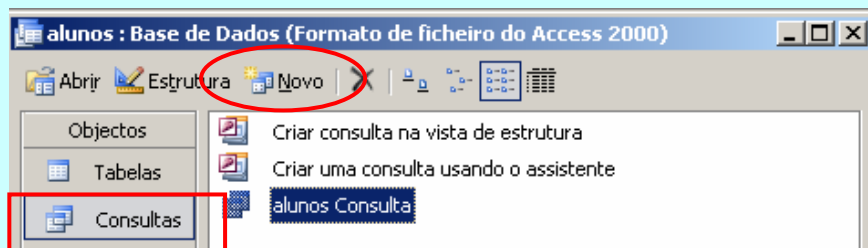
## **De sumarizar**

Possibilitam efectuar cálculos como a adição, a média, a contagem, etc. e reestruturar dados para facilitar a sua análise. São do tipo referência cruzada.

## **De acção**

Têm consequências sobre a BD. Podem ser do tipo criar tabela, actualizar dados, adicionar e eliminar registos.

Optar pelo objecto consultas e carregar no botão novo. Quando surgir outro ecrã escolher qual o tipo de consulta que se pretende efectuar.

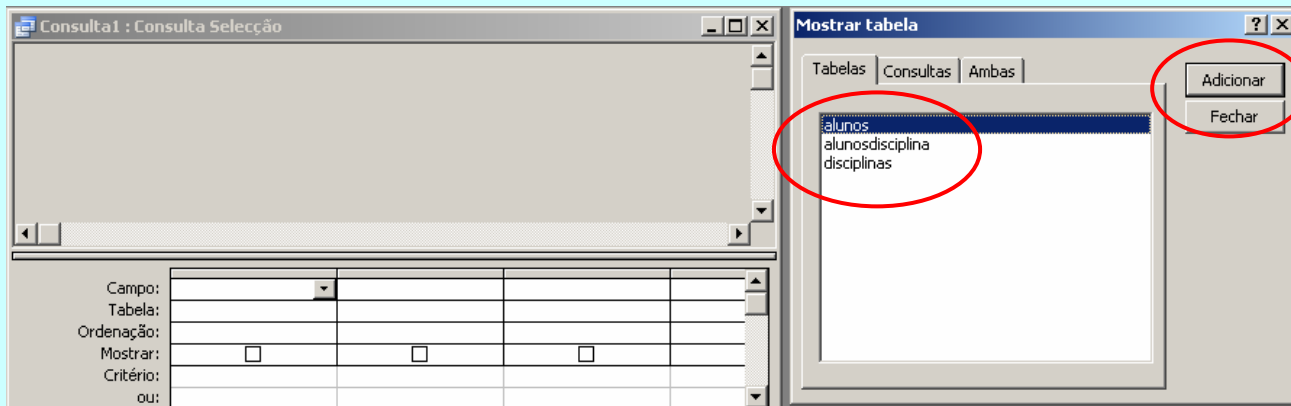


## Nota

*Reparar que no lado esquerdo deste ecrã são fornecidas algumas indicações sobre o tipo de consulta a efectuar em função da escolha*

