**Database Management System (DBMS):** “is a generalpurpose software system that facilitates the processes of defining, constructing, manipulating, and sharing databases among various users and applications.”

* O acesso à BD é sempre mediado pelo SGDB
* Existe uma interface de acesso que esconde os detalhes de armazenamento físico dos dados
* Elevada abstração ao nível aplicacional
* Os dados estão integrados (nível lógico) numa mesma unidade de armazenamento
* Suporta uma ou mais BD
* Independência entre programas e dados
* Integridade dos dados
* Consistência dos dados
* Eficiência no acesso aos dados
* Isolamento utilizadores
* Melhor gestão do acesso concorrencial
* Serviços de Segurança
* Mecanismos de backup e recuperação de dados
* Administração de dados
* Linguagem de desenho e manipulação de dados

**Utilizadores**

**Utilizadores Finais**

▪ aqueles que usam o sistema com determinada finalidade com recurso a ferramentas disponibilizadas pelo fabricante do sistema ou aplicações de terceiras entidades.

**Programadores de Aplicações**

▪ Desenvolvem aplicações que permitem que os utilizadores interajam com a base de dados. Podem utilizar várias linguagens de programação.

**Administradores da Base de dados**

▪ Tratam dos processos de gestão e manutenção da base de dados.

**Modelos de Base de Dados**

1ª Geração (Pré-relacional)

▪ Hierárquico

▪ Rede

2ª Geração

▪ **Relacional**

3ª Geração (Pós-relacional)

▪ Object-relational

▪ Object-oriented

▪ Key-value store

▪ Document-oriented

▪ Column-oriented

▪ Graph database

**Modelo Hierárquico**

* Dados estão armazenados numa estrutura hierárquica (árvore).
* Os nós da árvore designa-se como registos que estão ligados por ponteiros (links).
* Um registo é composto por um conjunto de atributos.
* Um link é uma associação entre dois registos do tipo pai-filho.
* Um registo pai encontra-se associado a N registos filhos (1:N)
* Adaptado a cenários de acesso sequencial aos dados (A maior parte das necessidades atuais requer acesso aleatório!)
* Redundância de informação
* Restrições de integridade, exemplo: A eliminação de um segmento pai, implica a remoção de todos os segmentos filhos associados.
* Não permite estabelecer associações N:M

**Modelo de Rede**

* Extensão do modelo hierárquico.
* Permite que um mesmo registo esteja envolvido em várias associações -> visão de rede.
* Melhorias na capacidade de navegação na estrutura de dados.
* Relações representadas através de grafos
* Um conjunto (set) suporta associação entre registos do mesmo tipo (Tipicamente implementados com listas ligadas circulares).
* Relacionamento 1:N entre dois tipos de registo.

**DER - Entidade/Relacionamento**

**Entidades**

* Fortes (Não dependem de outras entidades)
* Fracas (Dependem de outras entidades)

**Atributos**

* Derivados (tracejado)
* Multivalor (acoes\_formação) forte
* Compostos (tem mais atributos)

**Relacionamentos**

Interacções entre 2 ou mais entidades. Podem ter atributos...

* Grau ( Número de entidades envolvidas no relacionamento)
* Obrigatoriedade (Da participação da entidade na relação)
* Cardinalidade (Relação entre o número de ocorrências numa entidade com as respetivas ocorrências na outra com a qual estabelece o relacionamento.)

**Uma imagem com diagrama, file, Tipo de letra, texto

Descrição gerada automaticamenteUma imagem com texto, diagrama, Tipo de letra, file

Descrição gerada automaticamenteGrau da Relação**

* Unária
* Binária

Binária

* Ternária

Unária

* Múltiplos
* Recursivos (unárias)

(assimétricas - é necessário indicar os papéis (roles)

**Uma imagem com diagrama, file, origami

Descrição gerada automaticamenteUma imagem com texto, diagrama, file, Tipo de letra

Descrição gerada automaticamente**

**Uma imagem com diagrama, file, design, origami

Descrição gerada automaticamente**

Recursivos

Múltiplos

Ternária

**Cardinalidade**

Relação entre o número de ocorrências numa entidade com as respectivas ocorrências na outra com que tem o relacionamento.

