

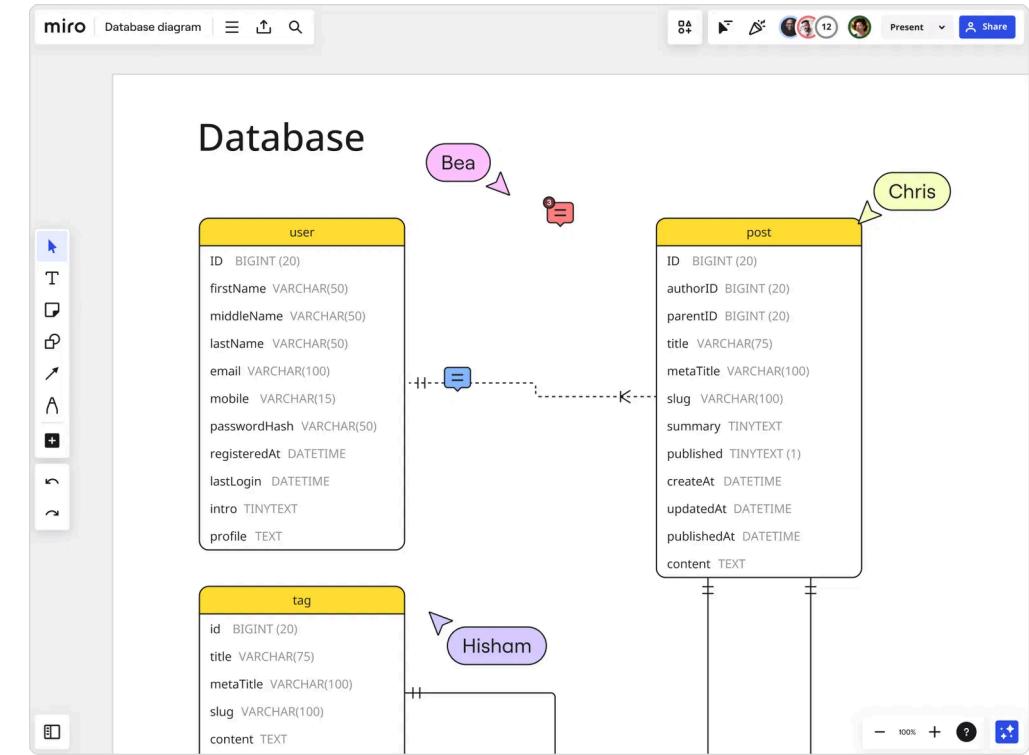
Projeto Integrador

Modelagem do Banco de Dados



Objetivos da Aula

- Apresentar os conceitos e os tipos de modelagem de dados (relacional e não relacional), compreendendo suas diferenças e aplicações.
- ☒ Capacitar os alunos a utilizar ferramentas de modelagem como Draw.io e Lucidchart para criar diagramas profissionais de banco de dados.
- 🕒 Orientar as equipes na elaboração do **Modelo Entidade-Relacionamento (MER)** ou de uma estrutura de dados não relacional, com base nos requisitos levantados na aula anterior.
- 💬 Incentivar a discussão e a tomada de decisão sobre a melhor modelagem para cada projeto, considerando as necessidades específicas identificadas.



💡 A modelagem de dados é a base para um sistema de informação eficiente e escalável

Revisão e Conexão com a Aula Anterior

Aula Anterior: Requisitos e Personas

Na aula anterior, trabalhamos com:

Requisitos Funcionais: O que o sistema deve fazer

Requisitos Não Funcionais: Como o sistema deve operar

Personas: Representações dos usuários finais



Aula Atual: Modelagem do Banco de Dados

Hoje vamos traduzir esses requisitos em uma estrutura de banco de dados que suporte as funcionalidades identificadas.

 **A modelagem de dados é a ponte entre os requisitos e sua implementação técnica.**

Uma modelagem bem feita garante:

Integridade: Dados consistentes

Eficiência: Operações otimizadas

Escalabilidade: Capacidade de crescimento

REQUISITOS NÃO FUNCIONAIS



REQUISITOS FUNCIONAIS QUE COMPÕEM AS CARACTERÍSTICAS QUE O SISTEMA DEVE RESPEITAR AO ATENDER CADA UM DOS REQUISITOS FUNCIONAIS SOLICITADOS.



Atividade Inicial

Cada equipe fará um resumo rápido (2 minutos) do seu projeto e das personas criadas na aula anterior.

 Foque nos requisitos que impactam o banco de dados.

Modelagem de Dados: Conceitos Fundamentais

O que é Modelagem de Dados?

É o processo de criar uma representação visual da estrutura de um banco de dados, definindo como os dados serão armazenados e relacionados entre si.

