

### 1. Qual é o objetivo da normalização de um banco de dados?

1) O objetivo de normalização de um banco de dados é evitar falhas de redundância, mistura de assuntos diferentes ou duplicidade em um mesmo projeto.

### 2. Explique os tipos de anomalias de atualização e dê um exemplo de cada uma.

2) Anomalias de alteração, inclusão e exclusão.

Alteração: Se um dado em duplicidade é alterado, o banco de dados apresentará

inconsistências a menos que todas as cópias do banco de dados sejam atualizadas. Se o preço de um produto for alterado, todas as linhas devem ser varridas para alteração.

Inclusão: Não é possível incluir uma informação no banco de dados, a não ser que outra informação não relacionada também seja incluída. Na inclusão de um novo cliente, o mesmo deve estar relacionado a uma venda;

Exclusão: Pode não ser possível apagar uma informação sem perder outra informação não relacionada. Na exclusão de um cliente, os dados dos pedidos relacionados são perdidos;

### 3. Explique o que é a 1FN e dê um exemplo.

3) A 1FN tem como objetivo eliminar colunas multi-valoradas. Um conjunto de colunas relacionados com os produtos são multi-valorados, ou seja, um pedido possui vários produtos;

### 4. Defina e dê exemplos de dependência funcional, dependência funcional parcial, dependência funcional total e dependência funcional transitiva.

4) Dependência funcional: Um conjunto de colunas não vazio A depende funcionalmente de um conjunto de colunas não vazio B (denotado por  $B \rightarrow A$ ) quando o valor de B determina o valor de A. Isto é, para cada valor do determinante B existe um único valor de A. Um exemplo temos  $NP \rightarrow PE$ , ou seja, prazo de entrega depende funcionalmente de número pedido.

Dependência funcional parcial: A dependência funcional parcial acontece quando o valor de uma coluna depende de parte de uma chave primária composta. Um exemplo temos  $CP \rightarrow UP$ , a coluna unidade depende parcialmente da coluna código do produto na tabela item-de-pedido.

Dependência funcional total: A dependência funcional total ocorre quando uma tabela possui uma chave primária composta (formada por várias colunas) e o valor de uma coluna (que não está na chave primária) depende de todas as colunas da chave primária. Na tabela item-de-pedido de nosso exemplo temos  $NP, CP \rightarrow QP$ , isto é, a coluna quantidade depende de forma total das colunas número do pedido e código do produto.

Dependência funcional transitiva: Quando uma coluna A depende de uma coluna B que por sua vez depende da chave primária C, então A é dependente transitivo de C. Voltando ao nosso exemplo, na entidade pedido temos  $NP \rightarrow V$ , pois  $NP \rightarrow CV$  e  $CV \rightarrow V$ . Isto significa que vendedor é dependente transitivamente do número do pedido.

5. Explique o que é a 2FN e dê um exemplo.

5) A 2FN elimina as dependências funcionais parciais, gerando novas tabelas para os conjuntos de colunas com dependência funcional parcial. Cada tabela gerada possui como chave primária as colunas das quais havia a dependência funcional parcial.

Um exemplo pode ser considerando a tabela item-do-pedido, existe uma dependência funcional parcial que pode ser eliminada gerando a tabela produto.

6. Explique o que é a 3FN e dê um exemplo.

6) 3FN elimina as dependências transitivas, quando retirada, cria-se uma nova tabela contendo as colunas com dependência transitiva. A chave primária desta tabela será a coluna que intermediou a dependência transitiva.

Um exemplo é na tabela Pedido temos que vendedor depende transitivamente de número pedido por intermédio de código vendedor. Além da dependência transitiva anterior temos uma dependência transitiva envolvendo as colunas do cliente. Neste caso, podemos tomar o CNPJ como chave primária da tabela a ser criada.