**Uma imagem com diagrama, círculo, captura de ecrã, file

Descrição gerada automaticamente**

**Uma imagem com texto, Tipo de letra, file, diagrama

Descrição gerada automaticamente**

**Obrigatoriedade de Participação na Relação**

* Participação total (obrigatório)

cada instância da entidade participa em pelo menos uma relação do conjunto de relações (linha dupla).

Uma imagem com Tipo de letra, file, diagrama, captura de ecrã

Descrição gerada automaticamente

* Participação parcial (opcional)

alguma(s) instância(s) da entidade podem não participar em qualquer relação do conjunto de relações.

**Uma imagem com texto, diagrama, file, Tipo de letra

Descrição gerada automaticamente**

**Obrigatoriedade - Notação E/R (min,max)**

Existe uma notação alternativa com (min,max) para impor restrições à participação de cada entidade na relação.

**Uma imagem com texto, captura de ecrã, diagrama, Tipo de letra

Descrição gerada automaticamente**

* Mínimo
  + Se “0”, é opcional a participação da entidade na relação.
  + Se “1”, é obrigatória a a participação da entidade na relação.
* Máximo
  + Se “1”, cada instância da entidade está, no máximo, associada a uma única instância da relação.
  + Se “N”, cada instância da entidade está associada a várias instâncias da relação. • Uma notação alternativa especifica o número máximo de associações, por exemplo: 4, 8, 20, etc

**Restrições de Integridade**

São invariantes que a base de dados deve garantir

Tipos de Restrições:

* Atributos
  + Cada atributo só tem um valor
  + Atributos chave são únicos
  + Atributo (deve / pode ter) ter um valor
  + Valor do atributo pode ter restrições (>, <, !=, not null, etc)
* Cardinalidade do Relacionamento
  + 1:1 (um-para-um)
  + 1:N (um-para-N)
  + N:M (muitos-para-muitos)
* Obrigatoriedade de participação das entidades nas associações.

**Agregação**

Às vezes temos necessidade de modelar uma relação entre uma entidade e outra relação envolvendo outras entidades.

Tornar uma relação numa entidade associativa.

Entidade Associativa - Permite associar entidades a relacionamentos.

**Uma imagem com texto, captura de ecrã, file, diagrama

Descrição gerada automaticamente**

**Generalização versus Especialização**

Classificação de entidades em hierarquia de classes.

As sub-entidades herdam os atributos das super-entidades.

**Uma imagem com texto, diagrama, Tipo de letra, file

Descrição gerada automaticamente**

**Restrição de Sobreposição (overlaping)**

* Disjuntas: uma entidade só pode pertencer, no máximo, a uma subclasse de especialização (disjoint – ao lado do Δ).
* Sobrepostas: uma ocorrência de entidade genérica pode ter mais de uma especialização.

**Restrição de Completude (covering)**

* Total: uma entidade de nível superior tem de pertencer a pelo menos uma subclasse de especialização (linha dupla).
* Parcial: pode não pertencer a nenhuma.

**Modelo Relacional**

**Conceitos**

* Base do Modelo Relacional – Relação (Tabela)

**Uma imagem com texto, captura de ecrã, diagrama, file

Descrição gerada automaticamente**

* Atributo (A1, A2,..., An)
  + Representam o tipo de dados a armazenar.
  + O número de atributos de uma relação define o grau da relação
  + Os atributos de uma relação devem ter nomes distintos
* Domínio (D1, D2,...,Dn)
  + Tipo de dados
  + Gama de valores possíveis para determinado atributo. (Sexo {‘M’, ‘F’} Cidade {Porto, Aveiro, Coimbra,...} Nome {Maria, João, Ana, Sofia,...})
  + Valores desconhecidos ou não existentes. (NULL)
* Esquema da Relação - R(A1, A2,...,An)
  + Relational Schema
  + Nome do esquema e lista de atributos, Pessoa(nome, bi, idade)
  + Opcionalmente: inclui o tipo dos atributos Pessoa(nome:string, bi:integer, idade:integer)
* Relação - r(R)
  + Estrutura bidimensional com determinado esquema e zero ou mais instâncias (tuplos). r = {t1, t2, ..., tm}
  + Formalmente é um subconjunto do produto cartesiano r(R) ⊆ (dom(A1) × dom(A2) × ... × dom(An))
* Tuplo
  + Linha de uma relação. t = <v1,v2,…,vn>
  + Devem ser distintos (numa relação) -> Set
  + A ordem das linhas é indiferente.
  + O número de tuplos define a cardinalidade da relação.
* Atomicidade
  + O valor de um atributo num tuplo é atómico (não é composto/multi-valor).
* Esquema da Base de Dados (Database Schema)
  + conjunto de todos os esquemas da relação da BD. D = {R1(X1), ..., Rn(Xn)}