“ A modelagem de dados é para o banco de dados o que a planta é para a construção de um edifício.

Níveis de Modelagem

Modelagem Conceitual

Visão de alto nível

Independente de tecnologia

Modelagem Lógica

Representação no SGBD

Específica para um tipo de banco

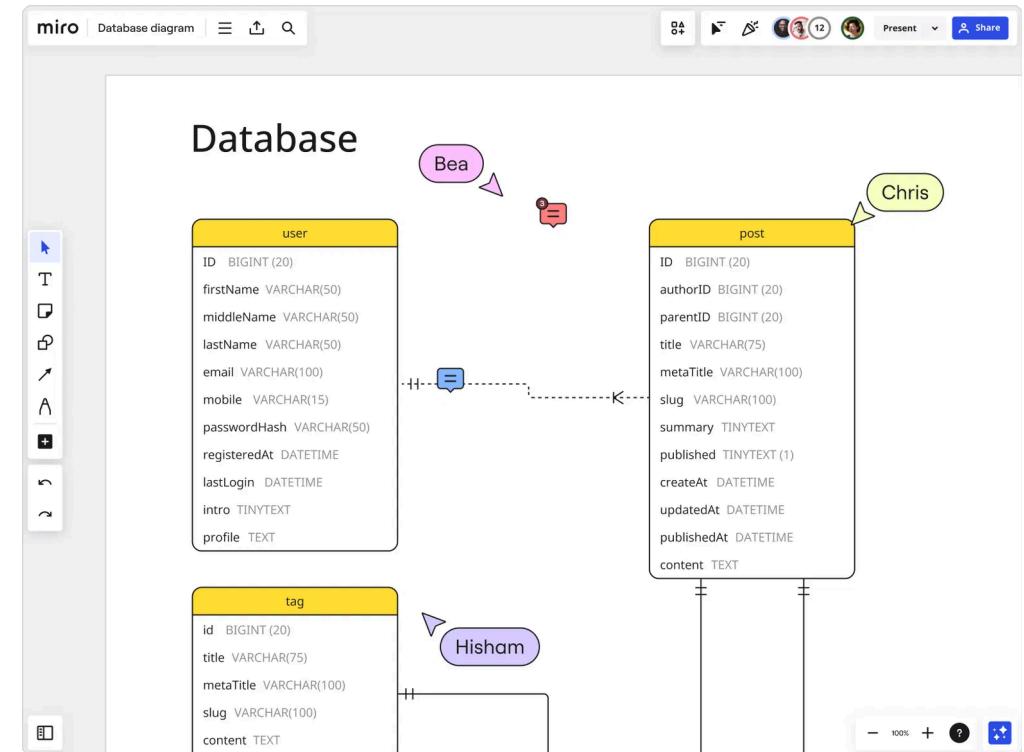
Foco: Entidades e relacionamentos principais

Público: Stakeholders e usuários

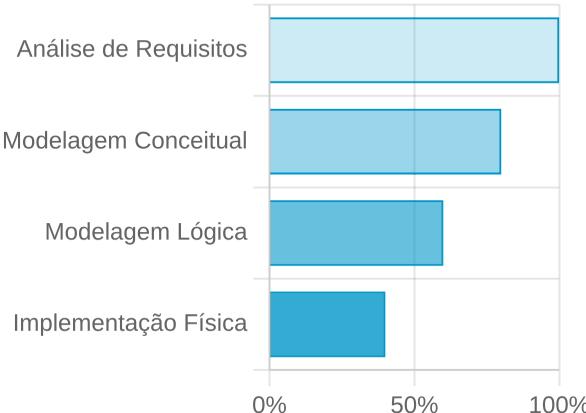
Foco: Tabelas, colunas, chaves, índices

Público: Desenvolvedores e DBAs

Nesta aula, nosso foco será a modelagem lógica, que é a base para a implementação do banco de dados.



Etapas do Processo de Modelagem



Modelo Relacional (MER)

O **Modelo Entidade-Relacionamento (MER)** é uma forma de representar a estrutura lógica de um banco de dados relacional, mostrando como os dados se relacionam entre si.

Entidade

Objeto do mundo real identificado de forma única (Ex: Cliente, Produto, Pedido).

Atributo

Características que descrevem uma entidade (Ex: nome, endereço, preço).

Relacionamento

Associação entre entidades (Ex: Cliente faz Pedido, Produto pertence a Categoria).

