

TECNOLOGIA DA INFORMAÇÃO

Banco de Dados Relacionais – Parte IV



Livro Eletrônico



SUMÁRIO

Apresentação.....	3
Banco de Dados Relacionais – Parte IV.....	4
Álgebra Relacional.....	4
Dependência Funcional.....	12
Normalização de Dados.....	14
Resumo.....	22
Questões Comentadas em Aula	24
Questões de Concurso	27
Gabarito.....	46
Referências.....	47

APRESENTAÇÃO

Olá, querido(a) amigo(a) 😊, tudo bem?

Após os estudos sobre os conceitos fundamentais de bancos de dados, vamos ao aprendizado da **álgebra relacional** e da **normalização de dados**.

Estarei adicionando inúmeros exercícios para melhor fixação da matéria. Espero que aproveite!

Finalizando, como qualquer questão pode ser decisiva para a sua aprovação, então, **vamos “arregançar as mangas” e partir para mais esta etapa do curso. Boa sorte nos estudos! Estou torcendo pelo seu sucesso 😊!**

Gostaria de convidá-los para participar do meu canal no Telegram. Lá, irei compartilhar materiais de estudo para TI e Informática, Coaching e dicas para concursos públicos. Acesse <https://t.me/coachpatriciaquintao> para participar.

Em caso de dúvidas, acesse o fórum do curso ou entre em contato.

Um forte abraço!

Profª Patrícia Quintão

Instagram: @coachpatriciaquintao

Telegram: <https://t.me/coachpatriciaquintao>

WhatsApp: (31) 99442.0615

BANCO DE DADOS RELACIONAIS – PARTE IV

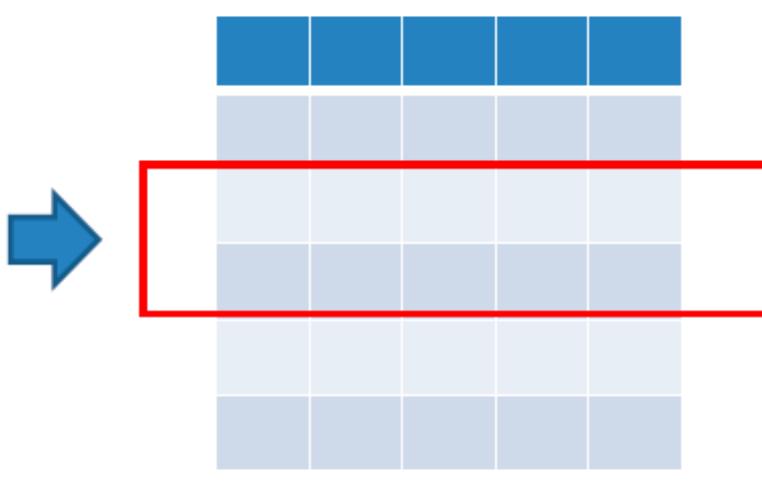
ÁLGEBRA RELACIONAL

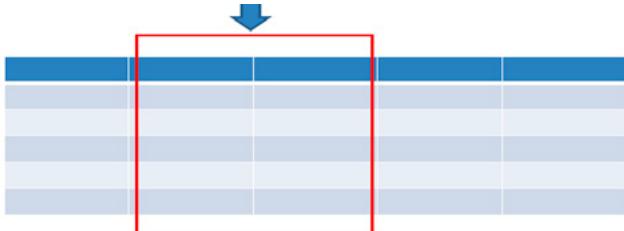
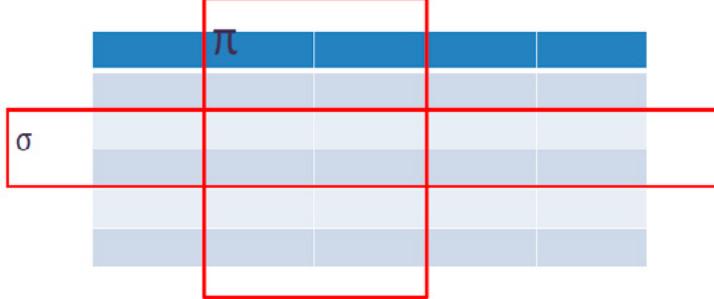
- A **álgebra relacional** é **uma linguagem de consulta formal**, porém **procedimental**, em que o usuário dá as instruções ao sistema para que o mesmo realize uma sequência de operações na base de dados para calcular o resultado desejado.
- A álgebra relacional é **uma forma de cálculo sobre conjuntos ou relações**.
- Existem diversas **operações** em álgebra relacional, como: **seleção, projeção, renomear, produto cartesiano, união e diferença** entre conjuntos, **interseção** de conjuntos, **junção natural**.

Veja que a álgebra relacional **é o conjunto de operações presentes no modelo relacional**. Você verá essas operações na **SQL** (Structured *Query Language* ou **Linguagem de Consulta Estruturada**).

Logo, não se preocupe com detalhes da notação. As principais operações (mais cobradas em prova) são: a **projeção**, a **seleção** e o **produto cartesiano**.

Principais operações da Álgebra Relacional:

Operação	Símbolo	Descrição
Seleção	σ	<p>Seleciona tuplas que satisfaçam uma determinada condição. O resultado é uma relação com a mesma estrutura da tabela original, e que contém as linhas que satisfazem à condição.</p> <p>Por selecionar linhas em uma relação, essa operação é também denominada particionamento horizontal, uma vez que dá um corte horizontal na relação, retornando algumas tuplas e deixando as demais.</p>  <p>Sintaxe para representação da seleção: $\sigma <\text{predicado}> (\text{nome da relação})$, em que: σ (letra grega minúscula sigma) é o símbolo da operação de seleção; $<\text{predicado}>$ é o critério de seleção; (nome da relação) é a relação na qual a operação vai agir.</p> <p><i>Cabe destacar que essa operação NÃO afeta a relação original, que fica intacta. Simplesmente uma nova relação é retornada como resultado da operação.</i> Exemplo:</p> <p>$\sigma DataNascimento > 30/07/1970 (Aluno)$</p> <p>A operação acima listada da Álgebra Relacional retorna as tuplas da relação Aluno nas quais o atributo DataNascimento é maior do que 1970.</p>

Projeção	π	<p>Retorna uma relação apenas com os atributos selecionados. Duplicatas são eliminadas.</p> <p>Por selecionar colunas em uma relação, essa operação é também denominada particionamento vertical, uma vez que a operação dá um corte vertical na relação, retornando algumas colunas e deixando as demais.</p>  <p>Sintaxe para representação da projeção:</p> <p>π<nome_atributos> (nome da relação), em que: π (PI) é o símbolo da operação de projeção; <nome_atributos> é a lista de atributos escolhidos; (nome da relação) é a relação na qual a operação vai agir.</p>
Seleção e projeção juntas		 <p>$\pi_{<\text{nome_atributos}>}(\sigma_{<\text{predicado}>}(\text{nome da relação}))$</p> <p>Exemplo:</p> <p>$\pi_{\text{CPF, Nome, Cidade}}(\sigma_{\text{Nascimento} > 30/07/1970}(\text{Aluno}))$</p> <p>Nesse caso irá primeiramente resolver o que está dentro dos parênteses, sendo retornada uma relação com as tuplas que obedecem à condição estabelecida (que o atributo Nascimento seja maior que 30/07/1970). Nessa relação retornada é feita uma operação de projeção, para retornar apenas os campos listados.</p>

Produto Cartesiano		Combina as informações de duas relações, contendo todos os pares de tuplas possíveis. O número de tuplas resultante será o produto entre o número de tuplas de cada relação.
União		Une dois conjuntos de relações. Ex: $p \text{ nome} (tab_clientes) \cup \text{ nome_cliente} (tab_devedor)$.
Interseção		Encontra as tuplas que estão tanto em uma relação quanto em outra.
Diferença		Encontra as tuplas que estão em uma relação, mas não em outra.
Junção		Esta operação fundamental une duas relações por meio de uma coluna em comum entre elas , efetivando os relacionamentos entre as entidades de um BD.
Divisão		Usada quando a consulta emprega a frase “para todos”, pois responde perguntas do tipo “quais fornecedores fornecem todas as peças?”. Útil em relacionamentos “muitos para muitos”.

Esquematizando

Álgebra Relacional

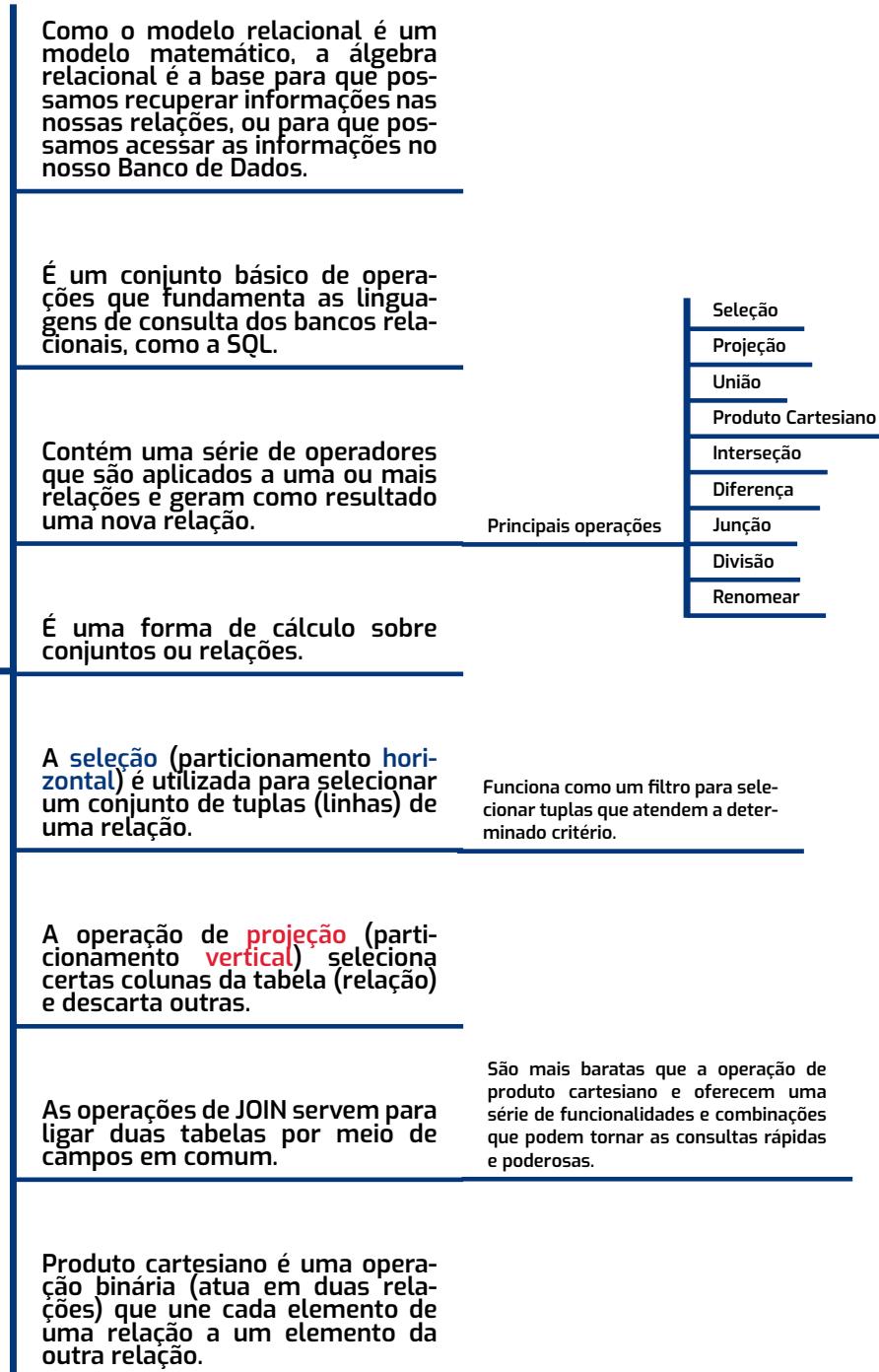


Figura. Álgebra Relacional. Fonte: Quintão (2021)

DIRETO DO CONCURSO

001. (CEBRASPE/CESPE/TCE-ES/ANALISTA ADMINISTRATIVO/INFORMÁTICA/2013) O conjunto de operações cujo resultado seja uma nova relação e que envolve seleção, projeção, união e produto cartesiano é denominado:

- a)** mapeamento de cardinalidades.
- b)** álgebra relacional.
- c)** generalização.
- d)** chave primária
- e)** herança.



