UNIVERSIDADE FEDERAL DE OURO PRETO – UFOP

CIÊNCIA DA COMPUTAÇÃO

BANCO DE DADOS I

RESUMO I

Gabriel Mace dos Santos Ferreira

Marcus Vinícius Souza Fernandes

Ouro Preto

2021

Introdução sobre Banco de Dados

A área de banco de dados é uma das mais importantes no contexto da ciência da computação, uma vez que em várias situações é desejável que dados sejam armazenados e manipulados.

1. Conceitos básicos:

1.1. Dados: Informações conhecidas e necessárias ao sistema que se deseja implementar, sendo possível armazená-las.

1.2. Banco de Dados: Coleção de dados relacionados que possuem as seguinte propriedades implícitas:

- Representa um aspecto do mundo real, ou seja, pertencem a um mesmo Universo de Discurso.
- Corresponde a uma coleção coerente de dados com significado inerente.
- Tem sua projeção, implementação e povoamento com dados feitos com um propósito específico em mente (tornando a inserção e manipulação desses mais eficiente).

1.3. Sistema de Gerência de Banco de Dados: Um software de propósito geral responsável pela criação e manutenção de um banco de dados, além disso o seu uso facilita alguns processos que envolvem um banco de dados, como:

- **Definir:** Especificar os tipos e as restrições sobre os dados armazenados em um banco de dados.
- **Construir:** Armazenar dados que seguem as diretrizes estabelecidas no banco de dados.
- **Manipular:** Possibilita a consulta e edição dos dados presentes no banco de dados.

1.4. Sistema de Banco de Dados: Sistema de informação responsável pela manipulação de um banco de dados por meio de um SGBD (Sistema de Gerência de Banco de Dados).

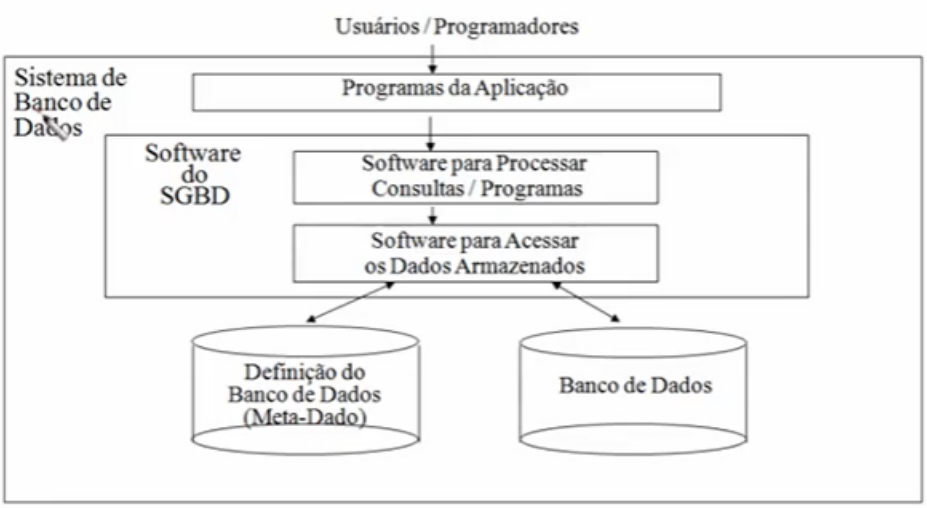

Imagem 1 - Estrutura do Sistema de Banco de Dados (retirado dos slides de introdução do professor Guilherme Tavares de Assis).

A imagem anexada acima ilustra perfeitamente o funcionamento de um sistema de Banco de Dados, onde usuários fazem uso de um determinado programa ou aplicação, esse software manipula as informações desejadas com a utilização de um **SGDB** que acessa um determinado banco de dados.