Cardinalidade

1:1 (um para um)

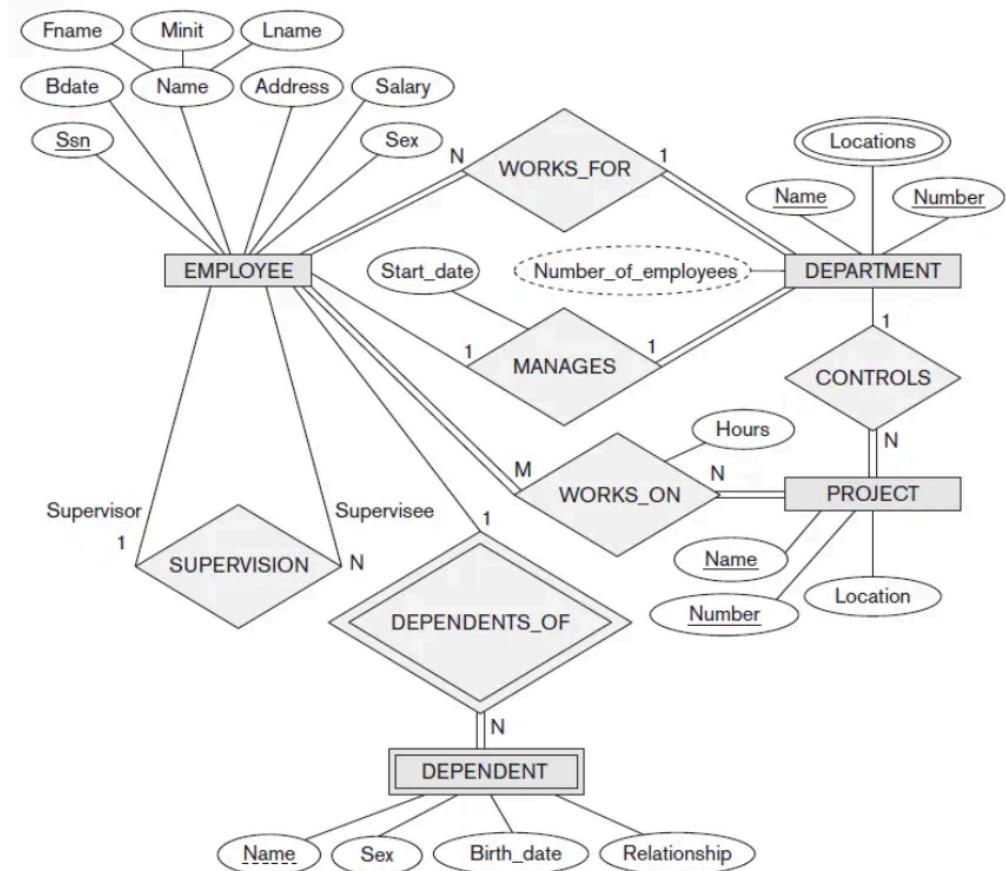
Um registro se relaciona com apenas um outro
Ex: Pessoa ↔ CPF

1:N (um para muitos)

Um registro se relaciona com vários outros
Ex: Cliente → Pedidos

N:N (muitos para muitos)

Vários registros se relacionam com vários outros
Ex: Alunos ↔ Disciplinas



Vantagens do Modelo Relacional

- Consistência e integridade dos dados
- Estrutura bem definida e previsível
- Suporte a transações ACID
- Ideal para sistemas com relacionamentos complexos

Modelo Não Relacional (NoSQL)

O que são Bancos NoSQL?

Bancos de dados **NoSQL** (Not Only SQL) são projetados para armazenar, distribuir e acessar dados usando modelos diferentes do tradicional modelo relacional.

- 💡 **Características principais:** Flexibilidade de esquema, escalabilidade horizontal e alta disponibilidade.

Principais Tipos de NoSQL

Documentos

Armazena dados em documentos JSON (MongoDB)

