

I - Introdução

BD e SGBD

Um **banco de dados (BD)** é uma coleção de dados relacionados e armazenados em algum dispositivo, que tem um significado inerente e um propósito específico. Uma **instância** de um BD é o estado do banco de dados num determinado instante.

Um **sistema de gestão de banco de dados (SGBD)** é um software que permite construir e manipular um BD que tem uma arquitetura com propriedades:

- uma interface de alto nível de abstração onde se pode ter acesso à consultas, manipulação e definição de dados, geração de relatórios e linguagem de 4ª geração
- tradutor/otimizador de consultas
- visões do usuário de BD
- controle de concorrência
 - sincronização de acessos simultâneos ao BD
- controle de integridade
 - validação de restrições de integridade
- controle de segurança
 - autorização de acesso aos dados
- controle de recuperação
 - tolerância à falhas
- eficiente sistema de arquivos com técnicas de indexação eficientes que permitem o armazenamento e manipulação de dados

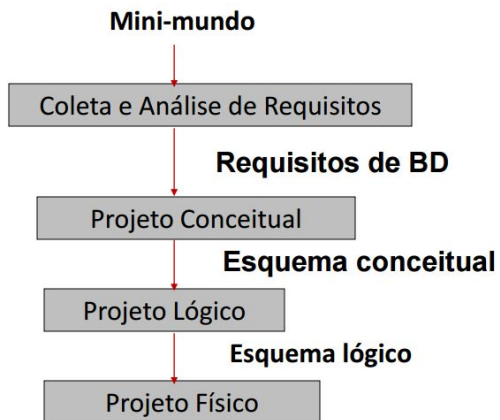
Em comparação com um **sistema de arquivos** o SGBD tem uma redundância de dados controlada, usa linguagem de alto nível (no sistema de arquivos a linguagem é procedural), tem independência entre linguagem-programa, oferece múltipla visão dos dados (no sistema é oferecida apenas uma visão) e além disso conta com concorrência, tolerância à falhas, integridade e segurança (no sistema isso é de responsabilidade da própria aplicação).

Evolução dos SGBD's

- Anos 60:
 - grafos
- Anos 70:
 - tabelas
- Anos 80:
 - distribuídos
- Anos 90:
 - orientado a objetos
 - relacional estendido (objeto-relacional)
- Anos 00:
 - BD em sensores e plataformas móveis (autonomic computing)
 - BD XML
- Anos 10:
 - Opensource

- memória do disco

Modelos e Esquemas



Conceitual

Um **modelo conceitual** é aquele que não é utilizado pelos SGBDs mas é utilizado na fase conceitual do projeto de um BD. Ex.: MER, UML.

Um **esquema conceitual** é a descrição conceitual de um BD específico, segundo um modelo conceitual. Ex.: ER.

Lógico

Um **modelo lógico** é o modelo utilizado pelos SGBDs e é lógico pois sua implementação não precisa ser conhecida. Ex.: Modelo Relacional.

Um **esquema lógico** é a descrição das estruturas e das operações de um BD específico, utilizando um modelo de dados. Ex.: Esquema Relacional

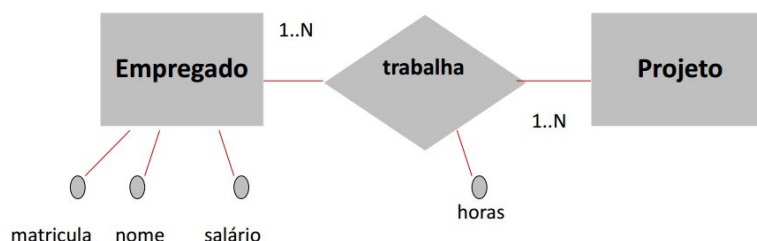
Interno (Físico)

Um **modelo interno (ou físico)** é aquele utilizado para implementar um modelo lógico. Cada SGBD tem seu modelo interno.

Um **esquema interno (ou físico)** é a descrição interna de um BD, segundo o modelo interno (ou físico).

II - Modelo Conceitual

Modelo de Entidades e Relacionamentos



Exemplo de um modelo simples de entidades e relacionamentos

Uma **entidade** é tudo aquilo sobre o qual se deseja manter informações, sendo um subconjunto de objetos (instâncias) possui os mesmos tipos de propriedades (atributos) e desempenha o mesmo papel semântico. Uma **entidade fraca** é um tipo de entidade que depende da existência ou identificação de uma outra entidade. As entidades são descritas em um **dicionário de dados**.



