

Unidade 1

Banco de dados conceitos e comparação



Abertura

Olá, amigo(a) discente! Seja bem-vindo(a)!

Diante da existência de uma grande demanda por desenvolvimento de software, diversas empresas vêm adotando diversos bancos de dados para conseguir atender a essas demandas. A boa aplicação e construção do banco de dados resulta em um software com alto desempenho. Nesta unidade vamos compreender e discutir os principais conceitos que envolvem banco de dados.

Na aula 01, vamos fazer um comparativo entre o banco de dados relacional e o não relacional. Na aula 02, dando continuidade iremos relembrar alguns conceitos e fundamentos importantes de banco de dados.

Objetivos

- Ter uma visão geral das diferenças entre os bancos de dados relacionais e não relacionais;
- Relembrar os conceitos e fundamentos relacionados a banco de dados.

Conteúdo Programático

Aula 01 – Conhecendo os bancos de dados relacionais e suas diferenças com banco de dados não relacionais

Aula 02 – Conceitos e fundamentos de banco de dados.



Videoaula Apresentação da Disciplina

Utilize o QR Code para assistir!

Assista!





Videoaula Minicurrículo

Utilize o QR Code para assistir!

Assista!



Banco de dados relacional x Banco de dados não relacional

Olá, estudante, como vai? Vamos começar nossos estudos fazendo um comparativo entre o banco de dados relacional e o não relacional. Você deve estar se perguntando: para que eu tenho que estudar isso? Às vezes, você pode ser surpreendido por projetos que demandam ser desenvolvidos por banco de dados relacionais ou não relacionais. Saber a diferença entre eles favorece no momento de tomar decisões técnicas dentro de cada projeto.

Hoje em dia, para qualquer aplicação que queremos desenvolver precisamos de um banco de dados. Em meio a esse processo, devemos ficar atentos a algumas especificações, para assim melhor avaliar e tomar a decisão correta sobre qual tipo de banco de dados utilizar. Deste modo, é preciso saber as diferenças entre os bancos de dados relacionais e os bancos de dados não relacionais.



Videoaula 1

Utilize o QR Code para assistir!

Assista o vídeo para entender melhor o assunto.



Os bancos de dados surgem por volta da década de 60 com a necessidade de armazenar e manipular os dados dentro dos sistemas, ocorrendo uma grande adesão do banco de dados relacional a partir de 1980.

Banco de dados precisam se atentar em alguns pilares, são eles:

- Segurança
- Desempenho
- Disponibilidade

Banco de Dados Relacional

Em banco de dados relacionais os dados são armazenados de acordo com um esquema específico, desse modo, cada tabela se relaciona com outras tabelas de acordo com a necessidade.

O banco de dados relacional é organizado da seguinte maneira:

- **Tabela:** Forma onde se encontram organizados os dados, uma tabela pode ou não se relacionar com outras tabelas. A tabela é o conjunto de dados. Dentro dela que se encontram os demais itens.
- **Coluna:** A coluna fica dentro da tabela, sendo ela responsável por identificar a descrição de cada informação contida na linha.
- **Linha:** A linha é onde se encontram os dados armazenados, para saber qual dado está armazenado na linha é necessário verificar a descrição (nome) da coluna.
- **Chave:** A chave é um conceito existente dentro de um banco de dados relacional onde se submete ao indicar os dados que serão utilizados em um relacionamento, podendo existir 3 tipos de chaves dentro de um banco de dados relacional: chave primária, chave estrangeira, chave composta (podendo essa ser uma chave primária).
- **Índice:** O índice funciona como uma referência da tabela para acesso mais rápido dos dados, em alguns bancos de dados relacionais as chaves são tratadas como índices para facilitar a consulta.
- **Relacionamento:** São associações criadas entre as tabelas utilizando instruções e chaves para assim tratar e recuperar dados.

A estrutura de uma tabela em um banco de dados relacional é parecida com a base de uma planilha.

Aluno		
id_aluno	nome	inscrição
1	Almeida	123
2	Souza	321
3	Silva	465

Na tabela 1, pode-se observar a estrutura de uma tabela de um banco relacional com base em uma planilha, sendo a tabela aluno com as colunas "id_aluno", nome e inscrição, dessa forma, nas linhas seguintes se encontram os dados presentes na tabela.