Colunar

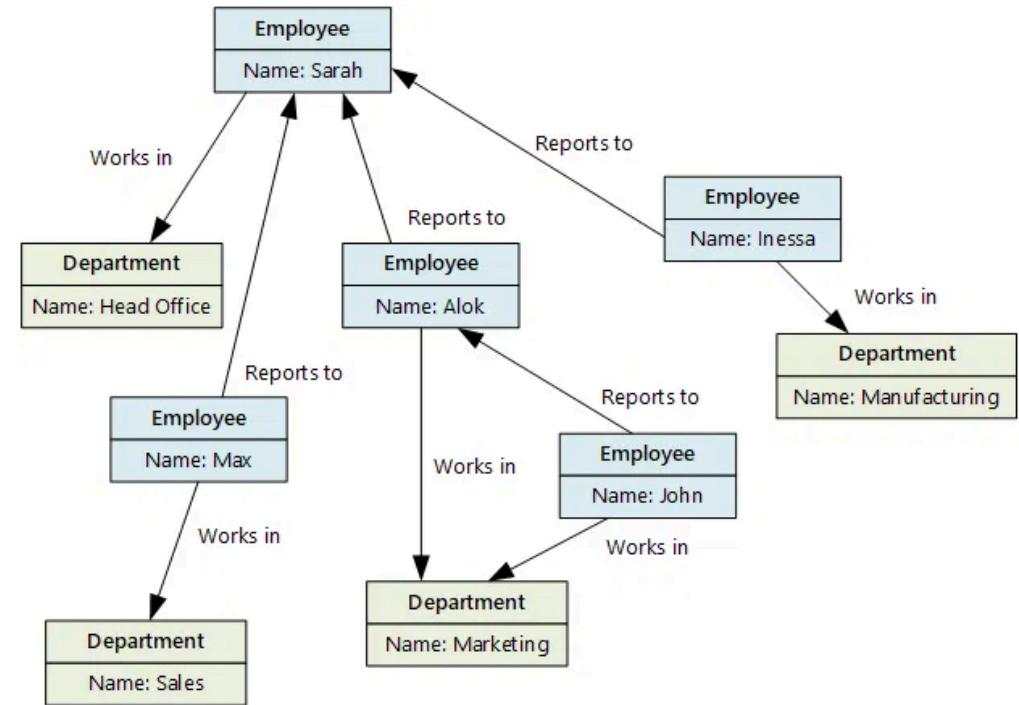
Armazena dados em colunas (Cassandra)

Vantagens do NoSQL

Flexibilidade: Esquema dinâmico que evolui com as necessidades

Escalabilidade: Facilidade para escalar horizontalmente

Desempenho: Otimizado para operações específicas



Exemplo: Documento JSON (MongoDB)

```
{  
  "_id": "5f8d0e1d2c3b4a5b6c7d8e9f",  
  "nome": "João Silva",  
  "email": "joao@email.com",  
  "telefones": [  
    {"tipo": "celular", "numero": "11 99999-8888"},  
    {"tipo": "trabalho", "numero": "11 3333-4444"}  
  ],  
  "endereco": {
```

Comparação: Relacional vs. Não Relacional

Característica	Modelo Relacional (SQL)	Modelo Não Relacional (NoSQL)
Estrutura	Tabelas com linhas e colunas	Varia (documentos, chave-valor, grafos, etc.)
Esquema	Rígido e predefinido	Flexível ou sem esquema
Escalabilidade	Vertical (hardware mais potente)	Horizontal (mais servidores)
Transações	ACID (Atomicidade, Consistência, Isolamento, Durabilidade)	Geralmente BASE (Basicamente Disponível, Estado Flexível, Eventualmente Consistente)
Consultas	SQL padronizado	API específica ou linguagem de consulta própria
Relacionamentos	Suporte nativo com chaves estrangeiras	Geralmente desnormalizados ou modelados na aplicação

Quando usar Relacional?

Dados estruturados e relacionamentos complexos

Sistemas com muitas entidades relacionadas (ERP, CRM)

Transações críticas

Sistemas financeiros, e-commerce, gestão de estoque

Quando usar Não Relacional?

Dados não estruturados ou semiestruturados

Redes sociais, logs, conteúdo gerado por usuários

Alta escalabilidade e desempenho

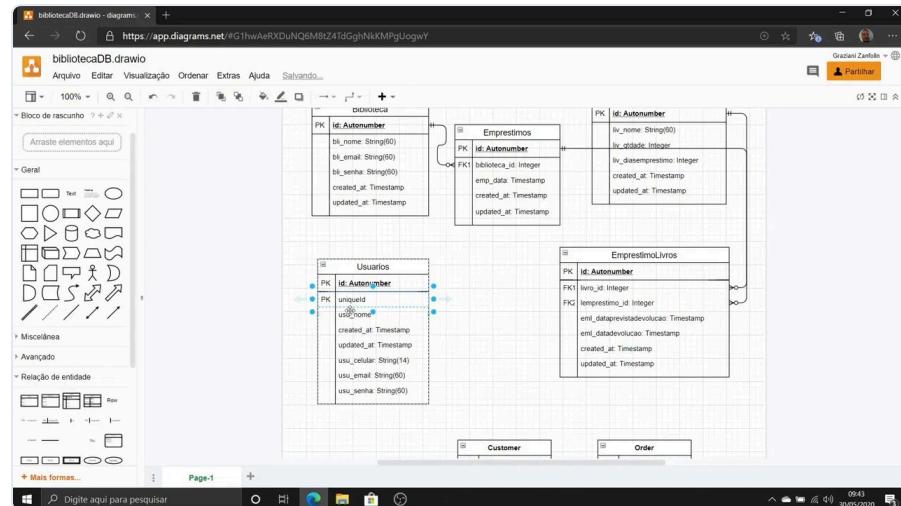
Aplicações com milhões de usuários, IoT, big data

 **Dica:** Muitos sistemas modernos utilizam abordagens híbridas, combinando bancos relacional e não relacional para diferentes partes da aplicação.