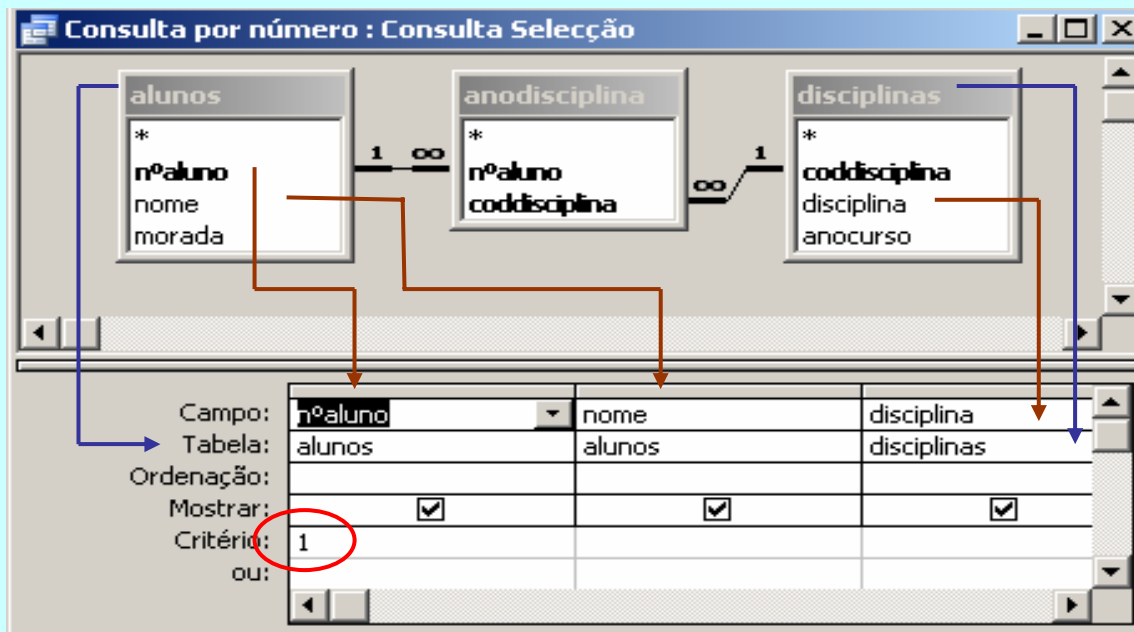
# SGBD: Como efectuar uma consulta TIC

No ecrã seguinte (vista de estrutura) seleccionamos as tabelas sobre as quais pretendemos efectuar interpelações.



# SGBD: Como efectuar uma consulta TIC

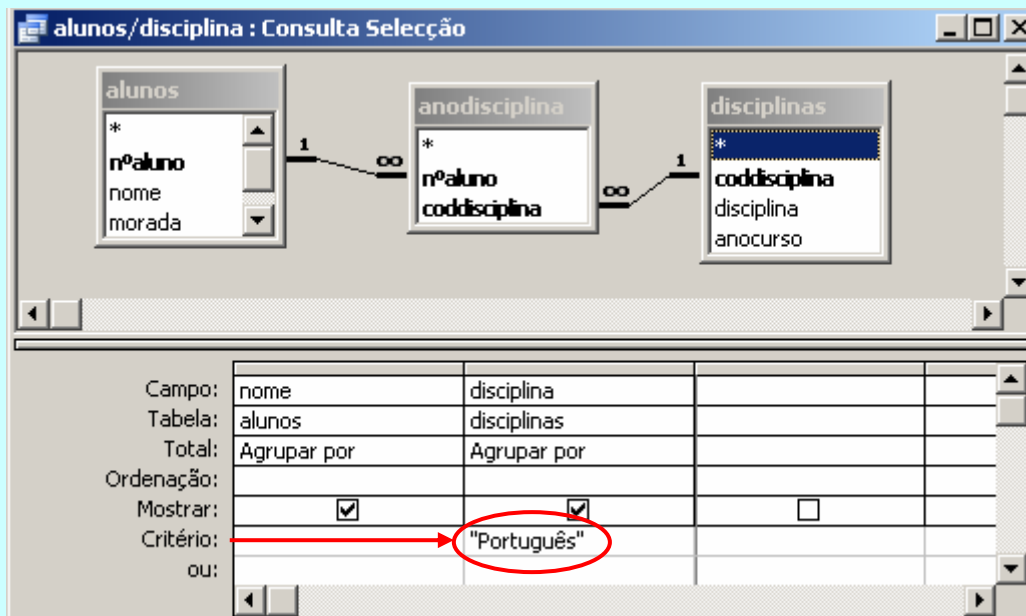
Seleccionamos os campos sobre os quais pretendemos questionar a base de dados e especificamos critérios ou condições de consulta.



| Operador                | Exemplo                               | Operador      | Descrição   |
|-------------------------|---------------------------------------|---------------|---|
| =<br>><br><<br>>=<br><> | = 1<br>(valores numéricos)            | soma<br>média | Para campos numéricos e monetários                |
| como                    | como "João"                           | min<br>max    | Em campos com texto usa a ordem alfabética        |
| não                     | não "João"                            | contar        | Conta o número de registos com valor não nulo     |
| entre ... e             | entre #1-1-1998# e #1-7-1998#         | agrupa<br>por | Agrupar registos pelos valores do campo escolhido |
| ou<br>e                 | "João" ou "Clara"<br>"João" e "Clara" |               |   |

## Exemplo

O objectivo desta consulta é visualizar os alunos que estejam inscritos à disciplina de Português.



The resulting query result window, titled "alunos/disciplina : Consulta Selecção", displays the following data:

|   | nome           | disciplina |
|---|----------------|------------|
| ▶ | Ana Figueiredo | Português  |
|   | Pedro Andrade  | Português  |

## 5ª Tarefa – Elaborar consultas

### Objectivos

- Criar consultas simples à base de dados:
  1. Com vários operadores;
  2. De referência cruzada;
  3. De acção.

😊 **Consultar a sala de aula virtual**