**Chaves**

* Superchave (superkey): conjunto de atributos que identificam de forma única os tuplos da relação.
* Chave Candidata (candidate key): subconjunto de atributos de uma superchave que não pode ser reduzido sem perder essa qualidade de superchave.
* Chave Primária (primary key): chave principal selecionada de entre as chaves candidatas.
* Chave Única (unique key): chave candidata não eleita como primária.
* Chave Estrangeira ou importada (foreign key): conjunto de um ou mais atributos que é chave primária noutra relação.

Cada relação tem pelo menos uma superchave (Conjunto de todos os atributos)

**Chave Primária**

A escolha da chave primária (de entre as candidatas) é arbitrária.

As chaves candidatas não eleitas (primária) designam-se como chaves únicas.

A chave primária não pode ter valor NULL.

**Conversão do DER em Modelo Relacional**

Um desenho conceptual de uma base de dados, utilizando DER, pode ser representado por intermédio de um conjunto de relações (tabelas)

Cada conjunto de entidades e relações do DER vai gerar uma única relação (tabela) com o nome do respectivo conjunto.

DER Empresa – Caso de Estudo

**Uma imagem com texto, diagrama, file, origami

Descrição gerada automaticamente**

**passo 1**

* Para cada entidade regular E do esquema ER, criar uma relação (tabela) R e inclui todos os atributos de E.
* Incluir os atributos compostos como elementos singulares.
* Selecionar uma das chaves de E para chave primária de R.

**Uma imagem com texto, captura de ecrã, Tipo de letra, número

Descrição gerada automaticamente**

**passo 2**

* Cada entidade fraca W do esquema ER é representada por uma relação (tabela) R que inclui os seus atributos, assim como a chave primária da entidade dominante E que passará a ser chave estrangeira em R.
* Incluir os atributos compostos de W, caso existam, como elementos singulares.
* A chave primária de R é a combinação da chave primária de E e da chave parcial de W.

**Uma imagem com texto, Tipo de letra, captura de ecrã, file

Descrição gerada automaticamente**

**passo 3**

* Para cada relacionamento 1:1 do esquema ER, envolvendo as relações S e T:
  + escolher uma das relações, digamos S, e incluir como chave estrangeira, a chave primária da outra relação.
  + incluir em S eventuais atributos do relacionamento.
  + devemos escolher como S uma relação com participação total.

**Uma imagem com texto, captura de ecrã, Tipo de letra, file

Descrição gerada automaticamente**

**passo 4**

* Para cada relacionamento 1:N do esquema ER, envolvendo as relações S e T:
  + escolher como S a relação que representa a entidade do lado N e como T a que representa a entidade do lado 1.
  + incluir em S, como chave estrangeira, a chave primária da relação T.
  + incluir os atributos do relacionamento em S.

**Uma imagem com texto, file, captura de ecrã, diagrama

Descrição gerada automaticamente**

**passo 5**

* Para cada relacionamento N:M do esquema ER, criar uma nova relação (tabela) R.
  + incluir como chave estrangeira as chaves primárias das relações que participam em R. Estas chaves combinadas formarão a chave primária da relação R.
  + incluir os atributos do relacionamento em R.

**Uma imagem com texto, captura de ecrã, Tipo de letra, diagrama

Descrição gerada automaticamente**

**passo 6**

* Para cada atributo multi-valor A do esquema ER, criar uma nova relação (tabela) R.
  + incluir um atributo correspondendo a A
  + incluir a chave primária K da relação que tem A como atributo.
  + a chave primária de R é a combinação de A e K.