Ferramentas de Modelagem



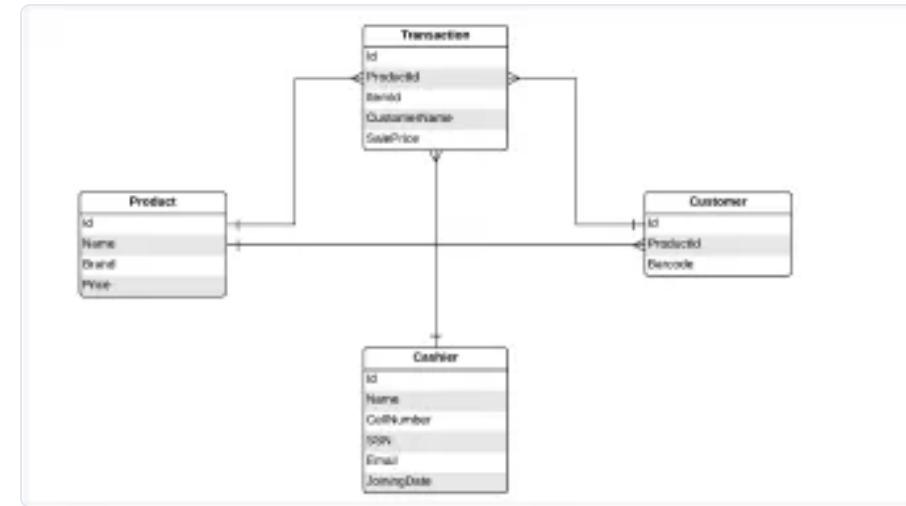
Ferramenta gratuita e de código aberto para criação de diagramas, incluindo modelos de banco de dados.



- ✓ Interface intuitiva com arrastar e soltar
- ✓ Versão web e desktop disponíveis
- ✓ Integração com Google Drive e OneDrive
- ✓ Biblioteca de formas para modelagem ER
- ✓ Exportação em diversos formatos



Plataforma profissional para criação de diagramas com recursos avançados para modelagem de dados.



- ✓ Colaboração em tempo real
- ✓ Modelos prontos para bancos relacionais e NoSQL
- ✓ Importação de esquemas de banco existentes
- ✓ Geração automática de SQL a partir do diagrama
- ✓ Versão gratuita com recursos limitados

[Acessar Draw.io](#)

[Acessar Lucidchart](#)

Atividade Prática em Equipe

👤 Nesta atividade, cada equipe irá escolher o tipo de modelagem mais adequado para seu projeto e começar a elaborar o modelo de dados.

1 Escolha do Tipo de Modelagem

Discutam em equipe: "Nosso sistema tem requisitos que se encaixam melhor em um banco de dados relacional ou não relacional?"

- ✓ **Relacional:** Estrutura rígida, muitas relações entre entidades
- ✓ **Não Relacional:** Dados flexíveis, alta escalabilidade

2 Elaboração da Modelagem

Com base na escolha anterior, iniciem a modelagem do banco de dados:

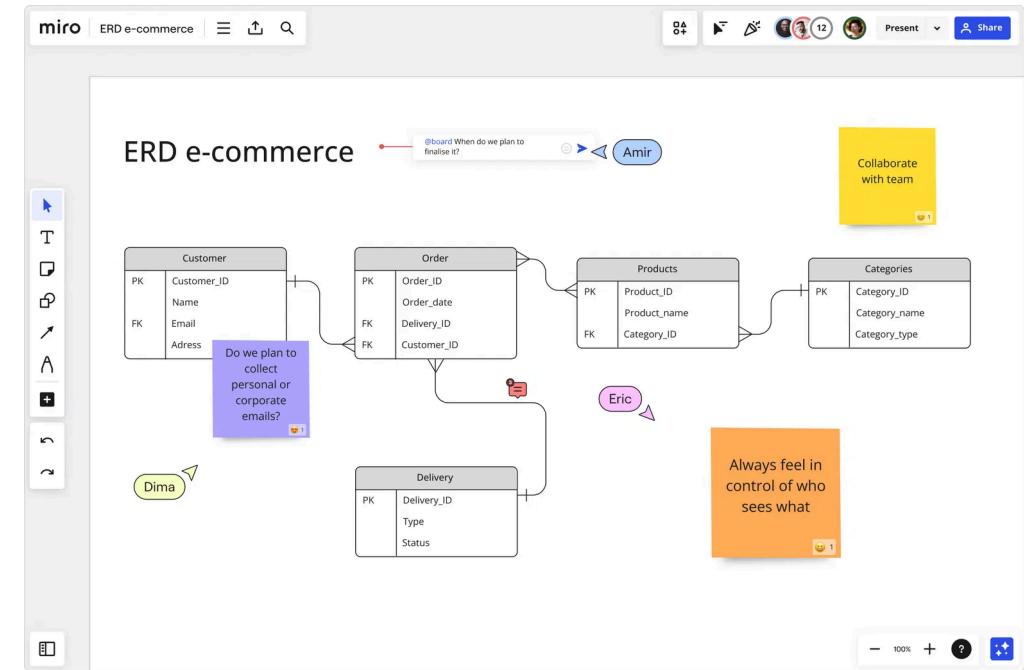
- ⌚ **Modelo Relacional:** Desenhar o MER completo (entidades, atributos, relacionamentos)
- ↔ **Modelo Não Relacional:** Definir coleções e estrutura de documentos