2. Banco de Dados x Sistema Tradicional de Arquivos:

2.1. Independência dos Dados:

- **Banco de Dados:** Há uma independência entre o programa e os dados, visto que as informações estão armazenadas no **Catálogo** (seção de Definição do Banco de Dados na Imagem 1) e são chamados de metadados, além disso ao utilizar a orientação por objetos, tem-se que as operações podem ser definidas como parte do banco de dados, gerando uma independência entre o programa e as operações.

- **Sistema tradicional de arquivos:** A estrutura dos dados é inerente ao programa que realiza o acesso desses, não sendo possível uma independência entre programas e dados.

2.2. Múltiplos usuários:

- **Banco de Dados:** É possível a definição e manutenção de um único banco de dados por diversos indivíduos, nesse caso faz-se necessário que o **SGDB** possua um software de controle de concorrência, visto que caso os dados estejam integrados e mantidos em um único banco de dados que permite o acesso de múltiplos usuários

concorrentemente, torna indispensável a verificação dos dados por meio do controle de concorrência.

- **Sistema tradicional de arquivos:** É necessária a definição e implementação dos arquivos necessários para cada aplicação de forma específica.

3. **Usuários de um SGBD:**

- **Projetista do Banco de Dados:** Profissionais responsáveis pela estruturação/modelagem do banco de dados, seja em sua representação ou em seu armazenamento, além disso, também são responsáveis pela identificação dos dados a serem armazenados.

- **Administradores do Banco de Dados:** Profissionais responsáveis pela administração do banco de dados, assim como de seu SGBD, por isso também possuem como responsabilidades: Autorizar o acesso dos usuários, coordenar e monitorar o uso do SGBD, verificar a necessidade de novos recursos de software e hardware para assegurar o bom funcionamento do SGBD.

- **Usuários finais:** Os usuários que irão acessar o banco de dados, realizando consultas, modificações e gerando relatórios, essa categoria pode ser dividida em:

  i) Ingênuos ou parametrizados: Usuários cuja função é realizar consultas e modificações no banco de dados frequentemente, por meio de ferramentas próprias construídas e testadas para esse intuito.

  ii) Casuais: Usuários que raramente precisam acessar o banco de dados, mas podem necessitar de informações distintas a cada vez que acessam.

  iii) Especializados: Usuários que realizam tarefas complexas no banco de dados, para isso devem possuir alta familiaridade com a SGBD.

4. **Funcionalidades de um SGBD:**

- **Controle de redundância:** A redundância de dados pode acarretar problemas como a ocupação do espaço em disco desnecessariamente, a repetição de uma alteração lógica e a inconsistência de dados. Vale ressaltar que em certos casos a

redundância dos dados pode ser proposital, feita pelo projetista de banco de dados com o intuito de facilitar a pesquisa dentro do banco (Utilizado apenas quando a pesquisa está ocorrendo de forma pior do que o desejado).

- **Restrição de acesso:** O controle do acesso sobre os dados e operações realizadas sobre esses é essencial em um banco de dados com múltiplos usuários.

- **Suporte a múltiplas visões dos dados:** Uma visão se trata de um subconjunto do banco de dados que pode se destinar a um grupo específico de usuários.

- **Garantia das restrições de integridade:** As restrições presentes nessa funcionalidade podem ser divididas em simples, como restrições sobre o tipo de dado ou podem ser complexas, como a verificação da relação de um item em um arquivo com itens nos demais arquivos (Usualmente as restrições se encontram no **Catálogo).**

- ***Backup* e recuperação:** A recuperação dos dados em caso de falha de hardware ou de software, além disso a possibilidade de gerar um banco de dados confiável e consistente.
    - Sequencial: *Backup* completo do banco de dados.
    - Incremental: *Backup* do banco de dados que tange as atualizações de um determinado intervalo de tempo.

- **Múltiplas interfaces de usuários:** Os usuários do banco possuem diferentes níveis de conhecimento técnico, dessa forma interfaces para cada um desses níveis são essenciais.

