

REVISÃO DE BANCO DE DADOS AV1

Banco de Dados (Aula 1)

- Um banco de dados é um local para armazenar e recuperar informações. No modelo relacional, ele organiza dados em linhas (registros) e colunas (atributos). A principal diferença entre os bancos de dados relacionais e outros modelos está na forma como esses relacionamentos entre as informações são estruturados.

Modelo Relacional

- Um banco de dados relacional segue o formato de linhas e colunas. Cada linha contém uma instância de dados (registro), enquanto as colunas representam os atributos desses dados. É importante planejar a função principal do sistema antes de construí-lo para garantir que atenda às necessidades.

Esses conceitos são fundamentais para a compreensão de como os dados são armazenados, organizados e visualizados em um sistema de banco de dados, facilitando o gerenciamento e a recuperação eficiente de informações.

Abstração de Dados

1. Nível Físico :

- Trata da infraestrutura e do armazenamento real dos dados, chegando ao nível de granularidade como bytes e páginas de dados. Quanto mais dados, maior a necessidade de espaço de armazenamento. Exemplo: armazenamento por ano, mês, semana ou dia.

2. Nível Lógico :

- Define como os dados estão organizados e seus relacionamentos. Cria uma interface a partir da infraestrutura de dados para torná-los visíveis e acessíveis. Muito utilizado por administradores de banco de dados para gerenciar e manipular estruturas.

3. Nível de Visão :

- Permitir ao usuário final visualizar e interagir com os dados, limitando ou filtrando o acesso conforme necessário. Esse nível estabelece regras para cada tipo de usuário, mostrando apenas as informações relevantes para cada um.

Modelos de Dados (Estrutura Lógica)

- **Modelo Relacional:**
 - Base da maioria dos sistemas que utilizamos hoje. Evolui continuamente, mas preserva sua estrutura principal: tabelas compostas por linhas (registros) e colunas (atributos). A "relação" no modelo é entre as linhas e colunas. Cada linha na tabela representa um relacionamento entre os valores.
 - **Tupla:** Conjunto de valores de uma linha.
 - **Esquema do Banco de Dados:** Define os atributos que serão usados (ex: nome, endereço, etc.).
 - **Um dado é válido em um momento específico.** Exemplo: um carro pode ter vários donos ao longo do tempo, mas cada propriedade é válida em momentos distintos. O atributo mais importante nesse contexto é a **data**.
- **Esquema de Banco de Dados:**
 - Conjunto de atributos que descrevem uma entidade. Exemplo: um supermercado tem como atributos o nome, endereço, etc.
- **Modelos Alternativos:**
 - **Modelo de Rede:** Estrutura em forma de rede, onde os dados não possuem ligação direta e um atributo pode se conectar a outros.
 - **Modelo Hierárquico:** Semelhante ao modelo de rede, mas organiza os dados em uma estrutura de árvores hierárquicas em vez de grafos. Exemplo: Um registro pai pode ter vários registros filhos, mas cada filho tem apenas um pai, o que limita um pouco a flexibilidade, mas facilita a navegação entre os dados.
 - **Modelo Orientado a Objetos:** Representa os dados como objetos, similar à programação orientada a objetos, combinando dados e métodos de manipulação dentro de uma estrutura única.

Modelos Entidade Relacionamento

- No Modelo Entidade-Relacionamento (MER), as entidades são utilizadas para representar objetos do mundo real que possuem relevância para o sistema. Esses objetos podem ser concretos (como "Aluno", "Professor", "Produto") ou abstratos (como "Pedido", "Contrato", "Curso").
- Um conjunto de entidades agrupa todas as entidades que pertencem ao mesmo tipo e compartilham as mesmas características (atributos). Por exemplo, um conjunto de entidades "Aluno" pode conter entidades individuais como "João", "Maria", "Carlos", que têm atributos em comum, como "nome", "matrícula" e "data de nascimento".
- Cada entidade do conjunto terá valores específicos para esses atributos, e o conjunto de entidades ajuda a organizar e estruturar o banco de dados ao

representar esses objetos do mundo real de forma clara e eficiente.

- Todas estas informações que identificam e de certa forma, constroem o objeto, são conhecidas como atributos do objeto.
- Por consequência, todas as entidades possuem uma série de atributos que oferecem informações sobre o objeto que está representando. Desta forma, podemos pensar em uma entidade como um conjunto de atributos.
- Podemos representar a entidade **funcionário** através de uma série de atributos:
Funcionário (nome, data de nascimento, matrícula, cpf, identidade, sexo, nr filhos, fone, endereço, bairro, cidade, cep).

ATRIBUTOS

- **Simples:** Não podem ser divididos. Ex: sexo_funcionário.
- **Compostos:** Podem ser divididos em novos atributos. Ex: endereço_cliente → rua, número, cidade, CEP.
- **Monovalorados:** Aceitam apenas um valor por entidade. Ex: matrícula_funcionário.
- **Multivalorados:** Aceitam múltiplos valores para uma entidade. Ex: fone_cliente.
- **Nulos:** Podem aceitar valores nulos. Ex: email_cliente.

Cardinalidade no Modelo Entidade-Relacionamento (MER) define o número de instâncias de uma entidade que podem estar associadas a instâncias de outra entidade em um relacionamento. Ela ajuda a descrever como as entidades se relacionam entre si. Existem três principais tipos de cardinalidade:

- **Um para Um (1:1):** Cada instância de uma entidade A pode se relacionar com no máximo uma instância de uma entidade B, e vice-versa. Exemplo: Cada pessoa possui um número de CPF, e cada CPF pertence a uma única pessoa.
- **Um para Muitos (1:n):** Uma instância da entidade A pode estar associada a várias instâncias da entidade B, mas cada instância de B só pode estar associada a uma instância de A. Exemplo: Um professor pode dar aula para vários alunos, mas cada aluno tem apenas um professor responsável.
- **Muitos para Muitos (N:N):** Instâncias de ambas as entidades A e B podem estar associadas a múltiplas instâncias umas das outras. Exemplo: Um aluno pode estar matriculado em vários cursos, e cada curso pode ter vários alunos.