3 Uso das Ferramentas

Utilizem uma ferramenta de diagramação para formalizar o desenho do modelo:

- ✗ **Draw.io:** Ferramenta gratuita online ou desktop
- ✗ **Lucidchart:** Ferramenta online com recursos colaborativos

⌚ **Tempo para a atividade:** 45 minutos para elaboração da modelagem



Dicas para a Atividade

- Revisitem os requisitos levantados na aula anterior
- Identifiquem as principais entidades/coleções do sistema
- Pensem nos relacionamentos entre essas entidades
- Definam os atributos essenciais de cada entidade
- Justifiquem a escolha do modelo com base nos requisitos

Passo a Passo: Elaboração do MER

1 Identificar as Entidades

Analise os requisitos e identifique os principais objetos do sistema.

Exemplo: Em um sistema de biblioteca: Livro, Autor, Usuário, Empréstimo.

2 Definir os Atributos

Para cada entidade, liste as características relevantes:

- Atributos identificadores (chaves primárias)
- Atributos descritivos e tipos de dados

Exemplo: Livro: ISBN (chave), título, ano_publicacao, editora.

3 Estabelecer os Relacionamentos

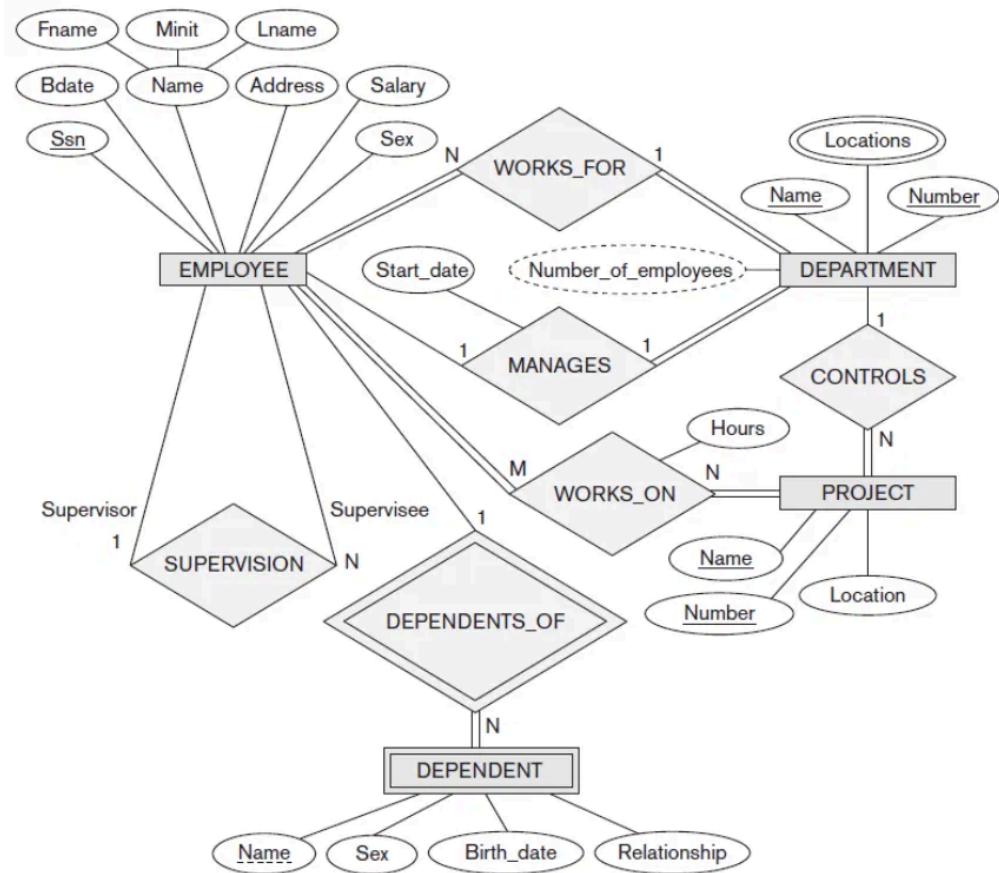
Identifique como as entidades se relacionam e defina a cardinalidade.