- **Armazenamento persistente para objetos e estruturas de dados de programas:** Caso um objeto continue a existir após o fim da execução de um programa ele é considerado persistente, além disso deve ser possível acessá-lo diretamente por outro programa

- **Inferência em banco de dados usando regras de dedução:** Essa funcionalidade é característica de Bancos de Dados Dedutivos, no entanto tais bancos não são mais amplamente utilizados. Basicamente gera informações inteligentes dentro dos dados armazenados no banco com base em regras pré criadas.

5. **Quando não usar o SGBD:**

5.1. Desvantagens SGBD: Ao utilizar um SGBD tem-se um aumento de custo em relação a um sistema tradicional de arquivos, devido a:

i) Investimento inicial em software, hardware e treinamento;

ii) Generalidade para definir e processar dados;

iii) Fornecimento de mecanismos de integridade, segurança, controle de concorrência e recuperação de falhas.

5.2. Quando utilizar um Sistema de Arquivos Tradicional:

i) O banco de dados, assim como as aplicações desenvolvidas são simples, bem definidas e a uma baixa possibilidade de mudança;

ii) Não é necessário o acesso de múltiplos usuários aos dados;

iii) São necessários requisitos de tempo-real do programa.

6. Processo metodológico para concepção de um banco de dados:

Imagem 2 - Processo metodológico para concepção de um Banco de Dados (retirado dos slides de introdução do professor Guilherme Tavares de Assis).

O processo é dividido em três etapas, onde cada uma delas corresponde a um projeto diferente (conceitual, lógico e físico), eles ocorrem logo após a definição do minimundo e a coleta e análise de requisitos

6.1. Modelo de dados: Trata-se de um conjunto de conceitos que podem ser utilizados na etapa de descrever a estrutura de um determinado banco de dados.

- Ele é categorizado de acordo com os tipos de conceito que podem descrever a estrutura de um banco, variando entre conceitual, lógico ou físico.
- A estrutura de um banco de dados é composta pelos tipos de dados, seus relacionamentos e restrições.

i) Conceitual (ou alto-nível): Fornece conceitos que estão próximos a concepção do usuário, tais conceitos: entidade (objetos e conceitos do mundo real que não dependem de outro indivíduo para existir), relacionamento (interação entre as entidades) e atributo (propriedade de uma entidade ou relacionamento).

ii) Lógico (ou representacional): Fornece conceitos que podem ser compreendidos pelos usuários e representam os dados em forma de registro. Exemplos: Modelo Relacional (mais utilizado), Rede e Hierárquico.

iii) Físico (ou baixo-nível): É bem distante da percepção do usuário, descreve detalhes de como os dados estão fisicamente armazenados. Alguns conceitos utilizados são: Formato e ordenação de registros e vias de acesso (recuperação de registros).

6.2. Esquema: É a descrição de um banco de dados em forma textual ou gráfica (diagrama do esquema) de acordo com um modelo específico.

6.3. Instância: Representa os dados que estão populando o banco num determinado instante de tempo. Estes dados são frequentemente

atualizados portanto a instância também será atualizada. O **SGBD** possui a função de garantir que toda nova instância seja válida.

7. **Arquitetura de Três Níveis (Três Esquemas):**

A arquitetura de três níveis tem como objetivo separar o banco de dados físico e as aplicações dos usuários. Ela possui os níveis:

i) Interno: Todas as definições conceituais abordadas no item abaixo se encontram neste passo, em sua forma física.

ii) Conceitual: Descrição da estrutura do banco de forma completa (Não possui relação com o modelo conceitual, é apenas uma coincidência na nomenclatura).

iii) Externo ou de Visão: Porção do esquema conceitual que um determinado grupo de usuários pode interagir.

Em suma, os esquemas representam apenas descrições de dados, estes dados só existem de fato no nível físico.

O **SGBD** tem o papel de transformar as requisições do esquema externo para o conceitual, que por sua vez, transforma para o esquema interno, onde são processadas sobre o banco de dados.