**Uma imagem com texto, Tipo de letra, diagrama, captura de ecrã

Descrição gerada automaticamente**

**passo 7**

* Para cada relacionamento n-ário (n>2):
  + criar uma nova relação (tabela) R
  + incluir, como chaves estrangeiras, as chaves primárias das relações que representam as entidades participantes
  + incluir os eventuais atributos do relacionamento
  + a chave primária de R é normalmente a combinação das chaves estrangeiras

**Uma imagem com texto, captura de ecrã, diagrama, Tipo de letra

Descrição gerada automaticamente**

**Esquema da BD**

**Uma imagem com texto, diagrama, captura de ecrã, Paralelo

Descrição gerada automaticamente**

**SQL – Tipo de Dados**

Tipos de dados básicos:

* Numbers
* Characters, strings
* Date e time
* Binary objects

**Álgebra Relacional**

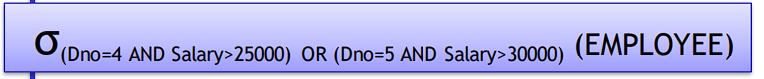
**Seleção**

Notação: σ <> (R)

* Utilizada para selecionar um subconjunto de tuplos da relação (t ∈ R) que satisfazem os critérios de seleção.
* “selection condition” é uma expressão boleana.

Relation2 ← σ (Relation1)

O resultado é uma nova relação (Relation2) que tem um esquema relacional igual à original (Relation1)



**Projeção**

Notação: π (R)

O resultado é uma nova relação só com os k atributos selecionados.

São removidas as linhas duplicadas do resultado

Condição de conjunto (set)

**Uma imagem com texto, Tipo de letra, captura de ecrã, Gráficos

Descrição gerada automaticamente**

**Renomeação**

Notação: ρR2(B1, B2, ..., Bn)(R1) ou ρR2(R1) ou ρ(B1, B2, ..., Bn)(R1)

* No primeiro caso o resultado é uma nova relação R2 com os atributos renomeados (B1, B2, ..., Bn).
* No segundo caso só renomeamos a relação.
* No terceiro só renomeamos os atributos.

**União**

Notação: R∪S={t:t∈R∨t∈S}

As tabelas têm de ser compatíveis

* Mesmo número de atributos
* Atributos com domínios compatíveis

O resultado é uma relação que inclui todos os tuplos de R e de S

* Os tuplos duplicados são eliminados

**Intersecção**

Notação: R∩S={t:t∈R∧t∈S}

As tabelas têm de ser compatíveis

* Mesmo número de atributos
* Atributos com domínios compatíveis

O resultado é uma relação que inclui os tuplos que existem simultaneamente em R e S

* Os tuplos duplicados são eliminados

Diferença

Notação: R-S={t:t∈r∧t ∉ s}

As tabelas têm de ser compatíveis

* Mesmo número de atributos
* Atributos com domínios compatíveis

O resultado é uma relação que inclui os tuplos de R que não existem em S

Em SQL existem os seguintes comandos ▪ UNION (ALL), INTERSECT (ALL) e EXCEPT

**Produto Cartesiano** “CROSS JOIN”

Notação: R X S

Permite-nos combinar tuplos de relações diferentes.

* ▪ O resultado é uma nova relação (Q) que combina cada elemento (tuplo) de uma relação (R) com um elemento (tuplo) da outra relação (S): Q(A1, A2, ..., An, B1, B2, ..., Bm) = R(A1, A2, ..., An) × S(B1, B2, ..., Bm)
* ▪ O número de tuplos de Q é n ∗ m.

**Junção θ (THETA JOIN)**

Notação: R ⋈C S

Pode ser visto como o resultado das seguintes operações: R3 ← R1 Χ R2 (produto cartesiano) σC (R3) (seleção com condição c)

C é <join condition> que pode tomar a seguinte forma: condition AND condition AND … AND condition

▪ Em cada condition podemos aplicar operadores de comparação: =, , ≥, ≠

**Divisão**

Notação: R ÷ S

Em SQL não existe um operador que implemente a divisão. Temos de recorrer a operadores básicos:

O resultado incluirá todos os tuplos de R1(A1 ,...,Ar) que tenham correspondência com todos os tuplos de S em R2(B1,…,BK). • R1 e R2 são projeções de R

número de atributos de R > número de atributos de S.