Representação de entidades no diagrama E - R (entidades e relacionamentos)

Uma **instância** é um objeto de uma entidade com propriedades distintas dos outros objetos.

Um **atributo** é uma propriedade que caracteriza e descreve uma entidade ou relacionamento. Cada atributo possui um domínio que identifica o conjunto de valores permitidos para aquele atributo.

Um atributo também deve ser descrito num dicionário de dados.

<u>Entidade: EMPREGADO</u>
<u>Atributo: Data de Admissão</u>
<u>Descrição: data na qual foi assinado o contrato de trabalho entre a empresa e o empregado</u>
<u>Domínio: data posterior a 03/01/78 (data de criação da empresa) e a data de nascimento do empregado</u>

Representação de um dicionário de dados

Um atributo pode ser:

- simples:
 - idade, nome
- composto:
 - endereço
- simplesmente valorado:
 - PESSOA:Idade
- multivalorados:
 - PESSOA:TitulacaoSuperior(nenhum, Bel., Msc., PhD)
- derivados: podem ser determinados a partir de outros atributos/entidades

- Idade e dataAniversario

Os **relacionamentos** são associações entre diversas entidades.

As **restrições de integridade** caracterizam as restrições nas quais os relacionamentos entre entidades estão submetidos (regras do negócio) e podem ser caracterizadas:

- **Cardinalidade:** quantidade de instâncias que podem participar de um relacionamento.
 - Um_para_Um (1:1)
 - Um_para_Muitos (1:N)
 - Muitos_para_Um (N:1)
 - Muitos_para_Muitos (M:N)
- **Totalidade:** obrigatoriedade da ocorrência do relacionamento entre as entidades envolvidas.
 - O uso do “zero” indica a totalidade do relacionamento: (0:1) ou (0:N)

As **chaves de identificação** são atributos de valor único e não nulo que todas as instâncias de uma entidade deve ter.

- superchave
- superchave mínima
- composta
- surrogates (geradas automaticamente pelo SGBD)
- regras de integridade:
 - toda instância de uma entidade possui um valor para chave de identificação própria da entidade
 - o valor da chave de identificação própria para uma instância é único e não nulo dentro da entidade
 - o valor da chave de identificação própria de uma instância não deve ser modificado

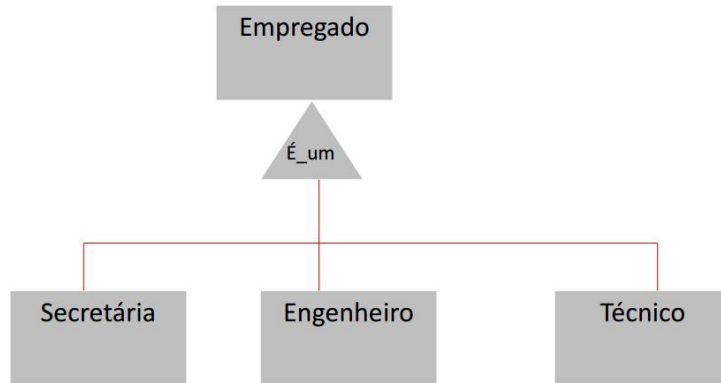
<i>Entidade</i>	<i>Chave</i>
Estado	Sigla do estado
Motorista	Sigla do estado + número da carteira de habilitação
Carro	Sigla do Estado Número da carteira de motorista Placa do Carro
Multa	Sigla do Estado Número da carteira de motorista Placa do Carro Número de sequência da multa

Exemplo de entidades e suas respectivas chaves

MER Estendido

Como, muitas vezes, uma entidade possui subentidades que necessitam ser representadas explicitamente há no MER Estendido a possibilidade de **superclasses** e **subclasses**.

A relação das subclasses é conhecida como “é um”, o que caracteriza uma **herança**. Vale ressaltar que nem toda instância da superentidade é membro de uma subentidade.



Representação visual de uma herança no MER

A **especialização** é o processo de definir um conjunto de subclasses de uma superentidade enquanto a **generalização** é o processo inverso.

As restrições de MER Estendido são:

- **cobertura total:** cada instância da superentidade deve ser instância de alguma subentidade.
- **cobertura parcial**
- **disjunção:** uma mesma instância pode ser membro de, no máximo, uma subentidade.
- **sobreposição:** uma mesma instância pode ser membro de mais de uma subentidade.

