

ATIVIDADE 4

EXERCICIO 1 E 2 - A obediência, ou não, à 1FN, à 2FN e à 3FN. Quais são os problemas observados

1FN

Verificar se os campos necessitam do id

EX. tabela venda

Id venda, cliente_id e funcionario_id da tabela venda necessita da coluna data

Se a resposta for sim, a coluna data permanece na tabela venda

Faça a mesma pergunta para todas as colunas....

A coluna escolhida para a pergunta deve ser dependente de todos os "id".

Exemplo de anomalia de inserção

Pense no cliente no balcão da loja...

Quando ele fechar com o vendedor uma compra, o vendedor irá inserir os dados do produto no sistema.

Posterior o cliente passará no caixa para pagamento, neste momento será inserido os dados de pagamento (n cartão, forma de pagamento....)

Conclusão: se manter dados do produto e pagamento juntos você terá problema de inserção, pois os atos acontecem em momentos diferentes.

Solução: separar as tabelas (tabela venda) (tabela pagamento), por exemplo

Outros exemplos de 1FN é você manter id que faz referência a um produto e na mesma tabela o nome deste produto, isso não é necessário.

2FN não achei no exercício

3FN

Procure por colunas não chave (que não são id) que dependem de outra coluna não chave (o nome disso é dependência transitiva)

EXERCICIO 3 - Qual é a solução proposta

A solução proposta seria que as tabelas fiquem assim, sem repetir dados e criamos uma nova tabela para pagamento_venda onde iremos deixar os dados que são relacionados a cartão, número de parcelas, entre outros, pra melhorar adicionamos "id_item_venda" na tabela item_venda

TABELA < venda >

id: < int not null auto increment PK >

data: < datetime >

status: < varchar(45) >

valor_total: < decimal(9,2) >

cliente_id: < int not null FK >

funcionario_id: < int not null FK >

TABELA < pagamento_venda >

id_pagamento_venda: < int not null auto increment >

tipo_pagamento: < char(1) >

data_pagamento: < datetime >

numero_cartao_pagamento: < varchar(16) >

numero_parcelas_pagamento: < int >

venda_id: < int not null FK >

TABELA < item_venda >

id_item_venda: < int not null >

quantidade: < int not null >

valor_unitario: < decimal(9,2) >

subtotal: < decimal(9,2) >

venda_id: < int not null FK >

produto_id: < int not null FK >

EXERCICIO 4 - O *script* SQL correspondente à alteração proposta

< SCRIPT - REMOVENDO COLUNA NOME_PRODUTO DA TABELA ITEM_VENDA >

Alter table item_venda drop column nome_produto;

< SCRIPT – ADICIONANDO COLUNA ID_ITEM_VENDA >

Alter table item_venda add column id_item_venda int not null auto increment primary key;

< SCRIPT – REMOVENDO COLUNA DATA_ENVIO DA TABELA VENDA >

Alter table venda drop column data_envio;

< SCRIPT – CRIANDO NOVA TABELA PAGAMENTO_VENDA >

Create table pagamento_venda (

 id_pagamento_venda int not null auto increment,

 tipo_pagamento char(1),

 data_pagamento datetime,

 numero_cartao_pagamento varchar(16),

 numero_parcelas_pagamento int,

 venda_id int not null,

 Primary key (id_pagamento_venda),

 Foreign Key (venda_id) references venda(id) on delete cascade);