Álgebra Relacional pode ser definida como **uma linguagem de consulta formal e procedural**, ou seja, um **conjunto de operações de alto nível sobre relações ou conjuntos** cujo resultado seja uma nova relação ou conjunto.

As principais operações são: seleção, projeção, junção, produto cartesiano, união, intersecção e diferença.

Nota: Qualquer operação sobre uma tabela resultará em uma nova tabela.

Letra b.

002. (CESPE/CEHAP-PB/PROGRAMADOR DE COMPUTADOR/2009) A álgebra relacional é a base matemática de bancos de dados relacionais. A álgebra relacional pode ser definida como linguagem de consulta formal e procedural. Para banco de dados, podem ser utilizadas diversas operações provenientes da teoria de conjuntos. A seleção (select), em banco de dados relacional, é:

- a)** o resultado de todas as tuplas que pertencem as relações presentes em uma operação.
- b)** uma relação que parte de duas outras, levando as tuplas comuns e não comuns a ambas.
- c)** utilizada para escolher subconjunto de tuplas em uma relação que satisfaça condição de seleção predefinida.
- d)** executada em apenas uma relação, e o resultado é uma nova relação.



A **seleção (particionamento horizontal)** é utilizada para **selecionar um subconjunto de tuplas (linhas) de uma relação**. Assim, ela funciona como um filtro para selecionar tuplas que atendem a determinado critério.

Letra c.

003. (VUNESP/MPE-ES/AGENTE ESPECIALIZADO/ANALISTA DE BANCO DE DADOS/2013)

Dentre os diversos tipos de operações disponibilizadas em um banco de dados relacional está, por exemplo, a realização de consultas sobre valores armazenados em tabelas. A operação que consiste em definir quais devem ser as colunas a serem exibidas em uma consulta é a

- a)** divisão.
- b)** multiplexação.
- c)** projeção.
- d)** seleção.
- e)** união.



A operação de **projeção (particionamento vertical)** seleciona certas colunas da tabela (relação) e descarta outras.

Letra c.

004. (CCV-UFC/UFC/ANALISTA DE TECNOLOGIA DA INFORMAÇÃO/2018) De acordo com a álgebra relacional, marque a opção que contenha apenas operações fundamentais.

- a)** Divisão, seleção, diferença.
- b)** Agregação, projeção, união.
- c)** Junção natural, seleção, projeção.
- d)** Seleção, projeção, produto cartesiano.
- e)** Interseção, produto cartesiano, junção natural.



Existem diversas operações em álgebra relacional, entre as quais: seleção, projeção, renomear, produto cartesiano, união e diferença entre conjuntos, interseção de conjuntos, junção natural.

Seleção, projeção, produto cartesiano são **operações fundamentais**:

- A **seleção** é utilizada para selecionar um conjunto de tuplas (linhas) de uma relação;
- A operação de **projeção (particionamento vertical)** seleciona certas colunas da tabela (relação) e descarta outras;
- **Produto cartesiano** é uma operação binária (atua em duas relações) que une cada elemento de uma relação a um elemento da outra relação.

Letra d.

005. (FIP/CÂMARA MUNICIPAL DE SÃO JOSÉ DOS CAMPOS-SP/PROGRAMADOR/2009)

Em um banco de dados relacional, duas tabelas foram concatenadas de forma a atender uma determinada condição. O resultado dessa operação representa a operação relacional de:

- a)** união.
- b)** projeção.

- c) seleção.
- d) intersecção.
- e) junção.



Trata-se da operação de **junção** (\bowtie), quando dois conjuntos de dados são concatenados de acordo com uma determinada condição.

Letra e.

006. (FCC/MPE-RS/TÉCNICO EM INFORMÁTICA/ÁREA DE SISTEMAS/2008) Quando dois conjuntos de dados são concatenados de acordo com uma determinada condição, representa o resultado da operação relacional:

- a) junção.
- b) união.
- c) restrição.
- d) projeção.
- e) intersecção.



Conforme vimos, em um banco de dados relacional, utilizamos a operação de **junção** (\bowtie) quando duas tabelas foram concatenadas de forma a atender uma determinada condição.

Letra a.

007. (NCE/UFRJ/ANALISTA DE TECNOLOGIA DA INFORMAÇÃO/BANCO DE DADOS/2010) Dadas duas relações A e B do mesmo tipo, uma relação de tipo igual, cujo conteúdo contém todas as tuplas que aparecem tanto em A quanto em B, será obtida por meio da operação relacional:

- a) união.
- b) intersecção.
- c) projeção.
- d) junção.
- e) produto cartesiano.



Interseção (\bowtie) é uma operação que produz como resultado uma tabela que contém, sem repetições, **todos os elementos que são comuns às duas tabelas fornecidas como operandos**.

Letra b.

008. (AOCP/TCE-PA/ASSESSOR TÉCNICO DE INFORMÁTICA) Em banco de dados, a diferença entre duas instâncias é:

- a)** o conjunto de todas as instâncias i pertencentes a A ou B ou ambos, quando A e B possuem uma união compatível.
- b)** o conjunto de todas as instâncias i pertencentes a A e B, quando A e B possuem uma união compatível.
- c)** o conjunto de todas as instâncias i pertencentes a A mas não pertencentes a B, quando A e B possuem uma união compatível.
- d)** o conjunto de todas as instâncias i , quando i é a concatenação de uma instância x de A com uma instância y de B.
- e)** o conjunto de valores n , tais que os pares (n, m) aparecem em A para todos os valores m que aparecem em B.



A **diferença** (-) é uma operação que produz como resultado uma tabela que contém todas as linhas que existem na primeira tabela e não existem na segunda.

Letra c.

DEPENDÊNCIA FUNCIONAL

Antes de explicitar as **formas normais** e destacar a linguagem **SQL (Structured Query Language ou Linguagem de Consulta Estruturada)**, vamos ao conceito de **dependência funcional**.

Dependência funcional nada mais é do que **um relacionamento que existe entre atributos de uma relação**.

Utilizemos um exemplo bem simples para entender esse conceito.

Antes disso, vale a observação seguinte: no **modelo relacional** podemos descrever as **tabelas, seus atributos e restrições com os atributos entre parênteses e as chaves sublinhadas**. Alguns autores sublinham “cheio” as chaves primárias e “pontilhado” as chaves estrangeiras.

Exemplo de Dependência Funcional:

ATOR

<u>IdAtor</u>	<u>nomeAtor</u>
1	José Carlos
2	Antônio da Silva
3	José Carlos

Considerando a Relação ATOR acima, temos que nomeAtor depende funcionalmente de idAtor.

O que significa isso? Ora, significa que pelo idAtor eu descubro o nomeAtor. Ou seja, idAtor determina nomeAtor, e nomeAtor depende funcionalmente de idAtor.

Isso é representado da seguinte forma:

idAtor -> nomeAtor

Mas será que o contrário é verdadeiro? Não, não é. Eu posso ter dois atores com o mesmo nome, mas eles teriam idAtor diferentes. Então, pelo nomeAtor eu não determino idAtor.

Uma chave primária SEMPRE determina os demais atributos de uma relação.

Mas aí vem a pergunta, somente as chaves primárias determinam outro atributo? Não, posso construir a seguinte relação:

ItemNotaFiscal

<u>IdNota</u>	<u>IdItem</u>	<u>CodProduto</u>	<u>Quantidade</u>	<u>ValorUnitario</u>	<u>Total</u>
1	1	100	2	10,00	20,00
1	2	200	5	5,00	25,00
2	1	205	3	12,00	48,00

Bem, já sabemos que a chave primária determina os outros atributos. Mas olhem o atributo Total. O que é total? Simplesmente é Quantidade x ValorUnitario. Então, esses dois campos determinam Total, existindo assim uma dependência funcional (DF):

Quantidade,Valor -> Total

Dessa forma, temos que: **um atributo Y é funcionalmente dependente de um atributo X se cada valor de X tenha associado a ele precisamente um valor de Y.**

Quando o atributo X é uma **chave primária**, então todos os atributos são, por definição, dependentes de X, pois **não podem existir dois registros com o mesmo valor para X**.

Notação: $R.x \rightarrow R.y$ (lê-se **a coluna x da tabela relacional R funcionalmente determina (identifica) a coluna y**).

A **dependência funcional** pode ser **classificada** em:

- **Total:** um atributo é **totalmente dependente** de outro se ele for funcionalmente dependente do outro e não dependente de um subconjunto de outro;
- **Parcial:** um atributo é **parcialmente dependente** de outro se ele for funcionalmente dependente de um subconjunto de outro.

Considere a tabela seguinte, em que a chave primária é formada pelos atributos código-func + código-curso.

- O atributo *avaliação* é dependente total da chave composta.
- Já o atributo *descrição-curso* tem dependência parcial com relação a esta chave, pois depende somente de parte dela, ou seja, de *código-curso*.

<i>código-func</i>	<i>código-curso</i>	<i>descrição-curso</i>	<i>avaliação</i>	<i>data-conclusão</i>
00001	ENG01	ENG. CIVIL	A	01/01/2005

NORMALIZAÇÃO DE DADOS

A **normalização** pode ser vista como o **processo no qual são eliminados esquemas de relações (tabelas) não satisfatórios**, decompondo-os, por meio da separação de seus atributos em esquemas de relações menos complexas, mas que satisfaçam às propriedades desejadas.

Em suma, a **normalização de dados** é uma série de passos que se segue no projeto de um banco de dados que **permite um armazenamento consistente e um eficiente acesso aos dados** em um banco de dados relacional. Esses passos **reduzem a redundância de dados e as chances dos dados se tornarem inconsistentes**.