III - Modelo Relacional

O **modelo relacional** é um modelo formal baseado na teoria matemática das relações que representa dados em um BD como uma coleção de tabelas (relações). Cada tabela terá um nome que será único e um conjunto de atributos com seus respectivos nomes e domínios. Vale lembrar que todos os valores de uma coluna são do mesmo tipo de dados.

Na **terminologia** de um BD relacional uma linha é chamada de **tupla**, a coluna é chamada de **atributo** e cada tabela é chamada de **relação**. O **domínio** é definido como um conjunto de valores atômicos e cada domínio é associado à um tipo de dado ou formato. O **grau** de uma relação é o número de atributos que contém no seu esquema.

Matr	Nome	Endereço	Função	Salário	Depart
100	Ana	R. Pedro I, 12, A. Branco	Secretária	500,00	D1
250	Pedro	R. J. Silva, 24, Liberdade	Engenheiro	1500,00	D1
108	André	R. Italia, 33, B. Nações	Técnico	950,00	D2
210	Paulo	R. Pará, 98, B. Estados	Engenheiro	1810,00	D2
105	Sônia	R. Oliveira, 76, A. Branco	Engenheiro	2500,00	D1

Tabela de empregados em uma empresa

Características das relações:

- a ordem das tuplas e atributos não tem importância
- todo atributo possui valor atômico
- o nome de um atributo numa relação é único
- todas as tuplas devem ser únicas

Num modelo relacional é comum que as **chaves de uma relação** venham sublinhada.

As **restrições de integridade** são definidas em:

- integridade de chave: toda tupla tem um conjunto de atributos que a faz única numa relação
- integridade de entidade: nenhuma valor de chave primária poderá ser NULO
- integridade referencial: todo valor de chave estrangeira numa relação deve corresponder a um valor de chave primária de uma segunda relação ou deve ser nulo.

Departamento		
CodDep	Nome	MatrGerent
D2	Produção	210
D1	Custos	105
D5	Pessoal	NULL

Empregado					
Matr	Nome	Endereço	Função	Salário	Depart
100	Ana	R. Pedro I, 12, A. Branco	Secretária	500,00	D1
250	Pedro	R. J. Silva, 24, Liberdade	Engenheiro	1500,00	D1
108	André	R. Italia, 33, B. Nações	Técnico	950,00	D2
210	Paulo	R. Pará, 98, B. Estados	Engenheiro	1810,00	D2
105	Sônia	R. Oliveira, 76, A. Branco	Engenheiro	2500,00	D1

Exemplo de integridade referencial

Um conjunto de atributos de uma relação R1 é uma **chave estrangeira** se:

- Os atributos da chave estrangeira tem o mesmo domínio dos atributos da chave primária de uma relação R2
- Um valor da chave estrangeira da tupla t1 em R1 é de valor igual à uma tupla t2 em R2 ou é NULO.

Operações em Relações

1. Inserção:
 - a. Inserir <>
2. Remoção:
 - a. Remover da tabela x a tupla com $y = z$
 - b. a operação só pode violar a integridade referencial
3. Modificação:
 - a. Modificar x de y com $z = w$

Álgebra relacional

É uma linguagem de BD procedural e formal.

São as **operações**:

1. Seleção (σ): um operador unário que seleciona um subconjunto de tuplas de uma relação de acordo com uma condição onde o grau da relação resultante é o mesmo da relação original e o número de tuplas da relação resultante é menor ou igual à original.

a. $\sigma_{\langle \text{predicado} \rangle}(\langle \text{Relação} \rangle)$

$$\sigma_{\text{DEPTO} = 4}(\text{Empregado})$$

Seleciona os empregados do departamento 4

2. Projeção (π): seleciona um subconjunto de atributos de uma dada relação. Nessa operação duplicatas podem aparecer e o número de tuplas resultante será menor ou igual ao da relação original

a. $\pi_{\langle \text{lista de atributos} \rangle}(\langle \text{Relação} \rangle)$

$$\pi_{\text{NOME, SALÁRIO}}(\pi_{\text{NOME, FUNÇÃO, SEXO, SALÁRIO}}(\text{Empregado})) = \pi_{\text{NOME, SALÁRIO}}(\text{Empregado})$$