A independência de dados é a capacidade de alterar um nível da arquitetura do esquema sem influenciar no nível presente acima. Existem dois tipos de independências:

i) Independência de dados lógica: Capacidade de alterar o esquema conceitual sem impactar mudanças em algum esquema externo ou aplicação.

ii) Independência de dados física: Capacidade de alterar o esquema interno sem implicar em mudanças no esquema conceitual ou em no externo.

8. **Linguagens de um SGBD:**

Em todo **SGBD** há linguagens internas que possuem comandos utilizados em propósitos específicos, vale ressaltar que cada uma dessas linguagens foi criada com a arquitetura de três níveis em mente. Dessa forma, podemos citar algumas, como:

i) Linguagem de Definição de Dados (LDD): Especifica o esquema conceitual da arquitetura de três níveis.

ii) Linguagem de Definição de Armazenamento (LDA): Especifica o esquema interno da arquitetura de três níveis (esquema físico).

iii) Linguagem de Definição de Visões (LDV): Especifica as visões dos usuários e o mapeamento para esquema conceitual da arquitetura de três níveis.

iv) Linguagem de Manipulação de Dados (LMD): Usada para consultar, inserir, remover e modificar dados do banco de dados.

Obs: A linguagem SQL é uma linguagem de banco de dados relacional compreensível, ou seja, é uma combinação de todas as linguagens citadas acima.

9. Interfaces de um SGDB:

Como mencionado previamente no texto, usuários com diferentes níveis de conhecimento técnico podem interagir com o **SGBD**, portanto são necessárias diferentes interfaces, como:

- **Interfaces baseadas em menus para browsing:** Seria a representação da estrutura do banco por meio de menus, ou seja, operações feitas no banco são feitas por meio dessas. Usualmente utilizado por usuários com baixo conhecimento técnico.

- **Interfaces baseadas em formulários:** O **SGBD** gera um formulário que por sua vez é disponibilizado aos usuários, de forma que esses possam interagir com o banco.

- **Interfaces gráficas:** Interfaces em que a estrutura do banco é representada de forma gráfica, permitindo ao usuário interagir com seus componentes, por exemplo a esquemática da ferramenta mySQL.

- **Interfaces de linguagem natural:** Interfaces voltadas para bancos inteligentes, previamente mencionados no texto. Nesse tipo, as operações são feitas por meio da linguagem natural. Esse tipo de interface apresenta problemas quanto a interpretação da linguagem, impedindo a eficiência ótima.

- **Interfaces para o Administrador de Banco de Dados:** Essas interfaces são utilizadas para uma interação direta com o banco, seja por meio de prompts de comando.

Vale ressaltar que os **SGBDs** devem oferecer diferentes tipos de interfaces para seus diferentes tipos de usuários.