DIRETO DO CONCURSO

- 009.** (FCC/MPE-MA/ANALISTA MINISTERIAL/BANCO DE DADOS/2013) Em um banco de dados relacional, a atividade de normalizar suas tabelas tem como objetivo
- gerar os triggers especificados para o banco de dados.
 - eliminar, ou pelo menos diminuir redundâncias de dados desnecessárias nessas tabelas.
 - impedir o acesso ao banco de dados, por parte de usuários desconhecidos.

- d) compilar todas as funções inseridas no banco de dados.
 e) preparar todas as tabelas do banco de dados para um backup completo.



A **normalização** de dados é uma série de passos que se segue no projeto de um banco de dados que permite um armazenamento consistente e um eficiente acesso aos dados em um banco de dados relacional. Esses passos **reduzem a redundância de dados** e as chances dos dados se tornarem inconsistentes.

Letra b.

As primeiras formas normais foram criadas na década de 1970 por Codd, e persistem até hoje. Elas são importantes para verificar se seu Banco de Dados está bem projetado.

Vamos ver aqui as principais formas normais.

- **Primeira Forma Normal (1FN)**

Uma relação estará na 1^a FN se não houver atributo representando agrupamento (não atômico) e nem atributo repetitivo (multivalorado).

A **1^a Forma Normal** prega que todos os atributos de uma tabela devem ser **atômicos (indivisíveis)**, ou seja, **NÃO** são permitidos **atributos multivalorados, atributos compostos ou atributos multivalorados compostos**.

Leve em consideração o esquema a seguir:

CLIENTE

1. Código
2. { Telefone }
3. Endereço: (Rua, Número, Cidade)

Gerando a tabela resultante:

Cliente	<u>Código</u>	Telefone 1	Endereço		
		Telefone <i>n</i>	Rua	Número	Cidade

Sendo que a mesma não está na 1^a Forma Normal pois seus atributos não são atômicos.

Para que a tabela acima fique na 1 Forma Normal temos que eliminar os atributos não atômicos, gerando as seguintes tabelas como resultado:

Cliente	<u>Código</u>	Rua	Número	Cidade
Cliente_Telefone		<u>Código_Cliente</u>	<u>Telefone_Cliente</u>	

- **Segunda Forma Normal (2FN)**

Uma relação estará na 2^a FN, se e somente se, estiver na 1a FN e os seus atributos não chaves forem dependentes funcionais completos da chave primária.

Em outras palavras, **se algum atributo da tabela depender funcionalmente apenas de parte da chave primária, então este atributo deverá migrar para outra tabela**. Dessa forma, eu não posso ter um atributo na chave primária que, sozinho, define um outro atributo não chave.

A **2^a Forma Normal** prega o conceito da **dependência funcional total**. Uma dependência funcional $X \rightarrow Y$ é **total** se removemos um atributo **A** qualquer do componente **X** e desta forma, a dependência funcional deixa de existir.

A dependência funcional $X \rightarrow Y$ é uma **dependência funcional parcial** se existir um atributo **A** qualquer do componente **X** que pode ser removido e a dependência funcional $X \rightarrow Y$ não deixa de existir.

{ RG_Empregado, Número_Projeto } → Horas

é uma dependência funcional total, pois se removermos o atributo **RG_Empregado** ou o atributo **Número_Projeto**, a dependência funcional deixa de existir.

Uma tabela T está na 2 Forma Normal se estiver na 1 Forma Normal e todos os seus atributos não chaves forem totalmente funcionalmente dependentes da chave primária C.

Se uma tabela não está na 2^a Forma Normal, a mesma pode ser normalizada gerando outras tabelas cujos atributos que não façam parte da chave primária sejam totalmente funcionalmente dependentes da mesma, ficando a tabela na 2^a Forma Normal.

Exemplo: seja a relação **AtorParticipa (IdDVD, IdAtor, NomeAtor)** aqui destacada. Essa relação **não** está na 2FN, porque um dos atributos que fazem parte da chave primária, de forma isolada, determina um atributo não chave. Ou seja, idAtor determina NomeAtor.

E como fazer para deixar essa relação na 2FN? Simples, basta tirar o atributo nomeAtor dessa relação e deixá-lo apenas na relação Ator.

Autor (IdAutor, nomeAutor)

- **Terceira Forma Normal (3FN)**

Uma relação estará na 3^a FN, se e somente se, estiver na 2^a FN e todos os seus atributos não chaves forem dependentes não transitivos da chave primária.

Assim, a 3 Forma Normal prega o conceito de dependência transitiva. Uma tabela está na 3^a Forma Normal se estiver na 2^a Forma Normal e não houver dependência transitiva entre atributos não chave.

Como exemplo, veja a relação seguinte:

ItemNotaFiscal (IdNota, IdItem, CodProduto, Quantidade, ValorUnitario, Total)

Bem, a chave idNota e IdItem determinam CodProduto, Quantidade e ValorUnitario. Mas Total não depende funcionalmente diretamente da chave. Esse atributo depende funcionalmente de Quantidade e ValorUnitario. Assim temos:

IdNota,IdItem -> Quantidade,ValorUnitario -> Total

Dessa forma, temos uma dependência transitiva, que deve ser evitada pela 3FN.

Mas trocando em miúdos, para uma relação estar em 3FN um campo não chave não pode ser determinado por outro(s) campo(s) não chave. No caso da 3FN, fazemos essa checagem para cada chave candidata.

O que seria essa Dependência Transitiva?

Quando você terminar de normalizar a **2FN**, TODOS os seus atributos serão plenamente dependentes da chave primária. Mas, isso não impede a dependência transitiva, pois se $A \rightarrow B$ e $B \rightarrow C$, por transitividade $A \rightarrow C$, logo C é plenamente dependente da chave primária. Em outras palavras se CPF \rightarrow CARGO e o CARGO \rightarrow SALARIO, CPF também determina o salário. Esse tipo de dependência funcional será resolvido pela 3FN.

A **normalização** feita a partir da regra definida pela terceira forma normal leva a relação para um estado específico. Neste, a relação tem que estar na segunda forma normal e ainda todo atributo não primário da relação não é transitivamente dependente de uma chave da relação.

Uma relação está na **Terceira Forma Normal (3NF) se ela está na 2FN e nenhum atributo não chave (não primário) é transitivamente dependente de uma chave candidata. Enfim, na 3FN não se aceita dependência transitiva.**

Se você achou essa definição de dependência transitiva complexa, deixa eu tentar explicar de outra forma. Primeiro você precisa ter em mente que, para existir a transitividade, temos que ter algumas premissas. Um atributo chave (primário), por exemplo, CPF, determina um outro atributo (não primário), por exemplo, telefoneResidencial; que, por sua vez, determina outro atributo (não primário), por exemplo, Endereco.

CPF Nome telResidencial Endereco
001 JUuca 31 525-1285 Alameda 3, vila
002 Luiz 31 525-1285 Alameda 3, vila
003 Carlos 31 555-1533 Alameda 3, vila
004 Mateus 31 555-1533 Alameda 3, vila
005 Ladjane 81 555-9299 Av. Portugal

Veja que, se você me passar um número de CPF, eu devolvo um telefone residencial. Da mesma forma, se você me der um número de telefone, eu devolvo um endereço único. Observe que alguns telefones aparecem mais de uma vez na coluna, contudo, eles determinam o mesmo endereço, ou, em outras palavras, eles têm o mesmo endereço associado. É justamente essa replicação que desejamos evitar na terceira forma normal.

- **Forma Normal de Boyce-Codd (BCNF)**

Além da 3FN, todo atributo não chave deve depender funcionalmente diretamente da chave primária, ou seja, **não pode haver dependências entre atributos não chave**. É uma afirmação um pouco mais forte do que a 3FN.

É difícil encontrar uma relação que esteja em 3FN e que não esteja em BCNF. Um conselho: não se preocupem muito com a BCNF, eu a citei apenas imaginando que a banca possa cobrar o conceito, e ver se vocês pelo menos ouviram falar dela. Entendendo os conceitos de 1FN a 3FN está ótimo!

Também ilustrei aqui a 4FN e a 5FN para uma visão geral, mas raramente são cobradas!

- **Quarta Forma Normal (4FN)**

Uma relação estará na 4^a FN, se e somente se, estiver na 3a FN e não possuir dependências multivaloradas.

Conforme destaca Oliveira (2002), “pode ocorrer de estarmos com um modelo na 3FN e mesmo assim haver alguma redundância. Isso ocorrerá quando um **atributo não chave contiver valores múltiplos para uma mesma chave**, o que indica uma **dependência multivalorada** (a repetição de dois ou mais atributos não chave, gerando uma redundância desnecessária no modelo)”.

Veja o exemplo com dependência multivalorada, citado por Oliveira (2002), a seguir:

Música	Intérprete	Gravadora
Será	Renato Russo	EMI
Será	Simone	PolyGram
Será	Renato Russo	PolyGram
Imagine	John Lennon	EMI
Imagine	Simone	EMI
Imagine	John Lennon	PolyGram

Observe que não podemos criar uma chave com música (ela se repete), nem com música + intérprete (também há repetição), e caso coloquemos os três campos como chave, teremos uma alternativa válida, mas redundante, pois haverá repetição de música e autor ou música e gravadora.

Solução (com aplicação da 4FN): é necessário dividir essa entidade em duas. Cada uma das entidades herdará a Música como chave e conterá o outro atributo. A primeira entidade ficaria assim:

<u>Música</u>	<u>Intérprete</u>
Será	Renato Russo
Será	Simone
Será	Renato Russo
Imagine	John Lennon
Imagine	Simone
Imagine	John Lennon

e a outra assim:

<u>Música</u>	<u>Gravadora</u>
Será	EMI
Será	PolyGram
Será	PolyGram
Imagine	EMI
Imagine	MI
Imagine	PolyGram

Outro exemplo em: http://www.inf.unioeste.br/~clodis/BDI/BDI_2007_Modulo2_6.pdf.

- **Quinta Forma Normal (5FN)**

Caso raro de ocorrer, é o que destaca Oliveira (2002). Segundo esse autor, tecnicamente, **utiliza-se a 5FN quando uma tabela na 4FN pode ser subdividida em duas ou mais tabelas, para evitar eventuais redundâncias ainda existentes.**

É como se tivéssemos, na aplicação da 4FN, deixado três campos em uma das tabelas criadas e esses campos tivessem novamente valores multivvalorados.

Solução: Para solucionar essa redundância, basta dividir novamente a tabela em duas outras, carregando o campo-chave e herdando o outro atributo.

A verificação da 5 FN somente precisa ser empreendida em relações que tenham 3 ou mais atributos como parte da chave.

A 5 a FN serve para eliminar dependência de junção.

Veja um exemplo em: http://www.inf.unioeste.br/~clodis/BDI/BDI_2007_Modulo2_6.pdf.

Obs.: Para um banco de dados se encontrar em cada um desses **estágios** ou **formas** (denominadas **formas normais**), cada uma de suas tabelas deve atender a alguns pré-requisitos.

Os **pré-requisitos são cumulativos**, isto é, para alcançar a 3^a forma normal (3NF) por exemplo, um banco de dados precisa atender aos pré-requisitos das 1^a e 2^a formas normais, acrescidos dos requisitos exclusivos da 3NF.

