

RELATÓRIO DE IDENTIFICAÇÃO E AJUSTE DAS TABELAS PARA ATIVIDADE UC4

Aluno: André Cararo Lowcke

Atividade

Observando as tabelas citadas no contexto e realizando as consultas no banco de dados, elabore um relatório respondendo aos seguintes pontos:

1. Por que a tabela "Cliente" não está na 1FN? Proponha uma solução para fazer sua normalização na 1FN.
2. Por que a tabela "Item_venda" não está na 2FN? Proponha uma solução para fazer sua normalização na 2FN.
3. Por que a tabela "venda" não está na 3FN? Proponha uma solução para fazer sua normalização na 3FN.
4. Crie o *script* SQL correspondente à alteração proposta com os comandos de criação e/ou alteração das tabelas normalizadas.

1. A tabela Cliente aparentemente segue a Primeira Forma Normal por quê cada coluna possui valores únicos e atômicos. Mas para o caso em que o cliente tenha mais de um telefone é apropriado criar uma tabela 'telefones_cliente':

Id	Nome	Cpf	Endereco	Telefone
1	André	74084269069	Avenida Aníbal Pereira 01,Centro,Afligid os,BA	(96) 2881-0817, (95) 2915-5787
2	Marcos	82864770067	Rua Anitápolis 283,Centro,Alfre do Wagner,SC	(69) 2512-2802

Tabela Normalizada

Tabela de 'clientes':

Id	Nome	Cpf	Logradouro	Bairro	Cidade	Estado
1	André	74084269069	Avenida Aníbal Pereira 01	Centro	Afligidos	BA
2	Marco s	82864770067	Rua Anitápolis 283	Centro	Alfredo Wagner	SC

Tabela de 'telefones_cliente':

IdTelefone	Telefone	cliente_id
1	(96) 2881-0817	1
2	(95) 2915-5787	1
3	(69) 2512-2802	2

2. Na tabela Item_venda para aplicar a Segunda Forma Normal (2FN), precisamos garantir que a tabela esteja em conformidade com a Primeira Forma Normal (1FN) e a tabela Item_venda não atende, e os atributos não-chave não são totalmente dependente da chave primária. Para normalizar a tabela para a 2FN, precisamos dividir a tabela em duas ou mais tabelas, de modo que todos os atributos não-chave dependam totalmente da chave primária.

Abaixo deixo um exemplo de como seria uma possível aplicação:

Quantidade	valor_unitario	Subtotal	Nome_produto	venda_id	produto_id
1	10,33	10,33	bola	3	1
2	22,11	44,22	chinelo	4	2

Tabela Normalizada

1. Tabela de 'item_venda' (contendo informações sobre a venda e o produto):

venda_id	Quantidade	produto_id	Subtotal
3	1	1	10,33
4	2	2	44,22

2. coluna inserida na tabela 'produto':

id	valor_unitario	nome	Descricao	Estoque	fabricante	nome_produto

3. Se data_pagamento depende de status (por exemplo, um pagamento só pode ser registrado se o status for "aprovado"), isso também indicaria uma violação da 3FN, pois status não é uma chave primária.

Tabela 'venda':

id	Data	data_envio	Status	valor_total	tipo_pagamento	numero_cartao_pagamento	data_pagamento	cliente_id
1								
2								

Como forma de normalização a estrutura da tabela "venda" poderia ser dividida em:

Tabela venda

id	data	data_envio	valor_total	cliente_id	pagamento_id

Com tipo_pagamento (chave estrangeira para uma nova tabela de pagamento), cliente_id (chave estrangeira para uma tabela de clientes).

Tabela pagamento :

id	tipo_pagamento	status	numero_cartao_credito

4. A partir do código de criação do exercício, depois de feitas as alterações para normalizar temos este código a seguir:

-- Tabela cliente (modificada)

```
CREATE TABLE IF NOT EXISTS cliente (  
  id INT NOT NULL AUTO_INCREMENT,  
  nome VARCHAR(255) NOT NULL,  
  cpf VARCHAR(11) NOT NULL,  
  logradouro VARCHAR(255) NOT NULL,  
  bairro VARCHAR(255) NOT NULL,  
  cidade VARCHAR(255) NOT NULL,  
  estado VARCHAR(2) NOT NULL,  
  PRIMARY KEY (id)  
) ENGINE = InnoDB DEFAULT CHARACTER SET = utf8mb3;
```

-- Tabela telefones_cliente

```
CREATE TABLE IF NOT EXISTS telefones_cliente (  
  idTelefone INT NOT NULL AUTO_INCREMENT,  
  telefone VARCHAR(45) NOT NULL,  
  cliente_id INT NOT NULL,  
  PRIMARY KEY (idTelefone),  
  CONSTRAINT fk_telefones_cliente  
  FOREIGN KEY (cliente_id)  
  REFERENCES cliente (id)  
) ENGINE = InnoDB DEFAULT CHARACTER SET = utf8mb3;
```

-- Tabela produto (nova tabela)

```
CREATE TABLE IF NOT EXISTS produto (  
  id INT NOT NULL AUTO_INCREMENT,  
  valor_unitario DECIMAL(9,2) NOT NULL,  
  nome VARCHAR(255) NOT NULL,  
  descricao TEXT NULL DEFAULT NULL,  
  estoque INT NOT NULL,  
  fabricante VARCHAR(255) NULL DEFAULT NULL,  
  PRIMARY KEY (id)  
) ENGINE = InnoDB DEFAULT CHARACTER SET = utf8mb3;
```

-- Tabela item_venda (modificada)

```
CREATE TABLE IF NOT EXISTS item_venda (  
  venda_id INT NOT NULL,  
  quantidade INT NOT NULL,  
  produto_id INT NOT NULL,  
  subtotal DECIMAL(9,2) NOT NULL,  
  PRIMARY KEY (venda_id, produto_id),  
  CONSTRAINT fk_item_venda_venda  
  FOREIGN KEY (venda_id)
```

```
REFERENCES venda (id),
CONSTRAINT fk_item_venda_produto
FOREIGN KEY (produto_id)
REFERENCES produto (id)
) ENGINE = InnoDB DEFAULT CHARACTER SET = utf8mb3;
```

```
-- Tabela pagamento (nova tabela)
CREATE TABLE IF NOT EXISTS pagamento (
  id INT NOT NULL AUTO_INCREMENT,
  tipo_pagamento VARCHAR(45) NOT NULL,
  status VARCHAR(45) NULL DEFAULT NULL,
  numero_cartao_credito VARCHAR(16) NULL DEFAULT NULL,
  PRIMARY KEY (id)
) ENGINE = InnoDB DEFAULT CHARACTER SET = utf8mb3;
```

```
-- Tabela venda (modificada)
CREATE TABLE IF NOT EXISTS venda (
  id INT NOT NULL AUTO_INCREMENT,
  data DATETIME NULL DEFAULT NULL,
  data_envio DATETIME NULL DEFAULT NULL,
  valor_total DECIMAL(9,2) NULL DEFAULT NULL,
  cliente_id INT NOT NULL,
  pagamento_id INT NOT NULL,
  PRIMARY KEY (id),
  CONSTRAINT fk_venda_cliente
  FOREIGN KEY (cliente_id)
  REFERENCES cliente (id),
  CONSTRAINT fk_venda_pagamento
  FOREIGN KEY (pagamento_id)
  REFERENCES pagamento (id)
) ENGINE = InnoDB DEFAULT CHARACTER SET = utf8mb3;
```