10. Módulos Componentes de um SGDB:

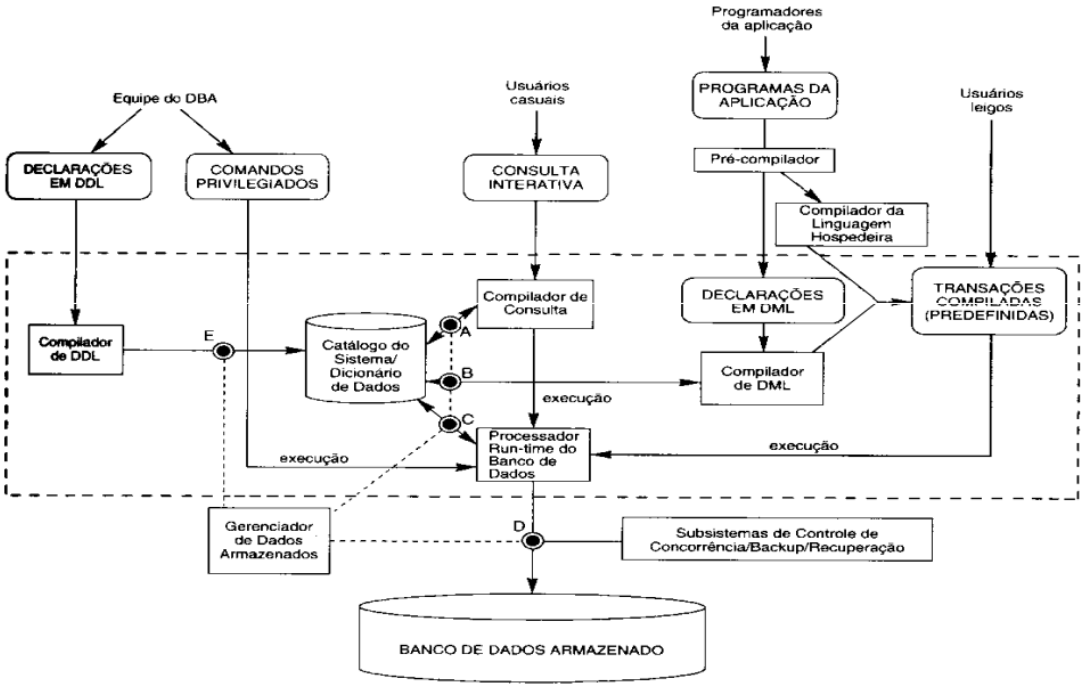

*Imagem 3 - Componentes de um **SGBD** (retirado dos slides de introdução do professor Guilherme Tavares de Assis).*

A figura acima apresenta os principais componentes de um bom **SGBD**, listando os componentes mais relevantes temos: Compilador de DDL, Compilador de Consulta, Gerenciador de Dados Armazenados, Processador Run-time do Banco de Dados, Pré-compilador e o Compilador de DML.

A parte tracejada corresponde ao **SGBD**, a parte superior representa a interação com o banco e a inferior os dados propriamente ditos no banco de dados armazenado.

Módulos componentes de um **SGBD**:

- **Gerenciador do banco de dados:** Gere o acesso aos dados armazenados no disco (Banco).

- **Compilador da LDD:** Compila as definições/configurações de um esquema, registrando estas no catálogo do SGBD.

- **Processador do banco de dados em tempo de execução:** Recebe todas as operações de recuperação e modificação e as executam sobre o banco.

- **Compilador de consultas:** Compila as consultas de alto nível que são fornecidas de forma interativa.

- **Pré-compilador:** Extrai comandos LMD de um programa de aplicação escritos em uma linguagem de programação hospedeira.

- **Compilador da LMD:** Recebe e compila os comandos LMD extraídos pelo pré-compilador, gerando o código objeto específico (O código do objeto e o restante do programa são unidos e compõem uma única transação).

11. Utilitários:

Ao adquirir um SGBD, alguns utilitários são disponibilizados através da compra ou como adicional gratuito, alguns deles são:

- **Carregador:** Utilizado para povoar (preencher dados) o banco com informações/arquivos existentes.

- **Backup:** Gera uma cópia de segurança (backup incremental e/ou sequencial) do banco.

- **Monitor de eficiência:** Disponibiliza estatísticas sobre o banco de dados através da monitoração.

- **Reorganizador:** Reorganiza os arquivos visando ganho de desempenho, melhorando a eficiência do banco de dados, usualmente utilizado em parceria com o Monitor de eficiência.

12. Critérios de Classificação de gens de SGBDs:

Existem diferentes **SGBDs** atualmente, dessa forma é necessária a sua classificação seguindo os seguintes critérios:

12.1. Modelo de dados lógico: Classificação de acordo com o modelo de dados lógicos escolhido na etapa de projeto lógico na concepção do banco, sendo possível dividi-los em: relacional (mais utilizado atualmente), hierárquico, rede, orientado a objeto e objeto-relacional.