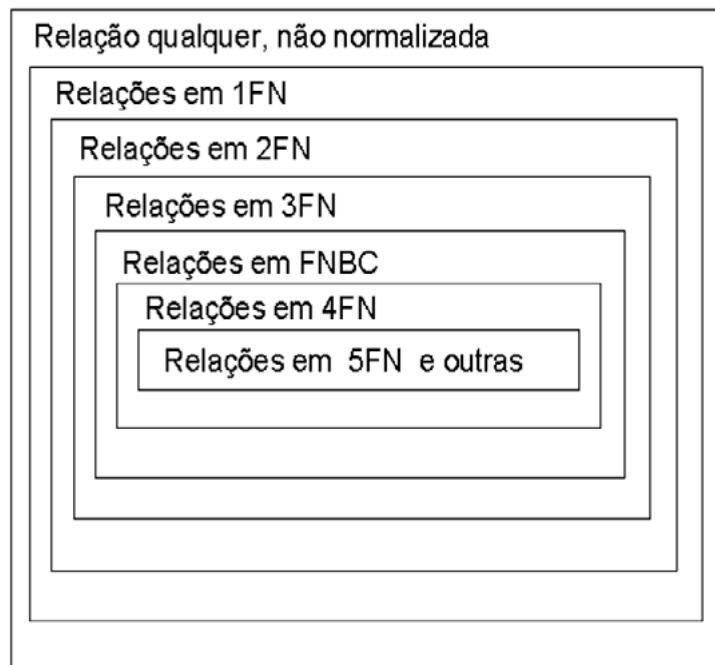


Figura. Normalização

RESUMO

Álgebra Relacional

Como o modelo relacional é um modelo matemático, a álgebra relacional é a base para que possamos recuperar informações nas nossas relações, ou para que possamos acessar as informações no nosso Banco de Dados.

É um conjunto básico de operações que fundamenta as linguagens de consulta dos bancos relacionais, como a SQL.

Contém uma série de operadores que são aplicados a uma ou mais relações e geram como resultado uma nova relação.

É uma forma de cálculo sobre conjuntos ou relações.

A **seleção** (particionamento horizontal) é utilizada para selecionar um conjunto de tuplas (linhas) de uma relação.

A operação de **projeção** (particionamento vertical) seleciona certas colunas da tabela (relação) e descarta outras.

As operações de **JOIN** servem para ligar duas tabelas por meio de campos em comum.

Produto cartesiano é uma operação binária (atua em duas relações) que une cada elemento de uma relação a um elemento da outra relação.

Principais operações

- Seleção
- Projeção
- União
- Produto Cartesiano
- Interseção
- Diferença
- Junção
- Divisão
- Renomear

Figura. Álgebra Relacional (QUINTÃO, 2021)



Figura. Formas Normais (QUINTÃO, 2021)

QUESTÕES COMENTADAS EM AULA

001. (CEBRASPE/CESPE/TCE-ES/ANALISTA ADMINISTRATIVO/INFORMÁTICA/2013) O conjunto de operações cujo resultado seja uma nova relação e que envolve seleção, projeção, união e produto cartesiano é denominado:

- a)** mapeamento de cardinalidades.
- b)** álgebra relacional.
- c)** generalização.
- d)** chave primária
- e)** herança.

002. (CESPE/CEHAP-PB/PROGRAMADOR DE COMPUTADOR/2009) A álgebra relacional é a base matemática de bancos de dados relacionais. A álgebra relacional pode ser definida como linguagem de consulta formal e procedural. Para banco de dados, podem ser utilizadas diversas operações provenientes da teoria de conjuntos. A seleção (select), em banco de dados relacional, é:

- a)** o resultado de todas as tuplas que pertencem as relações presentes em uma operação.
- b)** uma relação que parte de duas outras, levando as tuplas comuns e não-comuns a ambas.
- c)** utilizada para escolher subconjunto de tuplas em uma relação que satisfaça condição de seleção predefinida.
- d)** executada em apenas uma relação, e o resultado é uma nova relação.

003. (VUNESP/MPE-ES/AGENTE ESPECIALIZADO/ANALISTA DE BANCO DE DADOS/2013) Dentre os diversos tipos de operações disponibilizadas em um banco de dados relacional está, por exemplo, a realização de consultas sobre valores armazenados em tabelas. A operação que consiste em definir quais devem ser as colunas a serem exibidas em uma consulta é a

- a)** divisão.
- b)** multiplexação.
- c)** projeção.
- d)** seleção.
- e)** união.

004. (CCV-UFC/UFC/ANALISTA DE TECNOLOGIA DA INFORMAÇÃO/2018) De acordo com a álgebra relacional, marque a opção que contenha apenas operações fundamentais.

- a)** Divisão, seleção, diferença.
- b)** Agregação, projeção, união.
- c)** Junção natural, seleção, projeção.
- d)** Seleção, projeção, produto cartesiano.
- e)** Interseção, produto cartesiano, junção natural.

005. (FIP/CÂMARA MUNICIPAL DE SÃO JOSÉ DOS CAMPOS-SP/PROGRAMADOR/2009)

Em um banco de dados relacional, duas tabelas foram concatenadas de forma a atender uma determinada condição. O resultado dessa operação representa a operação relacional de:

- a)** união.
- b)** projeção.
- c)** seleção.
- d)** intersecção.
- e)** junção.

006. (FCC/MPE-RS/TÉCNICO EM INFORMÁTICA/ÁREA DE SISTEMAS/2008) Quando dois conjuntos de dados são concatenados de acordo com uma determinada condição, representa o resultado da operação relacional:

- a)** junção.
- b)** união.
- c)** restrição.
- d)** projeção.
- e)** intersecção.

007. (NCE/UFRJ/ANALISTA DE TECNOLOGIA DA INFORMAÇÃO/BANCO DE DADOS/2010)

Dadas duas relações A e B do mesmo tipo, uma relação de tipo igual, cujo conteúdo contém todas as tuplas que aparecem tanto em A quanto em B, será obtida por meio da operação relacional:

- a)** união.
- b)** intersecção.
- c)** projeção.
- d)** junção.
- e)** produto cartesiano.

008. (AOCP/TCE-PA/ASSESSOR TÉCNICO DE INFORMÁTICA) Em banco de dados, a diferença entre duas instâncias é:

- a)** o conjunto de todas as instâncias i pertencentes a A ou B ou ambos, quando A e B possuem uma união compatível.
- b)** o conjunto de todas as instâncias i pertencentes a A e B, quando A e B possuem uma união compatível.
- c)** o conjunto de todas as instâncias i pertencentes a A mas não pertencentes a B, quando A e B possuem uma união compatível.
- d)** o conjunto de todas as instâncias i, quando i é a concatenação de uma instância x de A com uma instância y de B.
- e)** o conjunto de valores n, tais que os pares (n, m) aparecem em A para todos os valores m que aparecem em B.

009. (FCC/MPE-MA/ANALISTA MINISTERIAL/BANCO DE DADOS/2013) Em um banco de dados relacional, a atividade de normalizar suas tabelas tem como objetivo

- a)** gerar os triggers especificados para o banco de dados.
- b)** eliminar, ou pelo menos diminuir redundâncias de dados desnecessárias nessas tabelas.
- c)** impedir o acesso ao banco de dados, por parte de usuários desconhecidos.
- d)** compilar todas as funções inseridas no banco de dados.
- e)** preparar todas as tabelas do banco de dados para um backup completo.

QUESTÕES DE CONCURSO

010. (CESPE/EBSERH/ANALISTA DE TECNOLOGIA DA INFORMAÇÃO/2018) Com relação a banco de dados, julgue o item seguinte. Em normalização, a primeira forma normal é caracterizada por uma tabela com a existência obrigatória de uma chave primária e uma chave estrangeira.



Em um projeto de banco de dados relacional o conceito de chave primária e chave estrangeira é um dos principais pilares para manter a integridade dos dados e não está diretamente relacionado à **primeira forma normal (1FN)**. Isso quer dizer que mesmo tendo uma tabela desnormalizada ela ainda precisa das chaves **primária** (identificar de maneira única o registro) e **estrangeira** (realizar a relacionamento entre os registros).

Uma tabela está na **1FN** se, e somente se, **todos os valores das colunas (todos os atributos) forem atômicos**. Assim, a **1FN** diz que um atributo não pode conter múltiplos valores (**não pode possuir atributos multivalorados nem compostos**).

Imagine o atributo ENDERECO que é composto pelo nome da rua, bairro e cidade. Várias informações em um único campo não podem existir (viola a 1FN). Nesse contexto, o valor do atributo deve ser indivisível ou único.

Conforme visto, o item da questão está errado!

Errado.

011. (ESAF/SUSEP-TI/2006) Em um Banco de Dados Relacional

- a)** uma relação está na 1FN (primeira forma normal) se nenhum domínio contiver valores atômicos.
- b)** uma Chave Primária corresponde ao identificador único de uma determinada relação. Em uma relação pode haver mais que uma coluna candidata a chave primária.
- c)** as colunas que irão compor as Chaves Primárias devem ser inicializadas com valores nulos.
- d)** em uma tabela existirão tantas Chaves Primárias quantas forem as colunas nela existentes.
- e)** uma Chave Externa é formada por uma coluna de uma tabela que se referencia a uma Coluna qualquer de outra tabela. Essas colunas, na tabela destino, não aceitam valores nulos. Uma tabela destino pode ter apenas uma Chave Externa.

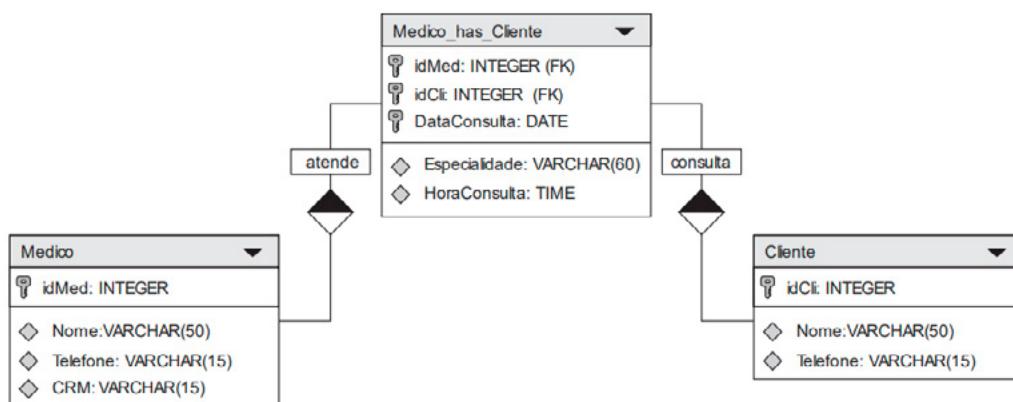


- a)** Errada. Na 1FN, todos devem conter valores atômicos.
- b)** Certa. Mais de uma chave pode ser candidata a chave primária.
- c)** Errada. A chave primária deve conter valores que identifiquem unicamente cada registro da tabela, não podendo ser nulos.
- d)** Errada. A chave primária deve ser única para a tabela.

e) Errada. Uma tabela pode conter mais de uma chave externa.

Letra b.