1. Num DER, as relações recursivas simétricas obrigam a indicar os papéis.

**Verdadeira**: No DER (Diagrama de Entidade-Relacionamento), as relações recursivas simétricas, onde uma entidade se relaciona consigo mesma, obrigam a indicar os papéis para distinguir os papéis das entidades envolvidas no relacionamento.

2.No Modelo Relacinal (MR), uma chave estrangeira não pode ter valor NULL.

**Falsa**: No Modelo Relacional (MR), uma chave estrangeira pode ter valor NULL, indicando que não há correspondência na tabela referenciada.

3. No processo de mapping do DER para Esquema Relacional umFiim atributo multi-valor resulta numa nova relação.

**Falsa**: No processo de mapeamento do DER para o Esquema Relacional, um atributo multi-valor resultaria em uma tabela adicional (ou seja, uma nova relação) apenas se não puder ser representado de outra forma, por exemplo, através de uma tabela de associação. Não é uma regra rígida que sempre resulta em uma nova relação.

4. Na definição da chave estrangeira, a opção "ON UPDATE SET NULL" coloca a chave primária (a referenciada) a NULL caso a chave estrangeira seja alterada.

**Falsa**: Na definição da chave estrangeira, a opção "ON UPDATE SET NULL" não coloca a chave primária a NULL caso a chave estrangeira seja alterada. Ela define o que acontece com a chave estrangeira quando a chave primária referenciada é atualizada, não o contrário.

5. O modelo de rede de base de dados tem menos redundância de dados do que o modelo hierárquico

**Falsa**: O modelo de rede de banco de dados geralmente tem mais redundância de dados do que o modelo hierárquico devido à possibilidade de múltiplas relações entre os mesmos conjuntos de dados.

6. No MR, uma tabela pode ter uma chave estrangeira para ela própria (isto é, importada da mesma tabela)

**Verdadeira**: No Modelo Relacional (MR), uma tabela pode ter uma chave estrangeira que faz referência à mesma tabela.

7. Num DER, uma relação IS-A com participação parcial (covering) obriga a que uma entidade de nível superior pertença a pelo menos uma subclasse de especialização

**Verdadeira**: Num DER, uma relação IS-A com participação parcial (covering) obriga a que uma entidade de nível superior pertença a pelo menos uma subclasse de especialização.

8. No MR, uma chave única é uma chave candidata, não eleita como primária e não pode conter valores NULL.

**Verdadeira**: No Modelo Relacional (MR), uma chave única (ou chave candidata) é uma chave que não foi escolhida como chave primária, mas ainda é única e não pode conter valores NULL.

9. No MR, uma tabela pode ter mais do que uma chave estrangeira para a mesma tabela destino.

**Verdadeira**: No Modelo Relacional (MR), uma tabela pode ter mais do que uma chave estrangeira para a mesma tabela destino.

10. Em SQL, uma vista(view) que utiliza agrupamento de dados pode ser utilizada como fonte de dados numa operação de inserção (SQL INSERT).

**Falsa**: Em SQL, uma vista (view) que utiliza agrupamento de dados geralmente não pode ser utilizada como fonte de dados numa operação de inserção (SQL INSERT), pois os dados agrupados podem não ter uma correspondência direta com as colunas na tabela de origem.

11. Um Natural JOIN ( R >< S) obriga a que as duas relações R e S tenham um atributo com o mesmo nome;

**Verdadeira**: Um Natural JOIN ( R >< S) requer que as duas relações R e S tenham um atributo com o mesmo nome para realizar a junção.

12. É possível executar o seguinte comando SQL "Alter table x add constraint myFK foreign key ( at1,at2) references Y (atI) and Z ( atII)".

**Falsa**: Não é possível executar o comando SQL "Alter table x add constraint myFK foreign key ( at1,at2) references Y (atI) and Z ( atII)". A cláusula **references** deve ser usada apenas uma vez por restrição de chave estrangeira. Para ter várias chaves estrangeiras, você precisa usar múltiplas cláusulas **foreign key**.