12.2. Número de servidores: Classificação de acordo com o número de servidores utilizados no armazenamento do **SGBD**, sendo possível dividir essa classificação em:

- **Centralizado:** Um único servidor é utilizado para armazenar o banco de dados e o **SGBD**, nesse tipo as máquinas clientes fazem requisições ao servidor centralizado.

- **Distribuído (SGBDD):** Diversos servidores armazenam o banco de dados e o **SGBD**, esse tipo pode ser dividido em:

  i) SGBDD Homogêneo: Todos os servidores utilizam um único SGBD.

  ii) SGBDD Federado: Os servidores utilizam diferentes **SGBDs** , portanto é necessário uma forma eficiente e eficaz de comunicação entre eles.

Modelo conceitual ER e ERE

1. **Modelo ER:**

O modelo de Entidades e Relacionamentos (**MER**) é o modelo conceitual mais utilizado atualmente no projeto de aplicações de banco de dados. Ele é baseado na percepção dos objetos no mundo real como entidades e a relação entre esses objetos. Vale ressaltar que esse modelo independe dos aspectos de implementação.

1.1. Exemplo de Modelos ER: Suponha a necessidade de criação de um banco de dados para gerência de um curso de Ciência da Computação, nesse sentido inicialmente deverão ser coletados e analisados os requisitos para a criação desse banco. Inicialmente os projetistas desenvolvem um “mini-mundo” ao fazer uma distinção dentro daquele curso entre professores, matérias e alunos. Cada um desses possui propriedades especiais e relações.

A partir do exemplo citado acima, é possível deduzir o funcionamento do modelo **ER** dado que os itens estão sendo percebidos como objetos do mundo real.

1.2. Entidade: Uma **Entidade** é um objeto presente no mundo real independente e distinguível dos demais, ou seja, ele possui características únicas e não necessita de outras entidades para existir. Por exemplo, dentre os matriculados em uma disciplina todos são alunos, no entanto cada indivíduo é uma **Entidade** por possuir uma matrícula exclusiva.

- **Tipo de Entidade:** Define um grupo de entidades que possuem os mesmos atributos, sendo possível descrever o esquema para um **conjunto de entidades**.

- **Conjunto de Entidades:** A coleção das instâncias de entidades de um tipo específico. Suponha um grupo de todos os alunos da UFOP, essas entidades podem ser definidas como o conjunto Aluno.

1.3. Atributo: Uma característica de uma entidade é denominada como um **Atributo**. Vale ressaltar que os atributos podem ser classificados em:

- **Simples (atômicos) ou Compostos:** Os atributos simples não podem ser subdivididos enquanto que os atributos compostos podem. Ex.: Suponha um atributo nome para a entidade Aluno acima, esse pode ser dividido em nome e sobrenome, enquanto isso um atributo simples como o sexo do Aluno não pode ser subdividido. Vale ressaltar que a classificação como atributo Composto é subjetiva e deve ser feita de acordo com a modelagem e o propósito do banco.

- **Mono-valorados ou Multi-valorados:** Quando um atributo possui um único valor para uma entidade em especial ele é denominado monovalorado, caso seja um conjunto de valores ele é multi-valorado. Ex.: Utilizando a ideia de um banco para o curso de Ciência da Computação mencionado acima, podemos imaginar que para o nome da entidade Matéria é mono-valorado, enquanto que o atributo alunos matriculados é multi-valorado, dado que diferentes alunos podem estar

matriculados em uma determinada disciplina.

- **Armazenados ou Derivados:** Os atributos armazenados são como o nome diz aqueles que estão armazenados no banco de dados, enquanto isso os derivados podem ser obtidos a partir dos armazenados. Ex.: É possível deduzir o atributo alunos_ensinados da entidade Professor, a partir da relação materia_ensinada e o número de alunos matriculados na entidade Matéria.

- **Atributos Complexos:** Este atributo pode ser tanto composto quanto multi-valorado. Ex.: Um atributo endereço da entidade Aluno pode ser composto por outros atributos, além disso um mesmo aluno pode ter diferentes endereços.