012. (FCC/TRT-16^a REGIÃO/MA/ANALISTA JUDICIÁRIO/TECNOLOGIA DA INFORMAÇÃO/2014) Em um consultório médico as consultas são agendadas de forma que um cliente não possa realizar mais do que uma consulta com o mesmo médico na mesma data, porém, pode passar em consulta com outros médicos. O modelo Entidade-Relacionamento do banco de dados do consultório é apresentado a seguir:



É correto afirmar que

- a) a tabela *Cliente* não está na Primeira Forma Normal.
- b) o campo *idMed* é chave estrangeira na tabela *Medico* e chave primária na tabela *Medico_has_Cliente*.
- c) a tabela *Medico_has_Cliente* não está na Segunda Forma Normal.
- d) a chave primária da tabela *Medico_has_Cliente* deve ser apenas *DataConsulta*.
- e) a tabela *Medico_has_Cliente* não é necessária. Deve-se ligar as tabelas *Medico* e *Cliente* em um relacionamento de grau 1:n.



- a) Errada. Uma tabela está na primeira formal normal quando todos os campos são atômicos. Como nenhum campo dessa tabela é multivlorado, a tabela está na primeira forma normal.
- b) Errada. O campo *idMed* é chave primária na tabela *Medico*.
- c) Certa. A letra C está correta pois o atributo *Especialidade* depende somente de *idMed* que é parte da chave. Logo a tabela não está na segunda forma normal.
- d) Errada. A consulta deve depender dos identificadores do cliente e do médico.
- e) Errada. A tabela *Medico_has_cliente* é necessária para registrar a consulta.

Formas Normais Básicas Principais

1^a Forma Normal. Uma relação estará na 1^a FN se não houver atributo representando agrupamento (não atômico) e nem atributo repetitivo (multivlorado).

2^a Forma Normal. Uma relação estará na 2^a FN, se e somente se, estiver na 1a FN e os seus atributos não chaves forem dependentes funcionais completos da chave primária.

3^a Forma Normal. Uma relação estará na 3^a FN, se e somente se, estiver na 2 a FN e todos os seus atributos não chaves forem dependentes não transitivos da chave primária.

Letra c.

013. (FCC/ANALISTA JUDICIÁRIO-TI/TRT-19^a/2011) Para uma tabela estar na FNBC (Forma Normal Boyce- Codd), ela

- a) não precisa da normalização 1FN.
- b) precisa estar somente na 2FN.
- c) também está normalizada na 3FN.
- d) tem de estar normalizada até a 4FN.
- e) tem de estar normalizada até a 5FN.



Para estar na FNBC (Forma Normal Boyce-Codd), além da 3FN, todo atributo não chave deve depender funcionalmente diretamente da chave primária, ou seja, **não pode haver dependências entre atributos não chave**.

Assim, ela também está normalizada na 3FN.

Letra c.

014. (FCC/TRT-23^a/ANALISTA JUDICIÁRIO-TI/2011) No contexto de normalização, quando a tabela não contém tabelas aninhadas e não possui colunas multivaloradas; não contém dependências parciais, embora contenha dependências transitivas, diz-se que ela está na

- a) primeira forma normal (1FN).
- b) segunda forma normal (2FN).
- c) terceira forma normal (3FN).
- d) quarta forma normal (4FN).
- e) quinta forma normal (5FN).



A banca cita que não possui tabelas aninhadas, ou seja, **não tem atributos compostos, e não possui campos multivalorados**. Então, até esse ponto já está pelo menos na **1FN**.

Em seguida, destaca que **a tabela não contém dependências parciais**, ou seja, nenhum atributo não chave depende funcionalmente de apenas uma parte da chave primária. Então, está já na **2FN**.

Em seguida, cita que a tabela **possui dependências transitivas**, o que **contraria a 3FN**. Ficamos então na 2FN.

Letra b.

015. (ESAF/PREFEITURA DE NATAL/RN/AUDITOR DO TESOURO NACIONAL/2008) A “Normalização” é um método utilizado para analisar um Banco de Dados e obter o mínimo de redundância e o máximo de integridade de dados.



A **normalização** consiste em um **processo formal de exame e agrupamento de dados para: suportar melhor as mudanças futuras; minimizar o impacto destas mudanças sobre a base de dados**. Trata-se de um conjunto de regras que leva à construção de modelos mais robustos, com menos dependências entre seus elementos e menos redundância de informações. **Normalização é, portanto, uma atividade de verificação do modelo lógico**. Principais objetivos: reduzir as redundâncias; reduzir a necessidade de reestruturar as tabelas do banco de dados quando novos tipos de dados são introduzidos.

Com relação às suas Formas Normais mais comuns, apesar de existirem outras, temos:

- **1^a Forma Normal (1FN): toda relação deve ter uma chave primária e deve-se garantir que todo atributo seja atômico. Atributos compostos devem ser separados.**

Por exemplo, um atributo Endereço deve ser subdividido em seus componentes: Logradouro, Número, Complemento, Bairro, Cidade, Estado e CEP.

Além disso, atributos multivvalorados devem ser discriminados separadamente ou separados em uma outra relação. Por exemplo, um atributo multivvalorado Telefones poderia ser separado em Telefone Residencial, Telefone Comercial e Telefone Celular ou, ainda, ser convertido em outra relação que pudesse representar um número indeterminado de telefones.

- **2^a Forma Normal (2FN): toda relação deve estar na 1FN e devem-se eliminar dependências funcionais parciais, ou seja, todo atributo não chave deve ser totalmente dependente da chave primária.**

Como exemplo, uma relação que contenha os atributos Código da Obra, Código do Fornecedor, Nome do Fornecedor e Preço de Venda, considerando que a chave primária é composta pelos atributos Código da Obra e Código do Fornecedor, não está na Segunda Forma Normal, uma vez que o Nome do Fornecedor depende apenas do Código do Fornecedor, e não do Código da Obra. Uma nova relação (Fornecedor) deve ser criada contendo os campos Código do Fornecedor (como chave) e Nome do Fornecedor. Na relação original, ficariam os atributos Código da Obra e o Código do Fornecedor, ambos formando a chave primária composta, e o atributo Preço de Venda. Além disso, o atributo Código do Fornecedor também seria uma chave estrangeira para a nova relação criada. Esta forma normal ajuda a diminuir redundâncias de informações criadas indevidamente.

- **3^a Forma Normal (3FN): toda relação deve estar na 2FN e devem-se eliminar dependências funcionais transitivas, ou seja, todo atributo não chave deve ser mutuamente independente.**

Como exemplo, uma relação que contenha os atributos Matrícula do Funcionário (atributo chave), Nome do Funcionário, Código do Departamento e Nome do Departamento não está na Terceira Forma Normal. O Nome do Departamento é dependente do Código do Departamento,

e não da Matrícula do Funcionário. Uma mudança no nome do departamento, por exemplo, levaria a modificações em todos os funcionários daquele departamento.

Para eliminar este problema, cria-se uma nova relação (Departamento) contendo Código do Departamento e Nome do Departamento. Na relação original, retira-se o Nome de Departamento, mantendo-se o Código do Departamento, agora como chave estrangeira. Esta forma normal também ajuda a diminuir redundâncias e aumentar a independência das relações.

Certo.

016. (FCC/INFRAERO/ANALISTA DE SISTEMA/BANCO DE DADOS E ADMINISTRADOR DE DADOS/2011) Em relação à normalização de dados, considere:

- I – Se existir um atributo multivalorado, deve-se criar um novo atributo que individualize a informação multivalorada.
- II – Se existir um atributo não atômico, deve-se dividi-lo em outros atributos que sejam atômicos.
- III – Todos os atributos primos devem depender funcionalmente de toda a chave primária.

Os itens I, II e III referem-se direta e respectivamente a

- a) 1FN, 1FN e 2FN.
- b) 1FN, 2FN e 2FN.
- c) 1FN, 2FN e 3FN.
- d) 2FN, 2FN e 3FN.
- e) 2FN, 3FN e 3FN.



Quanto à questão, os itens I e II estão relacionados à **1FN**. Deve-se garantir que **todo atributo contenha valores atômicos**.

O item III está relacionado à **2FN**, que destaca que **todo atributo que não seja chave deve ser totalmente dependente da chave primária**.

Letra a.

017. (FUNCAB/PRODAM/PROCESSAMENTO DE DADOS AMAZONAS S.A/ANALISTA DE TI-ANALISTA DE BANCO DE DADOS/2014) O objetivo principal de um banco de dados é armazenar informações, com segurança, e disponibilizá-las quando houver uma solicitação qualquer. A relação é um objeto criado no banco de dados que mantém as informações armazenadas e possui o conceito que é:

- a) um subconjunto do produto cartesiano dos atributos-chave da relação.
- b) o próprio conjunto do produto cartesiano dos atributos-chave da relação.
- c) um subconjunto do produto cartesiano dos domínios dos atributos não chave da relação.
- d) o próprio conjunto do produto cartesiano dos domínios dos atributos da relação.
- e) um subconjunto do produto cartesiano dos domínios dos atributos da relação.



A questão tenta confundir o candidato apresentando o conceito de relação na qual os atributos são definidos a partir de um subconjunto dos valores de seus domínios.

Letra e.

018. (FUNCAB/PRODAM/PROCESSAMENTO DE DADOS AMAZONAS S.A/ANALISTA DE TI/ANALISTA DE BANCO DE DADOS/2014) Seja a relação R constituída de n atributos denominados de A, A , A ,..., A . O domínio de um atributo A qualquer uma dessa relação define-se por $\text{dom}(A)$. A indicação do número total de valores, ou cardinalidade, em um domínio de um atributo A qualquer é $|\text{dom}(A)|$, considerando que todos os domínios são finitos. O produto de cardinalidades de todos os domínios, definido da seguinte forma $|\text{dom}(A)| \times |\text{dom}(A)| \times |\text{dom}(A)| \dots |\text{dom}(A)|$, representa:

- a)** o número total de possíveis tuplas da relação R.
- b)** a quantidade de valores de domínios existentes da relação R.
- c)** o número total possíveis de cardinalidades das tuplas da relação R.
- d)** a quantidade de valores de cardinalidades de domínios da relação R.
- e)** a quantidade de possíveis valores de cardinalidades de domínios das tuplas da relação R.



A multiplicação de todos os possíveis valores de todos os atributos de uma tabela define a quantidade de possíveis registros (tuplas). Em geral este valor é muito grande visto que a maioria dos tipos utilizados possui valores bem amplos.

Letra a.

019. (ESAF/STN/DESENV SISTEMAS/2008) Se uma dada variável de relação R possui seus atributos não-chaves mutuamente independentes e irredutivelmente dependentes da chave primária, R está na

- a)** primeira forma normal – 1FN.
- b)** segunda forma normal – 2FN.
- c)** terceira forma normal – 3FN.
- d)** forma normal nula.
- e)** desnormalização.



1ª Forma Normal (1FN): toda relação deve ter uma chave primária e devese garantir que todo atributo seja atômico. Atributos compostos devem ser separados.

2ª Forma Normal (2FN): toda relação deve estar na 1FN e devem-se eliminar dependências funcionais parciais, ou seja, todo atributo não chave deve ser totalmente dependente da chave primária.