Cardinalidades (Exemplo)

- **Um funcionário pode estar vinculado a apenas um departamento:** neste caso, para uma ocorrência de funcionário existirá apenas uma ocorrência

possível de departamento (**um**), entretanto, para uma única ocorrência de departamento poderemos ter várias ocorrências de funcionários (**muitos**).

Neste caso a cardinalidade entre funcionários e departamentos é do tipo muitos para um. Geralmente, neste tipo de associação um atributo da entidade que identifique o lado “um” da associação é posicionado na entidade que representa o lado “muitos”.

TIPOS DE CHAVES

- **Chaves Candidatas:** As chaves candidatas são um conjunto de atributos de uma determinada entidade que garantem a identificação única de uma ocorrência daquela entidade.
- No exemplo utilizado neste material vamos analisar o conjunto de atributos cpf e nome da entidade funcionários, este conjunto (cpf, nome) não pode ser considerado uma chave candidata, pois apesar de garantir que não existirão duas tuplas com os mesmos valores daquela combinação (cpf, nome) – propriedade da unicidade -, um subconjunto (cpf) da combinação inicial também apresentará a propriedade da unicidade.
Desta forma esta combinação não atende os dois princípios necessários para a sua classificação como chave candidata. Já o conjunto dos atributos (nome, pai, mãe, dt_nascimento) poderá ser classificado como chave candidata, pois atende às duas propriedades necessárias.
- **Chaves Primária:** A chave primária poderá ser escolhida a partir do conjunto de chaves candidatas possíveis para aquela entidade, da mesma forma que as chaves candidatas representam a identificação exclusiva das tuplas daquela entidade. É bom lembrar que uma chave primária representa um valor único e NÃO NULO.
- **Chaves Estrangeira:** Uma chave estrangeira é um conjunto de atributos de uma entidade **E1** cujos valores devem corresponder a valores de alguma chave candidata de outra entidade **E2**.

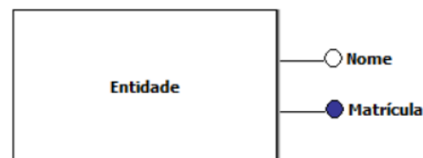
Diagrama Entidade Relacionamento

- Cardinalidade: mínima e máxima
- Mostra a quantidade de vezes que uma entidade referencia a outra. Ou seja, para um ocorrência da entidade A, quantas ocorrências de B podem existir.



Diagrama Entidade Relacionamento

- Atributos
- São representados por uma linha e um círculo, podendo diferenciar os atributos das chaves que podem ser círculos pintados.



Linguagens de Banco de Dados

- Modelo Entidade / Relacionamento
- Retângulos: entidades
- Elipses: atributos
- Losangos: relações
- Linhas: ligação entre atributos, conjuntos, relações e entidades.

NORMALIZAÇÃO

Formas Normais

Primeira Forma Normal – 1FN: A tabela não deve possuir grupos de repetição.

Exemplo (Aluno, Pai, Mãe)

João, Carlos, Maria

Marcos, Carlos, Maria

Segunda Forma Normal – 2FN: Todos os atributos dependem totalmente da chave primária, os que não dependem vão para outra tabela.

Exemplo (Aluno, Nome do Curso, Código do Curso)

João, Computação, 1

Paulo, Computação, 1

Terceira Forma Normal – 3FN: Não podem existir atributos que dependam de outros atributos.

Exemplo (Aluno, Código do Curso, Nome do Curso)

Propriedades ACID

- Todo banco de dados para ser considerado relacional, precisa obedecer as propriedades ACID.
- **Propriedades ACID:**
 - Atomicidade
 - Consistência
 - Independência ou Isolamento
 - Durabilidade
- **Atomicidade** (Atomicity):
 - **Tudo ou nada:** As transações devem ser realizadas de forma completa ou não realizadas de modo algum. Se uma parte da transação falhar, todas as mudanças feitas até aquele ponto são desfeitas, retornando o sistema ao estado anterior. Exemplo: Se um depósito é feito em uma conta, mas o sistema falha antes de atualizar o saldo, o depósito é revertido.
- **Consistência** (Consistency):
 - A transação deve levar o banco de dados de um estado válido para outro estado válido, mantendo as regras e restrições de integridade dos dados. Ou seja, os dados devem permanecer corretos e seguir as regras do sistema. Exemplo: Uma transação que subtraia fundos de uma conta não pode deixar o saldo negativo, se essa for uma regra do banco de dados.
- **Isolamento** (Isolation):
 - Transações simultâneas devem ser executadas de forma que seus efeitos sejam independentes umas das outras, como se fossem processadas de forma sequencial. Mesmo quando várias transações ocorrem ao mesmo tempo, os resultados são os mesmos como se tivessem sido processadas uma após a outra. Exemplo: Se dois usuários tentam fazer compras ao mesmo tempo, cada transação deve ser isolada para que não haja interferência nos valores finais.

- **Durabilidade** (Durability):
 - Após uma transação ser confirmada (committed), suas alterações no banco de dados devem ser permanentes, mesmo em caso de falhas do sistema, como quedas de energia. O banco de dados usa técnicas como logs ou backups para garantir que os dados sejam recuperáveis. Exemplo: Após uma compra ser concluída com sucesso, o registro da compra não será perdido, mesmo que o sistema caia logo depois.