



INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Bahia

Banco de Dados

SGBD Relacional – Normalização



Roteiro

- SGBD Relacional – Normalização:
 - O que é Normalização?
 - Objetivos da Normalização;
 - Formas Normais:
 - Primeira Forma Normal (1FN);
 - Segunda Forma Normal (2FN);
 - Terceira Forma Normal (3FN);
 - Atividade.



O que é Normalização?

A **normalização** é o processo de organização dos dados de um banco de dados para reduzir redundâncias e evitar anomalias de inserção, exclusão e alteração. Isso melhora a integridade dos dados e facilita a manutenção.



Objetivos da Normalização

- **Minimizar a duplicação** de dados;
- **Facilitar atualizações e consultas** no banco de dados;
- **Eliminar anomalias** como inconsistências em inserções, exclusões ou alterações.



Formas Normais

- **Primeira Forma Normal (1FN);**
- **Segunda Forma Normal (2FN);**
- **Terceira Forma Normal (3FN);**
- **...**



Primeira Forma Normal (1FN)

- Cada **coluna** deve conter valores atômicos (sem listas ou grupos de valores).
- **Não** deve haver atributos multivalorados ou **repetidos**.



Primeira Forma Normal (1FN)

Exemplo de problema:

```
CREATE TABLE Venda (  
  id INT NOT NULL,  
  data DATE NOT NULL,  
  itens_vendidos VARCHAR(255) NOT NULL,  
  PRIMARY KEY(id)  
);
```

O campo **itens_vendidos** armazena **múltiplos itens**,
violando a 1FN.



Primeira Forma Normal (1FN)

Solução:

```
CREATE TABLE Venda (  
    id INT NOT NULL AUTO_INCREMENT,  
    data DATE NOT NULL,  
    PRIMARY KEY (id)  
);  
  
CREATE TABLE Item_Venda (  
    id INT NOT NULL,  
    venda_id INT NOT NULL,  
    quantidade INT NOT NULL,  
    PRIMARY KEY (id, venda_id),  
    FOREIGN KEY (venda_id) REFERENCES Venda(id)  
);
```



Segunda Forma Normal (2FN)

- **Atende à 1FN.**
- **Todos os atributos não chave devem depender da chave primária (elimina dependências parciais).**



Segunda Forma Normal (2FN)

Exemplo de problema:

```
CREATE TABLE Notas (  
    aluno_id INT NOT NULL,  
    disciplina_id INT NOT NULL,  
    aluno_nome VARCHAR(100),  
    nota DECIMAL(5,2),  
    PRIMARY KEY (aluno_id, disciplina_id)  
);
```

O **aluno_nome** depende apenas de **aluno_id**, causando uma dependência parcial.



Segunda Forma Normal (2FN)

Solução:

```
CREATE TABLE Aluno (  
    id INT NOT NULL AUTO_INCREMENT,  
    nome VARCHAR(100) NOT NULL,  
    PRIMARY KEY (id)  
);  
  
CREATE TABLE Notas (  
    aluno_id INT NOT NULL,  
    disciplina_id INT NOT NULL,  
    nota DECIMAL(5,2) NOT NULL,  
    PRIMARY KEY (aluno_id, disciplina_id),  
    FOREIGN KEY (aluno_id) REFERENCES Aluno(id),  
    FOREIGN KEY (disciplina_id) REFERENCES Disciplina(id)  
);
```



Terceira Forma Normal (3FN)

- **Atende à 2FN.**
- **Elimina dependências transitivas: nenhum atributo não chave deve depender de outro atributo não chave.**



Terceira Forma Normal (3FN)

Exemplo de problema:

```
CREATE TABLE Funcionario (  
  id INT NOT NULL AUTO_INCREMENT,  
  nome VARCHAR(100) NOT NULL,  
  departamento VARCHAR(100) NOT NULL,  
  gerente_departamento VARCHAR(100) NOT NULL,  
  PRIMARY KEY (id)  
);
```

O **gerente_departamento** depende do **departamento**, e **não** diretamente do **id** do **funcionário**, criando uma dependência transitiva.



Terceira Forma Normal (3FN)

Solução:

```
CREATE TABLE Funcionario (  
    id INT NOT NULL AUTO_INCREMENT,  
    nome VARCHAR(100) NOT NULL,  
    departamento_id INT NOT NULL,  
    PRIMARY KEY (id),  
    FOREIGN KEY (departamento_id)  
    REFERENCES Departamento(id)  
);
```

```
CREATE TABLE Departamento (  
    id INT NOT NULL AUTO_INCREMENT,  
    nome VARCHAR(100) NOT NULL,  
    gerente VARCHAR(100) NOT NULL,  
    PRIMARY KEY (id)  
);
```



Atividade

1. Normalize a seguinte tabela até a 3FN:

```
CREATE TABLE Pedido (  
    id INT NOT NULL AUTO_INCREMENT,  
    cliente_nome VARCHAR(100) NOT NULL,  
    cliente_endereco VARCHAR(255) NOT NULL,  
    produto_id INT NOT NULL,  
    produto_nome VARCHAR(100) NOT NULL,  
    quantidade INT NOT NULL,  
    preco_total DECIMAL(10,2) NOT NULL,  
    PRIMARY KEY (id)  
);
```



Atividade - Resposta

```
CREATE TABLE Cliente (  
    id INT NOT NULL AUTO_INCREMENT,  
    nome VARCHAR(100) NOT NULL,  
    endereco VARCHAR(255) NOT NULL,  
    PRIMARY KEY (id)  
);
```

```
CREATE TABLE Produto (  
    id INT NOT NULL AUTO_INCREMENT,  
    nome VARCHAR(100) NOT NULL,  
    PRIMARY KEY (id)  
);
```



Atividade - Resposta

```
CREATE TABLE Pedido (  
    id INT NOT NULL AUTO_INCREMENT,  
    cliente_id INT NOT NULL,  
    preco_total DECIMAL(10,2) NOT NULL,  
    PRIMARY KEY (id),  
    FOREIGN KEY (cliente_id) REFERENCES Cliente(id)  
);
```



Atividade - Resposta

```
CREATE TABLE Pedido_Produto (  
    pedido_id INT NOT NULL,  
    produto_id INT NOT NULL,  
    quantidade INT NOT NULL,  
    PRIMARY KEY (pedido_id, produto_id),  
    FOREIGN KEY (pedido_id) REFERENCES Pedido(id),  
    FOREIGN KEY (produto_id) REFERENCES Produto(id)  
);
```



Obrigado!

Questões?