3^a Forma Normal (3FN): toda relação deve estar na 2FN e devem-se eliminar dependências funcionais transitivas, ou seja, todo atributo não chave deve ser mutuamente independente.

Letra c.

020. (CETRO/ANVISA/ANALISTA ADMINISTRATIVO/ÁREA 5/2013) Quanto à normalização de banco de dados, marque V para verdadeiro ou F para falso e, em seguida, assinale a alternativa que apresenta a sequência correta.

- I – () É o processo matemático formal fundamentado na teoria dos conjuntos.
- II – () Substitui gradativamente um conjunto de entidades e relacionamentos por outro que diminui ou elimina as anomalias de atualização.
- III – () Permite a dependência transitiva entre atributos.
- IV – () Torna o modelo de dados mais estável, isto é, sujeito a um número menor de manutenções.

- a)** F/ V/ V/ V
b) F/ V/ V/ F
c) V/ F/ F/ F
d) V/ V/ F/ V
e) V/ F/ F/ V



A normalização é um processo progressivo e hierárquico e deve ser feito na ordem: **1FN** → **2FN** → **3NF**, etc. O objetivo é eliminar inconsistência, redundância, anomalias e melhorar a recuperação da informação e um modelo só estará normalizado se passar por todas as formas normais. Em um modelo de dados **não se pode permitir a dependência transitiva entre atributos**. Isso quer dizer que um atributo deve depender exclusivamente da chave e não de outro atributo que não seja a chave.

Um modelo pode estar na 1FN e na 2FN e ainda conter um atributo que depende de outro que não seja a chave. Isso só seria verificado na 3FN que trata exatamente deste tema.

A única alternativa incorreta, portanto, é a III, porque de acordo com a 3^a Forma Normal (3FN) toda relação deve estar na 2FN e **devem-se eliminar dependências funcionais transitivas**, ou seja, todo atributo não chave deve ser mutuamente independente. **Parte superior do formulário Letra d.**

021. (FCC/TJ-AP/ANALISTA JUDICIÁRIO/ÁREA APOIO ESPECIALIZADO/TECNOLOGIA DA INFORMAÇÃO/2014) Considere a tabela de um banco de dados relacional, sendo ID a chave primária, Nome do tipo caractere e Fone do tipo inteiro.

ID	Nome	Fone		
1	Ana	348	269	...
2	Ivo	658	...	
3	Fábio	159	204	...

Essa tabela NÃO está na primeira forma normal porque

- a) contém atributos numéricos e literais em uma única tabela, o que não é permitido.
- b) não contém atributos do tipo data, impedindo a determinação do instante de geração dos registros.
- c) contém uma chave primária simples, o que impede a confirmação do proprietário do banco de dados.
- d) a chave primária é do tipo inteiro, impossibilitando a geração do código do banco de dados.
- e) o atributo Fone contém valores que podem ser divididos, mantendo ainda um significado (não atômicos).



A **normalização** é a aplicação de várias regras ou formas normais para garantir a eficiência no armazenamento e recuperação dos dados. Veja os conceitos das principais formas normais:

- **1FN**: define que **um atributo deve ser atômico (indivisível) e não pode ser multivalorados (composto)**. Isso quer dizer que cada atributo de um registro deve ser único e simples.

Exemplo: um modelo que tenha uma tabela com o atributo **Endereço**: (**Rua, Número, Cidade**) deve ser decomposto. Neste caso, é necessário criar três atributos na tabela Rua, Número e Cidade.

No exemplo da questão, a tabela NÃO está na primeira forma normal porque o atributo Fone contém valores que podem ser divididos, mantendo ainda um significado (não atômicos).

- A **2FN** define que **os atributos de uma tabela que não forem chaves tenham DEPENDÊNCIA FUNCIONAL COMPLETA da chave primária**. Isso quer dizer que se a chave primária da tabela for composta, todos demais atributos (não chave) deve depender de todos os campos que formam a chave e não por parte dela.

Exemplo: imagine uma tabela de VENDA com seus atributos (cod_produto, num_pedido, nome_produto, quantidade, valor_unitario, subtotal). Levando em conta que cod_produto e num_pedido seja a chave primária composta, o atributo nome_produto não depende de toda chave, mas sim de parte dela (somente o cod_produto). Portanto, não está na 2FN! Para normalizar é necessário remover o atributo nome_produto para uma outra tabela (PRODUTO) que se relacione com a tabela VENDA. **TODOS OS ATRIBUTOS QUE NÃO SÃO CHAVE DEVE DEPENDER DA CHAVE POR COMPLETO**.

- A **3FN** define que **não deve haver DEPENDÊNCIA TRANSITIVA entre atributos não chave**. Isso quer dizer que todos os atributos **que não são chave** devem depender somente da chave e não entre si.

Exemplo: Imagine a tabela VENDA do exemplo anterior normalizada (cod_produto, num_pedido, quantidade, valor_unitario, subtotal). Chave primária composta pelos atributos cod_produto, num_pedido. Vejamos o atributo subtotal! ELE NÃO DEPENDE DA CHAVE, mas sim dos atributos quantidade e valor_unitário. **NÃO PODE HAVER DEPENDÊNCIA TRANSITIVA**.

Muito importante: há uma **hierarquia na normalização!** Só estará na 2FN se, e somente se, estiver na 1FN e só estará na 3FN se, e somente se, estiver na 2FN.

Letra e.

022. (CESPE/IPHAN/ANALISTA I/ÁREA 7/2018) Acerca da abordagem relacional, da normalização e do SGBD, entre outros conceitos relativos a banco de dados, julgue o item a seguir. Uma relação está na quarta forma normal (4FN) quando o conteúdo do registro não pode ser mais reconstruído (efetuar join) a partir de outros registros menores extraídos desse registro considerado.



A quarta forma normal (**4FN**) diz respeito à **dependência multivalorada** em uma relação e não a restrições da efetivação de *Joins* como afirma o item na questão proposta.

A utilização de **join** permite a busca de um registro navegando pelas tabelas sem a necessidade de replicar o campo em outra tabela. Quando um campo (atributo) está em outra tabela sem a necessidade de estar na tabela pesquisada esta relação **não** está na 5FN.

Desde modo, **ao referir-se o uso de join para recuperar informações a partir de outros registros** o item da questão refere-se à **5FN** e não à **4FN**.

Errado.

023. (FCC/TST/ANALISTA JUDICIÁRIO/ANÁLISE DE SISTEMAS/2017) Considere, por hipótese, os requisitos abaixo, levantados durante a modelagem de dados:

- I – A entidade Empregado exerce dois papéis: Gerente e Gerenciado. Cada empregado tem um só gerente e um gerente chefia diversos empregados.
- II – Um empregado (entidade Empregado) ocupa somente um cargo (entidade Cargo) que também pode ser ocupado por outros empregados.
- III – Um departamento (entidade Departamento) aloca diversos empregados (entidade Empregado). Um empregado só pode ser alocado em um único departamento.
- IV – Os atributos identificadores das entidades Empregado, Cargo e Departamento são, respectivamente, Matrícula, Código do Cargo e Código do Departamento.

Em um modelo normalizado, são chaves estrangeiras em I, II e III, respectivamente,

- a) Matrícula (do gerenciado), Código do Cargo e Código do Departamento.
- b) Matrícula (do gerente), Matrícula e Código do Departamento.
- c) Matrícula (do gerente), Código do Cargo e Código do Departamento.
- d) Matrícula (do gerenciado), Matrícula e Matrícula.
- e) Matrícula (do gerenciado), Código do Cargo e Matrícula.



Em I, há um **auto-relacionamento** porque um empregado também é um gerente, logo a chave estrangeira é a **Matrícula do gerente**.

Em II, há uma relação entre **EMPREGADO** e **CARGO**. Como um empregado pode ter somente um cargo e um cargo pode ser de vários empregados, a chave estrangeira é **Código do Cargo** que fica na entidade **EMPREGADO**.

Em III, há uma relação entre EMPREGADO e DEPARTAMENTO. Como um empregado pode ser alocado em somente um departamento e um departamento pode ter vários empregados, a chave estrangeira é **Código do Departamento** que fica na entidade EMPREGADO.

Logo, a alternativa correta é **letra C**.

Letra c.

024. (FCC/TRE-AM/PROGRAMADOR DE COMPUTADOR/2010) A Forma Normal Boyce-Codd é considerada uma variação forte da:

- a) 1FN
- b) 2FN
- c) 3FN
- d) 4FN
- e) 5FN



Uma tabela está na **Forma Normal de Boyce-Codd (FNBC ou BCNF)** se, e somente se, **estiver na 3FN e todo determinante é chave candidata**. Dessa forma, **não há dependências entre atributos não chave**. É considerada uma variação forte da 3FN - sendo conhecida, inclusive, como 3,5FN.

Letra c.

025. (FGV/AL-RO/ANALISTA LEGISLATIVO/BANCO DE DADOS/2018) Considere uma tabela relacional R com atributos A, B, C, D, e as seguintes dependências funcionais.

- A → B
- B → C
- A → D
- B → A

Estabelecendo-se que os atributos sublinhados identificam chaves, primárias ou candidatas, o esquema correto para que se obtenha um projeto normalizado até a forma normal Boyce-Codd, é

- a) R (A, B, C, D)
- b) R (A, B, C, D)
- c) R (A, B, C, D)
- d) R1 (A, C, D)
R2 (A, B)
- e) R1 (A, B, D)
R2 (B, C)



Via de regra, **a chave primária é responsável por identificar uma tupla em uma relação, logo a chave primária é a coluna (ou conjunto de colunas) determinante e as outras colunas são dependentes**.

Para estar na BCNF ou FNBC (Forma Normal Boyce-Codd), além da 3FN, todo atributo não chave deve depender funcionalmente diretamente da chave primária, ou seja, não pode haver dependências entre atributos não chave. A FNBC é considerada uma variação forte da 3FN.

Assim, dadas as dependências funcionais:

$A \rightarrow B$

$B \rightarrow C$

$A \rightarrow D$

$B \rightarrow A$



Temos que **A e B são determinantes, assim devem ser chaves da tabela**. Vamos então à análise dos itens:

- a) Certa. Em R (A, B, C, D) **A e B são determinantes e chaves**.
 - b) Errada. R (A, B, C, D) B é também um determinante e, portanto, deveria ser chave.
 - c) Errada. Em (A, B, C, D) A é também um determinante e, portanto, deveria ser chave.
 - d) Errada. R1 (A, C, D)
- R2 (A, B) B é também um determinante e, portanto, deveria ser chave.
- e) Errada. R1 (A, B, D)
- R2 (B, C) B é determinante de C e A e não somente de C.

Letra a.

026. (CESPE/STJ/ANALISTA JUDICIÁRIO-TECNOLOGIA DA INFORMAÇÃO/2008) Acerca do modelo relacional e da normalização de dados, julgue os itens a seguir.

A normalização é um processo no qual são analisados esquemas de relações, com base em dependências funcionais e chaves primárias, visando minimizar redundâncias e anomalias de inserção, exclusão e atualização. Na normalização, se ocorrer a decomposição de uma relação, cada dependência funcional existente antes da decomposição terá de ser representada em alguma relação existente depois da decomposição.