- **Valores nulos:** Caso uma entidade não tenha um valor aplicado a seu atributo ou não se tenha conhecimento desse valor, é utilizado o valor nulo.

1.4. Conjunto de Valores (Domínio de Valores): Os atributos de um tipo de entidade podem estar associados a um **domínio de valores**, nesse são especificados os valores que podem ser atribuídos a um atributo para as diferentes instâncias de uma entidade.

> **Ex.:** Suponha que sua entidade possua um atributo nome, usualmente o **conjunto de valores** desse atributo é uma cadeia de caracteres, além disso é possível definir uma faixa de valores.

1.5. Atributo Chave (Chave): Atributos ou conjunto de atributos cujos valores são únicos para cada uma das instâncias de uma entidade com um mesmo tipo de entidade.

> **Ex.:** Utilizando os exemplos acima podemos inferir que a matrícula de um estudante é um atributo chave para as instâncias da entidade Estudante. Outro exemplo seria a combinação dos atributos nome_estudante+endereço podem ser chaves.

- **Superchave:** Semelhante a uma chave, ou seja, um conjunto de um ou mais atributos cujos valores são distintos para cada instância da entidade.

- **Chave Candidata:** Ocorre quando uma superchave não possui subconjuntos próprios de superchaves, ou seja, uma

superchave com um único atributo. Ex.: Identidade de uma entidade Aluno.

- **Chave Primária:** A chave candidata escolhida pelo projetista do banco de dados como a principal forma de identificação do tipo de entidade em questão. As chaves candidatas não escolhidas são denominadas **chaves alternativas** ou **chaves secundárias.** Vale ressaltar que a escolha de uma chave primária deve ser feita pela escolha do atributo que melhor representa uma instância daquele entidade dentro do contexto do banco.

1.6. Relacionamento: Um fato do mundo real representado por meio da associação entre entidades do banco de dados. As associações entre entidades pode ser divididos em:

- Tipos de Relacionamentos:

  i) Grau de um tipo de relacionamento: Os relacionamentos podem ser divididos de acordo com o número de tipos de entidades participantes.

  ii) Relacionamentos com atributos: Relacionamentos podem possuir atributos.

  iii) Relacionamento Recursivo: Ocorrem quando o relacionamento é entre entidades de um mesmo tipo. Ex.: A relação *ensinou* entre os professores seniors e novatos de uma instituição estudantil.

Vale ressaltar que o relacionamento entre entidades possuem restrições como:

- **Restrição de cardinalidade:** Define o número de instâncias de um relacionamento em que uma instância de uma entidade pode participar. Nesse caso, temos as seguintes restrições:

  i) 1:1 (um-para-um): Uma entidade de tipo A pode estar associada a apenas outra entidade de tipo B e vice-versa.

  ii) 1:N (um-para-muitos): Uma entidade de tipo A pode estar associada a várias entidades de tipo B (0 até N), no entanto uma entidade de B pode estar associada a uma única entidade

de A.

iii) M:N (muitos-para-muitos): Uma entidade de A pode estar associada a várias entidades de B (0 até N), assim como uma entidade de B pode estar associada a várias entidades de A (0 até M).

- **Restrição de participação:** Define se a existência de uma entidade depende da associação desta com outra entidade por meio de um relacionamento. Nesse caso, temos as seguintes restrições:

  i) Total (dependência de existência): Todas as instâncias do tipo de entidade **devem** participar de um relacionamento, portanto é possível deduzir que todas as entidades fracas possuem dependência total de outras entidades. Ex.: No relacionamento *ensina* entre uma entidade Professor e uma entidade Matéria é necessário pelo menos um professor para cada matéria.

  ii) Parcial: A participação das instâncias em um relacionamento não é obrigatória.