Exemplo: Livro pertence a Categoria (N:1), Livro escrito por Autor (N:N).

4 Refinar o Modelo

Revise o modelo para garantir que atenda aos requisitos:

- Normalize as tabelas para evitar redundância
- Verifique a integridade referencial



Dicas para um bom MER

- Use nomes claros para entidades e relacionamentos
- Evite redundância de dados entre tabelas
- Defina corretamente as chaves primárias e estrangeiras
- Valide o modelo com os requisitos do sistema

Passo a Passo: Estrutura Não Relacional

1 Identificar as Coleções

Determine os principais grupos de dados que seu sistema precisa armazenar.

Exemplo: Em um e-commerce, as coleções podem ser: produtos, usuários, pedidos, avaliações.

2 Definir a Estrutura dos Documentos

Para cada coleção, defina a estrutura interna dos documentos, incluindo:

- Campos obrigatórios
- Tipos de dados
- Documentos aninhados
- Arrays de valores ou subdocumentos

3 Decidir entre Incorporação ou Referência

Determine como relacionar dados entre coleções:

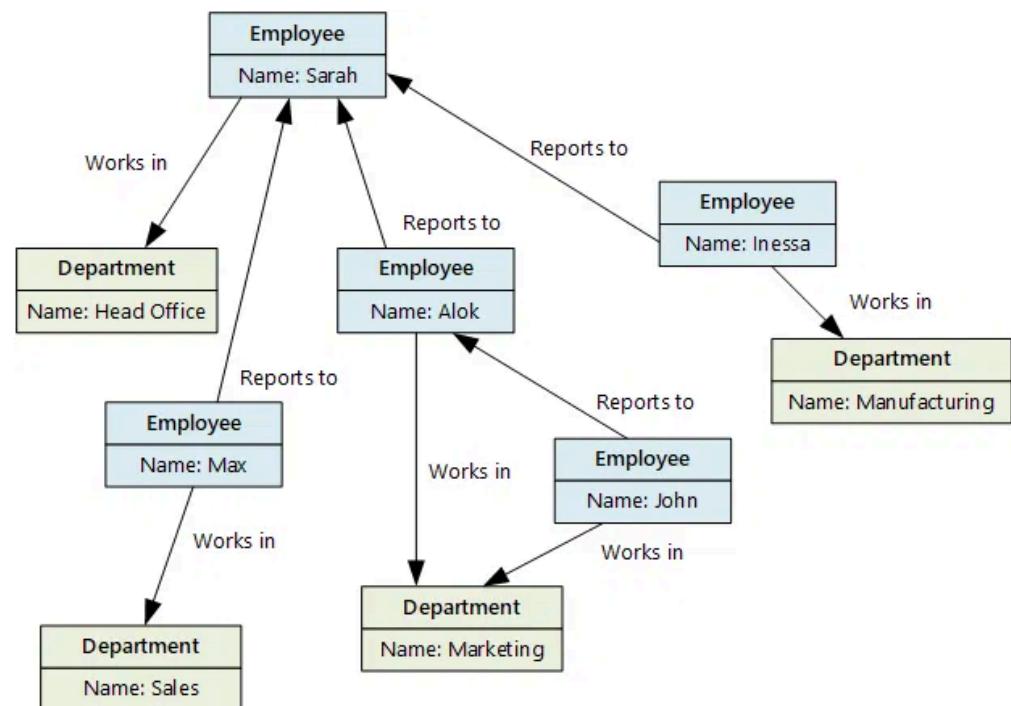
- **Incorporação:** Incluir documentos dentro de outros (1:1, 1:N com poucos itens)
- **Referência:** Armazenar IDs que apontam para outros documentos (N:N, 1:N com muitos itens)

Exemplo: Um usuário pode ter seus endereços **incorporados** (poucos), mas seus pedidos seriam **referenciados** (muitos).