Normalização de dados é uma **série de passos** que se segue no projeto de um BD, em que se analisa relações (tabelas) baseadas em dependências funcionais e chaves primárias, **permitindo um armazenamento consistente e um eficiente acesso aos dados em um BD relacional**. Esses passos reduzem a redundância de dados e as chances dos dados se **tornarem inconsistentes**.

Ainda, é correto destacar que cada dependência funcional se torne uma nova relação após a decomposição pela normalização.

Certo.

027. (CESPE/MPU/TÉCNICO DE INFORMÁTICA/2010) Uma relação que não esteja normalizada possibilita a redundância de dados, o que pode levar à inconsistência de tais dados.



O processo de **normalização** consiste em **decompor as tabelas** com base em um conjunto de regras.

O objetivo é **reduzir a redundância dos dados**, ou seja, evitar que as informações fiquem duplicadas no banco desnecessariamente. Como consequência, evitamos anomalias de inserção, atualização e exclusão. Isso diminui a chance de inconsistências nos dados.

Assim, conforme visto, relações que não estejam normalizadas realmente possibilitam que haja redundância de dados, que pode causar inconsistências dos dados.

Certo.

028. (CESPE/BANCO DA AMAZÔNIA/TÉCNICO CIENTÍFICO/ADMINISTRAÇÃO DE DADOS/2012) Em um banco de dados, a despeito de a normalização reduzir a replicação de dados, ela nem sempre proporciona o bom desempenho das operações de acesso a esses dados.



O processo de **normalização** consiste em **decompor as tabelas** com base em um conjunto de regras. O objetivo é **reduzir a redundância dos dados**, ou seja, evitar que as informações fiquem duplicadas no banco desnecessariamente e, nem sempre proporciona o bom desempenho das operações de acesso a esses dados, isto é, consultas de dados têm geralmente um desempenho pior.

Certo.

029. (CESPE/TRT-AL/APOIO ESPECIALIZADO/ANÁLISE DE SISTEMAS/2014) Os bancos de dados relacionais são uma das principais ferramentas computacionais para armazenamento, organização e exploração da informação. Quanto à modelagem de dados por entidades e relacionamentos, ao modelo relacional e à álgebra que o sustenta, julgue o item que se segue. Uma relação encontra-se na primeira forma normal caso tenham sido eliminadas todas as superchaves dessa relação.



Uma **superchave** é um atributo (ou conjunto de atributos) que não se repete na tabela.

Uma relação estará na **1FN** se não houver atributo representando agrupamento (não atômico) e nem atributo repetitivo (multivalorado). Assim, para estar na 1FN, **todos os atributos** da

relação **devem ser atômicos** (ou indivisíveis). Nesse contexto, não há nenhuma relação com eliminação de superchaves.

Obs.: Vários alunos já me perguntaram a diferença entre uma superchave e uma chave primária! **Lembre-se de que a chave primária é uma superchave mínima**, ou seja, uma superchave da qual não se pode retirar nenhum atributo e manter a propriedade da unicidade. Se uma relação tiver várias superchaves mínimas, cada uma delas é chamada de **chave candidata**, mas apenas uma será escolhida para ser a chave primária.

Errado.

030. (FCC/TRT-2ª/ANALISTA JUDICIÁRIO/TI/2008) Para eliminar a condição de existência de valores não atômicos em uma coluna de tabela relacional,

- a)** deve ser aplicada, no mínimo, a primeira Forma Normal.
- b)** devem ser aplicadas, no mínimo, as quatorze regras de Codd.
- c)** deve ser aplicada, no mínimo, a Forma Normal Boyce-Codd.
- d)** deve ser aplicada, no mínimo, a terceira Forma Normal.
- e)** devem ser aplicadas, no mínimo, as regras de integridade referencial.



a) Certa. **Uma relação estará na 1FN** se não houver atributo representando agrupamento (não atômico) e nem atributo repetitivo (multivalorado). Assim, para estar na 1FN, **todos os atributos** da relação **devem ser atômicos** (ou indivisíveis). Assim, para eliminar a existência de valores não atômicos em uma tabela, ela **deve estar normalizada, pelo menos, até a 1FN**.

b) Errada. Não existe essa quantidade de regras. As **Doze Regras de Codd** são, na verdade, um conjunto de **treze regras** (Elas são enumeradas de zero a doze!).

c) Errada. Não é necessário chegar até a FNBC. **Para estar na BCNF ou FNBC (Forma Normal Boyce-Codd), além da 3FN, todo atributo não chave deve depender funcionalmente diretamente da chave primária**, ou seja, **não pode haver dependências entre atributos não chave**.

d) Errada. Não é necessário chegar até a 3FN.

Uma relação estará na 3ª FN, se e somente se, estiver na 2ª FN e todos os seus atributos não chaves forem dependentes não transitivos da chave primária.

e) Errada. As **regras de integridade referencial** não possuem utilidade para eliminar valores não atômicos.

Letra a.

031. (UNIRIO/2013) Uma relação está na primeira forma normal:

- a)** se e somente se não contém dependências funcionais.
- b)** se e somente se não é uma relação derivada.
- c)** somente antes da primeira fase do processo de normalização.
- d)** se e somente se os valores de seus atributos são atômicos e monovalorados.
- e)** se e somente se não está em nenhuma outra forma normal.



- a) Errada. Uma relação na 1FN poderá conter dependências funcionais.
- b) Errada. O fato de ser uma relação derivada (também chamada de *view*) não possui relação com a 1FN.
- c) Errada. A 1FN é a primeira fase do processo de normalização.
- d) Certa. Uma das estratégias para a melhoria e otimização de um Banco de Dados é o uso da **normalização**. **Uma relação estará na 1FN** se não houver atributo representando agrupamento (não atômico) e nem atributo repetitivo (multivalorado). Assim, para estar na 1FN, **todos os atributos** da relação **devem ser atômicos** (ou indivisíveis).
- e) Errada. Para estar na 1FN, **todos os atributos** da relação **devem ser atômicos** (ou indivisíveis).



- Todos os valores das colunas (todos os **atributos**) são **atômicos (indivisíveis)**.
- **Não** pode possuir atributos **multivalorados** nem **compostos**.

Letra d.

032. (CESPE/TRT-10ª/ANALISTA JUDICIÁRIO/ANÁLISE DE SISTEMAS/2004) A segunda forma normal (2FN) requer que todos os dados dependam de todas as chaves das tabelas. Dependências parciais também são permitidas. Um exemplo de 2FN é: Empregado (nome, endereço, trabalho, salário).



Não são permitidas dependências parciais na **2FN**. Para estar na 2FN:

- está na 1FN.
- **cada atributo não-chave é dependente da chave primária** (ou candidata) **inteira** (Dependência funcional total).
- cada atributo não-chave não pode ser dependente de apenas parte da chave.

Guarde isso:



- Todos os valores das colunas (todos os **atributos**) são **atômicos (indivisíveis)**.
- **Não** pode possuir atributos **multivalorados** nem **compostos**.

- **Está na 1FN e os seus atributos não chaves são dependentes funcionais completos da chave primária.**
- **Cada atributo não chave não poderá ser dependente de apenas parte da chave.**

033. (CESGRANRIO/DECEA/TÉCNICO DE DEFESA AÉREA/ANALISTA DE SISTEMAS/2006) Suponha que todos os atributos de uma relação R contenham apenas valores atômicos. É possível afirmar, com certeza e mais especificamente, que R está na forma normal:

- a) 5FN
- b) 4FN
- c) 3FN
- d) 2FN
- e) 1FN



Se todos os valores das colunas (todos os atributos) são atômicos, então a relação está na **1FN**.



- Todos os valores das colunas (todos os **atributos**) são **atômicos** (**indivisíveis**).
- **Não** pode possuir atributos **multivalorados** nem **compostos**.

Letra e.

034. (CESPE/TRT-10ª/ANALISTA JUDICIÁRIO/ANÁLISE DE SISTEMAS/2004) A terceira forma normal (3FN) requer que os dados das tabelas dependam da chave primária e de uma única chave secundária da tabela. Um exemplo de 3FN é: Empregado (nome, endereço, projeto, localização-projeto).



Uma relação estará na 1FN se não houver atributo representando agrupamento (não atômico) e nem atributo repetitivo (multivalorado).

Uma relação estará na 2FN, se e somente se, estiver na 1FN e os seus atributos **não chaves** forem **dependentes funcionais completos** da chave primária.

Uma relação estará na 3FN, se e somente se, estiver na 2FN e todos os seus atributos **não chaves** forem **dependentes não transitivos** da chave primária.



- **Está na 2FN e todos os seus atributos não chaves são dependentes não transitivos da chave primária.**
- Prega o conceito de dependência transitiva.

Uma tabela está na 3FN se estiver na 2FN e não houver dependência transitiva entre atributos não chave. Assim, conforme visto, os dados da tabela não devem depender de uma única chave secundária, devem depender apenas da chave primária.

Errado.

035. (CESPE/BANCO DA AMAZÔNIA/TÉCNICO CIENTÍFICO/BANCO DE DADOS/2012) Julgue os itens seguintes, a respeito do processo de normalização. Uma relação está na terceira forma normal se ela contém dependências funcionais e também dependências transitivas.



Uma tabela está na 3FN se estiver na 2FN e não houver dependência transitiva entre atributos não chave.



- **Está na 2FN e todos os seus atributos não chaves são dependentes não transitivos da chave primária.**
- Prega o conceito de dependência transitiva.

Errado.

036. (FCC/TRT-15ª/ANALISTA JUDICIÁRIO-TI/2013) Uma dependência funcional transitiva ocorre quando um atributo, além de depender da chave primária da entidade, depende de outro atributo ou conjunto de outros atributos não identificadores da entidade. Um exemplo de dependência transitiva pode ser encontrado em um sistema acadêmico universitário hipotético, no qual em uma entidade “aluno” fosse mantida a informação “escola de origem” e “endereço da escola de origem”. O endereço é dependente da escola, que depende do identificador do aluno. Assim, para normalizar, criamos a entidade escola, contendo nome e endereço (e outros campos necessários), eliminamos esses campos da entidade “aluno”, e finalmente criamos o relacionamento entre aluno e escola.

Estando uma tabela normalizada para a forma normal anterior e, se for aplicado a normalização explicada acima, ela estará na:

- a) quinta forma normal.
- b) quarta forma normal.
- c) segunda forma normal.
- d) sexta forma normal
- e) terceira forma normal.



Uma tabela está na 3FN se estiver na 2FN e não houver dependência transitiva entre atributos não chave.



- Está na 2FN e todos os seus atributos não chaves são dependentes não transitivos da chave primária.
- Prega o conceito de dependência transitiva.

Assim, ao se eliminar as dependências transitivas, a tabela fica na 3FN.

Letra e.

037. (CESPE/TJ-SE/ANALISTA JUDICIÁRIO-BD/2014) Na passagem para a terceira forma normal (3FN), por se tratar de uma dependência transitiva na qual as colunas não-chaves dependem de outras colunas não chaves, será criada uma nova entidade sem a presença de chave primária.