7. Aplique todas as formas normais estudadas nos seguintes esquemas lógicos de banco de dados:

- (a) Empregado (Número Empregado, nome do Empregado, Número Departamento, Nome do Departamento, Número do Gerente, Nome do Gerente, Número do Projeto, Nome do Projeto, Dia de Inicio do Projeto, Número de Horas trabalhadas no Projeto)
- (b) Ordem de Compra (Código ordem compra, data emissão, Código fornecedor, CNPJ fornecedor, endereço fornecedor, Código material (n vezes), descrição material (n vezes), quantidade comprada (n vezes), valor unitário (n vezes), valor total item (n vezes), valor total ordem)
- (c) Tabela de Notas Fiscais (Número NF, Série, Data emissão, Código Cliente, Nome Cliente, Endereço Cliente, CNPJ cliente, Código Mercadoria, Descrição Mercadoria, Quantidade vendida, Preço de venda, Total da venda da Mercadoria, Total geral da Nota)
- (d) Gestão de Projetos (Número Projeto, Número Empregado, Nome Projeto, Nome Empregado, Função, Salário, Horas)
- (e) Vendedor (Número vendedor, nome vendedor, Morada vendedor, telefone, Código postal, Localidade, Número produto, Descrição Produto, estoque, Preço Unitário, Número Factura, Quantidade vendida, Total)
- (f) Trabalhador (Número trabalhador, Nome trabalhador, Número empresa, Nome empresa, Número departamento, Nome departamento)
- (g) Aluno (Código aluno, nome, naturalidade, ano nascimento, Código curso, curso, diretor, ano inscrição, Código disciplina, disciplina, nota)
- (h) Animal (Código animal, raça, nome, data nascimento, código fazenda, nome fazenda, município, código vacina, vacina, data vacinação, lote vacina)

7) tive muita dificuldade de entender essa 7, como ficaria na pratica, ate peguei o exemplo que postou no fórum mas não soube resolver.

a) Para a primeira forma normal (1FN), devemos eliminar os atributos multi-valorados. Nesse caso, podemos entender que um empregado trabalha em vários projetos. Então criamos uma relação com esses dados:

empregado (num\_empregado, nome\_empregado, num\_departamento,  
nome\_departamento, num\_gerente, nome\_gerente)  
empregado\_projeto (\*num\_empregado, num\_projeto, nome\_projeto, data\_inicio,  
horas\_trabalhadas)  
\*empregado\_projeto.num\_empregado: empregado.num\_empregado

Na segunda forma normal (2FN), eliminamos as dependências funcionais parciais, ou seja, dados que dependem de um "pedaço" da chave primária. Isso ocorre na tabela empregado\_projeto. Criamos uma nova tabela e movemos esses dados para ela:

empregado (num\_empregado, nome\_empregado, num\_departamento,  
nome\_departamento, num\_gerente, nome\_gerente)  
projeto (num\_projeto, nome\_projeto, data\_inicio)

empregado\_projeto (\*num\_empregado, \*num\_projeto, horas\_trabalhadas)  
\*empregado\_projeto.num\_empregado: empregado.num\_empregado  
\*empregado\_projeto.num\_projeto: projeto.num\_projeto

Já na terceira forma normal (3FN), eliminamos as dependências funcionais transitivas, isto é, dados que dependem de um atributo que, por sua vez, depende de outro atributo. Podemos observar isso nos dados de departamento presente na tabela empregado.

departamento (num\_departamento, nome\_departamento, num\_gerente,  
nome\_gerente)  
empregado(num\_empregado, nome\_empregado, \*num\_departamento)  
\*empregado.num\_departamento: departamento.num\_departamento

projeto (num\_projeto, nome\_projeto, data\_inicio)  
empregado\_projeto (\*num\_empregado, \*num\_projeto, horas\_trabalhadas)  
\*empregado\_projeto.num\_empregado: empregado.num\_empregado  
\*empregado\_projeto.num\_projeto: projeto.num\_projeto

Observe que o esquema ainda não está na 3FN por que ainda existe uma dependência funcional transitiva nos dados de gerente presentes na tabela departamento. Portanto, temos que aplicar a transformação novamente (isso pode acontecer com qualquer forma normal). Como já existe uma tabela de empregado, simplesmente colocamos uma chave estrangeira apontando para essa tabela para indicar o gerente do departamento:

departamento (num\_departamento, nome\_departamento, \*num\_gerente)  
\*departamento.num\_gerente: empregado.num\_empregado

empregado(num\_empregado, nome\_empregado, \*num\_departamento)  
\*empregado.num\_departamento: departamento.num\_departamento

projeto (num\_projeto, nome\_projeto, data\_inicio)  
empregado\_projeto (\*num\_empregado, \*num\_projeto, horas\_trabalhadas)  
\*empregado\_projeto.num\_empregado: empregado.num\_empregado  
\*empregado\_projeto.num\_empregado: projeto.num\_empregado