3. Combinando seleção e projeção:

$$\pi_{\text{NOME, SALÁRIO}}(\sigma_{\text{DEPTO} = 5}(\text{Empregado}))$$

Duas opções de combinar seleção e projeção

$$\text{EmpDepto5} \leftarrow \sigma_{\text{DEPTO} = 5}(\text{Empregado})$$

$$\text{Resultado} \leftarrow \pi_{\text{NOME, SALÁRIO}}(\text{EmpDepto5})$$

4. União (\cup): a união de duas relações é o conjunto das tuplas que estão em R **ou** S **ou ambas**. Neste caso as duplicatas são eliminadas.

$$\text{EmpDepto5} \leftarrow \sigma_{\text{DEPTO} = 5}(\text{Empregado})$$

$$\text{Temp1} \leftarrow \pi_{\text{MATRÍCULA}}(\text{EmpDepto5})$$

$$\text{Temp2} \leftarrow \pi_{\text{SUPERVISOR}}(\text{EmpDepto5})$$

$$\text{Resultado} \leftarrow \text{Temp1} \cup \text{Temp2}$$

Como obter a matrícula dos empregados que trabalham ou supervisionam quem trabalha no departamento 5

5. Interseção (\cap): é uma relação que inclui todas as tuplas que estão em R **e** em S.

6. Diferença ($-$): a diferença $R - S$ resulta no conjunto de tuplas que estão em R mas não estão em S .
7. Produto cartesiano (\times): um $R \times S$ combina cada tupla de R com cada tupla de S . Se R possui n atributos e S m , a relação resultante possui $n + m$ atributos. Se R possui x tuplas e S y , então a relação resultante possuirá $x \cdot y$ tuplas.

$Mulher \leftarrow \sigma_{\text{sexo} = 'F'} (Empregado)$

$NomesMulheres \leftarrow \pi_{\text{matricula}, \text{nome}} (Mulher)$

$DependentesMulher1 \leftarrow NomesMulheres \times Dependentes$

$DependentesMulher2 \leftarrow \sigma_{\text{matr} = \text{matrEmp}} (DependentesMulher1)$

$Resultado \leftarrow \pi_{\text{nomeE}, \text{nomeDep}} (DependentesMulher2)$

Obter para cada empregado do sexo feminino, uma lista com o nome dos seu dependentes

8. Junção (\bowtie): usada para combinar tuplas de duas relações em uma só com $n+m$ atributos

a. $R \bowtie_{\langle \text{condição de junção} \rangle} S$

$DeptoGer \leftarrow Departamento \bowtie_{\text{matrGer} = \text{matr}} Empregado$

$Resultado \leftarrow \pi_{\text{nomeD}, \text{nomeE}} (DeptoGer)$

Obter o nome do gerente de cada departamento

9. Divisão (\div): Numa divisão de R por S , os atributos de S devem estar contido nos atributos de R e esta operação é resulta numa relação $T = \{ \text{atributos}(R) - \text{atributos}(S) \}$. *(geralmente é utilizada em consultas com “para todos”)*

R	A B		S	A	R ÷ S	B
	A	B				
	a1	b1		a1		b1
	a2	b1		a2		
	a3	b1		a3		
	a4	b1				
	a1	b2				
	a3	b2				
	a2	b3				
	a3	b3				
	a4	b3				
	a1	b4				
	a2	b4				
	a3	b4				

10. Rename (ρ): permite que se renomeie relações e/o atributos com objetivo de evitar ambiguidade na hora de comparar atributos com mesmo nome de diferentes relações.

a. $\rho_{S(B1, B2, \dots, Bn)}(R)$, onde S é a nova relação, e $B1..Bn$ o nome dos novos atributos.

Importante: As operações de conjunto (união, interseção e diferença) devem ser **compatíveis de união**, ou seja, $\text{domínio}(R) = \text{domínio}(S)$ e $\text{grau de } R = \text{grau de } S$.

IV - Projeto de Banco de Dados Relacional

Transformações de Diagramas MER em Diagramas ER

Regras:

1. Entidades Regulares
 - a. vão ser transformadas em relações (tabelas). As propriedades todos se tornarão atributos (colunas) simples. Uma das chaves da entidade se tornará chave primária da relação.
2. Entidades Fracas
 - a. semelhante à regular, porém a chave primária da dominante será a chave estrangeira na entidade fraca, e a chave primária da fraca será a original com a dominante.
3. Relacionamentos 1:1
 - a. A totalitária obtém como chave estrangeira a chave primária da outra entidade.
4. Relacionamentos 1:N que não envolvem entidades fracas
 - a.
5. Relacionamentos N:M
6. Atributos Multivalorados
7. Especialização e Generalização