A nova entidade necessitará da presença de chave primária. Está errado também dizer que as colunas não-chave (nesse contexto, todas elas!) dependem de outras colunas não-chave. Isso não é verdade em todos os casos. Para existir problemas com a 3FN precisamos que **apenas um dos** atributos não chave seja funcionalmente dependente de outro atributo não chave.



- Está na 2FN e todos os seus atributos não chaves são dependentes não transitivos da chave primária.
- Prega o conceito de dependência transitiva.

Errado.

038. (FCC/TRT-19^a/ANALISTA DE TI/2011) Para uma tabela estar na FNBC (Forma Normal Boyce-Codd), ela:

- não precisa da normalização 1FN.
- precisa estar somente na 2FN.
- também está normalizada na 3FN.
- tem de estar normalizada até a 4FN
- tem de estar normalizada até a 5FN.



Uma tabela está na **Forma Normal de Boyce-Codd (FNBC ou BCNF)** se, e somente se, **estiver NORMALIZADA na 3FN e todo determinante é chave candidata**. Dessa forma, **não há dependências entre atributos não chave**. É considerada uma variação forte da 3FN - sendo conhecida, inclusive, como 3,5FN.

Obs.: Uma tabela está na FNBC se, e somente se, **estiver normalizada na 3FN** e, para cada dependência $x \rightarrow y$, x deverá ser uma superchave.

Letra c.

039. (COVEST-UFPE/ANALISTA DE TI/SISTEMAS/2019) Quais formas normais lidam com Dependência Funcional Parcial e Dependência Funcional Transitiva, respectivamente?

- a) 2^a e 3^a
- b) 3^a e 2^a
- c) 3^a e 4^a
- d) 4^a e 3^a
- e) 4^a e 5^a



Conforme visto na figura seguinte, a 2FN lida com dependências funcionais parciais e a 3FN lida com dependências funcionais transitivas.



Figura. Formas Normais (QUINTÃO, 2021)

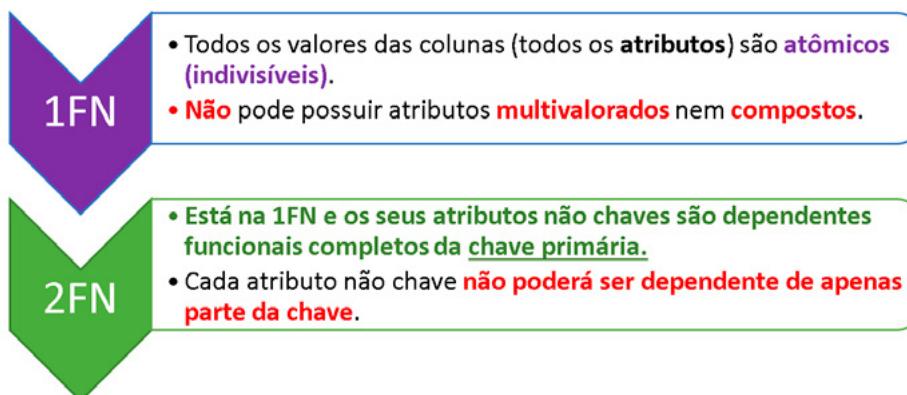
O conteúdo deste livro eletrônico é licenciado para MARIO LUIS DE SOUZA - 41250799864, vedada, por quaisquer meios e a qualquer título, a sua reprodução, cópia, divulgação ou distribuição, sujeitando-se aos infratores à responsabilização civil e criminal.

040. (FCC/TJ-MA/ANALISTA DE SISTEMAS/ DESENVOLVIMENTO/2019) Uma entidade de ligação possui uma chave primária composta pelos atributos que são chaves primárias nas entidades ligadas a ela. A verificação para saber se os atributos não chave são dependentes total ou parcialmente da chave primária composta, com objetivo de eliminar as dependências funcionais parciais, é feita na:

- a) 5FN**
- b) 4FN**
- c) 3FN**
- d) 1FN**
- e) 2FN**



A verificação para saber se os atributos não chave são dependentes total ou parcialmente da chave primária composta, com objetivo de eliminar as dependências funcionais parciais, é feita na **2FN**.



Assim, caso uma tabela esteja na 1FN e não tenha dependências funcionais parciais (tenha apenas dependências funcionais totais), ela estará na **2FN**.

Letra e.

GABARITO

- 1. b
- 2. c
- 3. c
- 4. d
- 5. e
- 6. a
- 7. b
- 8. c
- 9. b
- 10. E
- 11. b
- 12. c
- 13. c
- 14. b
- 15. C
- 16. a
- 17. e
- 18. a
- 19. c
- 20. d
- 21. e
- 22. E
- 23. c
- 24. c
- 25. a
- 26. C
- 27. C
- 28. C
- 29. E
- 30. a
- 31. d
- 32. E
- 33. e
- 34. E
- 35. E
- 36. e
- 37. E
- 38. c
- 39. a
- 40. e

REFERÊNCIAS

BRAGA, Regina. **Notas de aula**, UFJF, 2012.

ELMASRI, R.; NAVATHE, S. B. **Sistemas de Banco de Dados**. 6. ed. São Paulo: Pearson Addison Wesley, 2011.

HEUSER, Carlos Alberto. **Projeto de banco de dados**. 4. ed. Porto Alegre: Sagra, 2001.

HERNANDEZ, Michael J. **Aprenda a projetar seu próprio banco de dados**. Tradução Patrizia Tallia Parenti. São Paulo: Makron, 2000.

KORTH, Henry F.; SILBERSCHATZ, Abraham. **Sistema de banco de dados**. Tradução Mauricio Heihachiro Galvan Abe. 6. ed. São Paulo: Makron, 2011.

MACHADO, Felipe Nery Rodrigues; ABREU, Maurício Pereira de. **Projeto de banco de dados: uma visão prática**. 6. ed. São Paulo: Érica, 2000.

QUINTÃO, P. L. **Notas de aula da disciplina “Tecnologia da Informação”**. 2021.

ROB, P.; CORONEL, C. **Sistemas de Banco de Dados Projeto, Implementação e Gerenciamento**. 2011.

SETZER, Valdemar W. **Banco de dados: conceitos, modelos, gerenciadores, projeto lógico, projeto físico**. 3. ed. rev. São Paulo: E. Blücher, 2002.

SETZER, http://www.ime.usp.br/~vwsetzer/dado-info.html_2001.

TAKAI, O.K.; ITALIANO,I.C.; FERREIRA, E.F. **Introdução a Banco de Dados**.

Patrícia Quintão



Mestre em Engenharia de Sistemas e computação pela COPPE/UFRJ, Especialista em Gerência de Informática e Bacharel em Informática pela UFV. Atualmente é professora no Gran Cursos Online; Analista Legislativo (Área de Governança de TI), na Assembleia Legislativa de MG; Escritora e Personal & Professional Coach.

Atua como professora de Cursinhos e Faculdades, na área de Tecnologia da Informação, desde 2008. É membro: da Sociedade Brasileira de Coaching, do PMI, da ISACA, da Comissão de Estudo de Técnicas de Segurança (CE-21:027.00) da ABNT, responsável pela elaboração das normas brasileiras sobre gestão da Segurança da Informação.

Autora dos livros: Informática FCC - Questões comentadas e organizadas por assunto, 3^a. edição e 1001 questões comentadas de informática (Cespe/UnB), 2^a. edição, pela Editora Gen/Método.

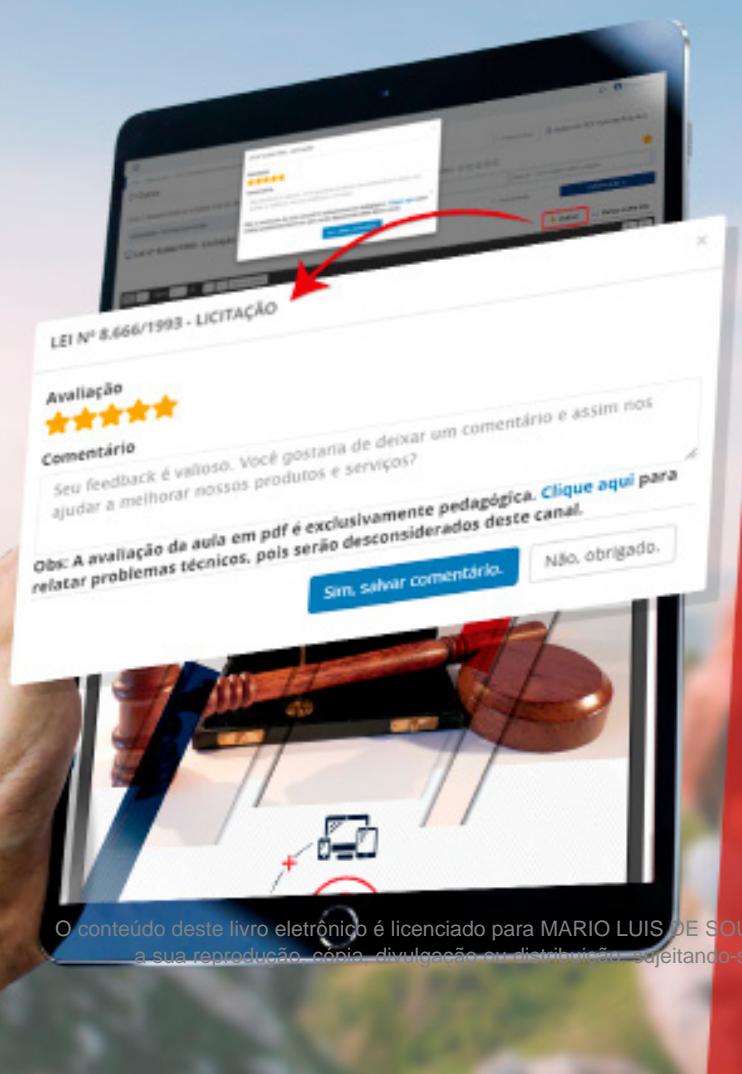
Foi aprovada nos seguintes concursos: Analista Legislativo, na especialidade de Administração de Rede, na Assembleia Legislativa do Estado de MG; Professora titular do Departamento de Ciência da Computação do Instituto Federal de Educação, Ciência e Tecnologia; Professora substituta do DCC da UFJF; Analista de TI/Suporte, PRODABEL; Analista do Ministério Público MG; Analista de Sistemas, DATAPREV, Segurança da Informação; Analista de Sistemas, INFRAERO; Analista - TIC, PRODEMGE; Analista de Sistemas, Prefeitura de Juiz de Fora; Analista de Sistemas, SERPRO; Analista Judiciário (Informática), TRF 2^a Região RJ/ES, etc.

@coachpatriciaquintao

/profapatriciaquintao

@plquintao

t.me/coachpatriciaquintao



NÃO SE ESQUEÇA DE AVALIAR ESTA AULA!

SUA OPINIÃO É MUITO IMPORTANTE PARA MELHORARMOS AINDA MAIS NOSSOS MATERIAIS.

ESPERAMOS QUE TENHA GOSTADO DESTA AULA!

PARA AVALIAR, BASTA CLICAR EM LER A AULA E, DEPOIS, EM AVALIAR AULA.

AVALIAR