4 Otimizar para Padrões de Acesso

Adapte a estrutura com base em como os dados serão consultados:

- Priorize os casos de uso mais frequentes
- Considere a desnormalização para melhorar a performance
- Planeje índices para consultas comuns



Dicas para Modelagem NoSQL

- Modele para as consultas, não para a normalização
- Evite documentos muito grandes (limite de tamanho)
- Considere a frequência de atualizações ao decidir incorporar
- Use nomes de campos consistentes entre coleções
- Documente o esquema mesmo sendo flexível

Exemplo: Estrutura de Documento

```
{  
  "_id": "u123",  
  "nome": "Ana Silva",  
  "email": "ana@email.com",  
  "enderecos": [  
    {  
      "tipo": "residencial",  
      "rua": "Rua das Flores, 123",  
      "cidade": "São Paulo"  
    },  
    {"pedidos_ids": ["p001", "p002"]}  
  ]}
```

Apresentação e Feedback

Discussão em Grupo

Cada equipe terá a oportunidade de apresentar a modelagem que criou, explicando suas escolhas e decisões de design.

Roteiro de Apresentação

1. Apresente brevemente o projeto e seus requisitos principais
2. Explique por que escolheram o modelo relacional ou não relacional
3. Mostre as entidades/coleções principais e seus atributos
4. Destaque os relacionamentos mais importantes
5. Mencione desafios encontrados e como foram resolvidos



Critérios de Feedback

- ✓ **Adequação ao problema:** O modelo atende aos requisitos levantados?
- ✓ **Completude:** Todas as entidades/coleções necessárias estão presentes?
- ✓ **Relacionamentos:** As conexões entre os dados estão bem definidas?
- ✓ **Normalização:** O modelo evita redundâncias e inconsistências?
- ✓ **Justificativa técnica:** As escolhas de modelagem são bem fundamentadas?

Tempo de Apresentação

5:00

minutos por equipe

Dicas para o Feedback

- Seja construtivo e específico
- Comece com pontos positivos
- Sugira melhorias concretas
- Faça perguntas para esclarecer dúvidas
- Relacione o feedback aos requisitos do projeto

Próximos Passos

Implementação do Banco de Dados na Nuvem

Na próxima etapa do projeto, vamos implementar o banco de dados na nuvem utilizando o modelo que foi desenhado hoje.

Lembre-se: O modelo pode ser ajustado à medida que a implementação avança, mas essa etapa inicial de modelagem é fundamental para o sucesso do projeto.

Escolha da Plataforma de Nuvem

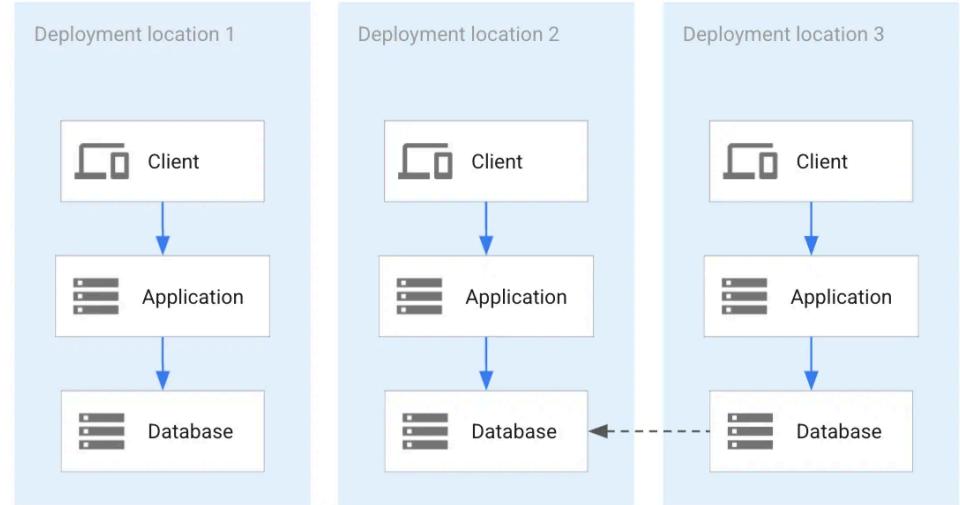
Seleção da plataforma mais adequada para hospedar o banco de dados (AWS, Azure, Google Cloud).

Configuração do Banco de Dados

Criação e configuração do banco de dados de acordo com o modelo desenvolvido.

Desenvolvimento da Camada de Acesso

Implementação da camada de acesso aos dados na aplicação.



Cronograma do Projeto

- 1 **Aula Anterior:** Levantamento de Requisitos e Personas
- 2 **Aula Atual:** Modelagem do Banco de Dados
- 3 **Próxima Aula:** Implementação do Banco de Dados na Nuvem
- 4 **Aulas Futuras:** Desenvolvimento da Aplicação e Testes