- **Restrição de Cardinalidade Mínima e Máxima:** Essa restrição envolve associar um par de números inteiros (min, max) a cada participação de um tipo de entidade E, formando um relacionamento onde 0<= min <= max e max >= 0.

  Nessa situação os valores min e max representam o número mínimo e máximo de de instâncias do relacionamento em questão em que uma entidade de tipo E deve participar.

  Ao utilizar esse método temos que min = 0 representa uma participação parcial enquanto que min > 0 indica uma participação total.

2. Modelo ERE:

Em certas situações os bancos de dados não poderão ser modelados utilizando os conceitos básicos do modelo ER, nesse sentido é utilizado o modelo de entidade relacionamento estendido (ERE), o qual possui extensões ao modelo básico, tais como:

2.1. Herança: Em um BD podem existir subclasses, ou seja, subconjuntos de uma mesma entidade/superclasse que herdam suas propriedades e relacionamentos. Ex.: Superclasse Aluno que pode ser dividido em Graduando, Doutorando, Mestrando e etc.

Vale ressaltar que nem sempre uma entidade de uma superclasse necessita ser uma subclasse, no entanto uma instância de uma entidade não pode existir unicamente como membro de uma subclasse, essa deve ser membro da superclasse.

Quanto à relação de herança é possível que uma subclasse descenda de mais de uma subclasse, sendo denominado uma **herança múltipla.**

2.2. Especialização: É o processo de determinar um conjunto de subclasses a partir de uma entidade específica.

2.3. Generalização: A **Generalização** ocorre em oposição a **Especialização**, visto que busca as semelhanças entre diversos tipos de entidade com o intuito de definir uma superclasse da qual as entidades analisadas irão herdar propriedades.

Esse processo, no entanto deve atender algumas restrições impostas sobre ele que podem dos tipos:

- **Restrições de Disjunção:** Quanto às restrições de Disjunção, o desenvolvedor pode escolher entre dois tipos para uma entidade:

  i) Disjunção: Uma entidade da superclasse pode ser membro de apenas uma subclasse da especialização ou generalização.

  ii) Sobreposição: Uma entidade da superclasse pode ser membro de mais de uma subclasse da especialização ou generalização.

- **Restrições de Completude:** Quanto às restrições de Completude, o desenvolvedor pode escolher entre dois tipos para uma entidade:

  i) Total: Toda entidade da superclasse pode ser membro de no mínimo uma subclasse da especialização ou generalização.

  ii) Parcial: Nem toda entidade da superclasse deve ser membro de alguma subclasse da especialização ou generalização.

2.4. Tipo União ou Categoria: Uma subclasse de **tipo união ou categoria** ocorre quando é necessária a modelagem de uma única relação superclasse/subclasse com diversas superclasses, nessa situação as superclasses representam tipos de entidades diferentes, enquanto a subclasse representa a união de instâncias das superclasses (subconjunto da união).

- **Total:** Ocorre quando todas as instâncias da superclasse que se encaixam na subclasse.

- **Parcial:** Ocorre quando instâncias é possível haver instâncias da superclasse que não se encaixam na subclasse.

De posse do conceito de **tipo união ou categoria e de herança múltipla** é possível diferenciá-los, visto que em uma herança múltipla uma entidade da subclasse deve existir em todas as suas superclasses, enquanto que em uma categoria uma entidade na subclasse deve existir somente em uma de suas superclasses. Além disso, os atributos das entidades são diferentes dado que em uma herança múltipla a entidade herda todos os atributos de suas superclasses, enquanto que na categoria a entidade herda apenas os atributos da superclasse a qual pertence.

2.5. Relacionamento Ternário: Um relacionamento ternário (tipo de relacionamento de grau 3) ocorre quando há 3 tipos de entidades participantes.

2.6. Agregação: Abstração utilizada na construção de objetos compostos, ou seja, corresponde a um tipo de relacionamento que devido aos requisitos de modelagem necessita ser transformado em um tipo de entidade que se relaciona com outras entidades.