Com essa estrutura, os bancos de dados relacionais favorecem a consistência de dados, dessa forma, tornando o dado sempre disponível e verdadeiro. Isso ocorre, pois em um banco relacional é possível existir diversas instâncias do mesmo banco.

A instância é como uma cópia do mesmo banco, desse modo, favorece a disponibilidade dos dados para quem precisa dessas informações. Outra característica marcante de um banco de dados relacional é o compromisso e a atomicidade. Isso ocorre, pois é possível aplicar regras de negócio dentro do próprio banco de dados relacional. Dessa forma, tratando as informações e realizando, junções, consultas e tratando os dados em um nível mais granular.

Em alguns casos, construir um banco de dados relacional que trate algumas regras de negócio pode resultar em demora para a consulta e aplicação dessas regras. O que resulta em atraso para o usuário e até mesmo demora para carregar informações nos sistemas.

Como o banco de dados relacional é acessado por mais de um usuário ao mesmo tempo e possui seus relacionamentos e regras de negócio que podem ser atribuídos, é preciso se preocupar com a concorrência de utilização.

Isso acontece quando dois ou mais usuários tentam acessar a mesma informação no banco de dados de forma simultânea, então, existem alguns conceitos que auxiliam a organizar a concorrência, são eles:

- Atomicidade: preciso definir quais são os elementos presentes em cada transação que será realizada.
- Consistência: É preciso manter os dados em um estado fixo para que seja possível realizar uma transação com segurança.
- Isolamento: Manter a transação invisível para os outros usuários até que ela seja confirmada.
- Durabilidade: garantir que quando uma transação é realizada os seus atos se tornem permanentes após serem confirmadas.

Vale lembrar que uma transação são alterações que são realizadas dentro de um banco de dados relacional, dessa forma, são operações executadas dentro do banco de dados, as transações realizadas normalmente são operações de:

- Busca
- Criação
- Atualização
- Deleção

Assim, cada banco de dados relacional trata as transações de uma forma, porém, em sua maioria, a tabela fica travada durante uma transação para assim garantir a consistência dos dados armazenados.

Os bancos de dados relacionais mais utilizados são:

- SQL Server
- Oracle
- DB2
- MySQL
- PostgreSQL



Videoaula 2

Utilize o QR Code para assistir!

Assista ao vídeo para entender melhor o assunto.



Banco de Dados Não Relacional

O banco de dados não relacional ou NoSQL, é um modelo de banco de dados onde não se possui um esquema rigidamente estruturado. Contudo, os aspectos são diferentes de um banco relacional.

Os bancos NoSQL aumentam o desempenho em consultas para recuperação dos dados. Por não possuir um esquema rígido, isso faz com que o banco de dados não relacional favoreça no desenvolvimento de sistemas onde não se tem muita clareza das regras de negócio. Sendo utilizado em diversos projetos ágeis por conta dessas características.

Indicação de Vídeo

NoSQL em 5 minutos (Banco de dados não-relacionais).

Os bancos de dados não relacionais armazenam os dados de formas diferentes, sendo elas: <https://youtu.be/Dnfcluqa0xo>

Documentos

É considerado um modelo de banco NoSQL bem versátil, pois com ele é possível que os campos de cada registro sejam de tipos diferentes. Esses formatos de documentos normalmente trabalham com documentos JSON para armazenar os dados, desse modo, favorecendo a leitura das informações armazenadas.

Figura 1. Modelo Documento JSON

```
1  {  
2    "id_pessoa": 1,  
3    "nome": "Banco de dados Avançado",  
4    "cpf": "822.324.006-52",  
5    "departamento": "COMERCIAL",  
6    "email": "banco.dados.avancado@unifil.com.br",  
7    "senha": "*****"  
8  }
```

Fonte: Autor, 2021.

Na figura 1, se descreve um exemplo de um documento no formato JSON que pode ser armazenado dentro de um banco de dados não relacional que trabalha com o tipo de armazenamento de documentos.

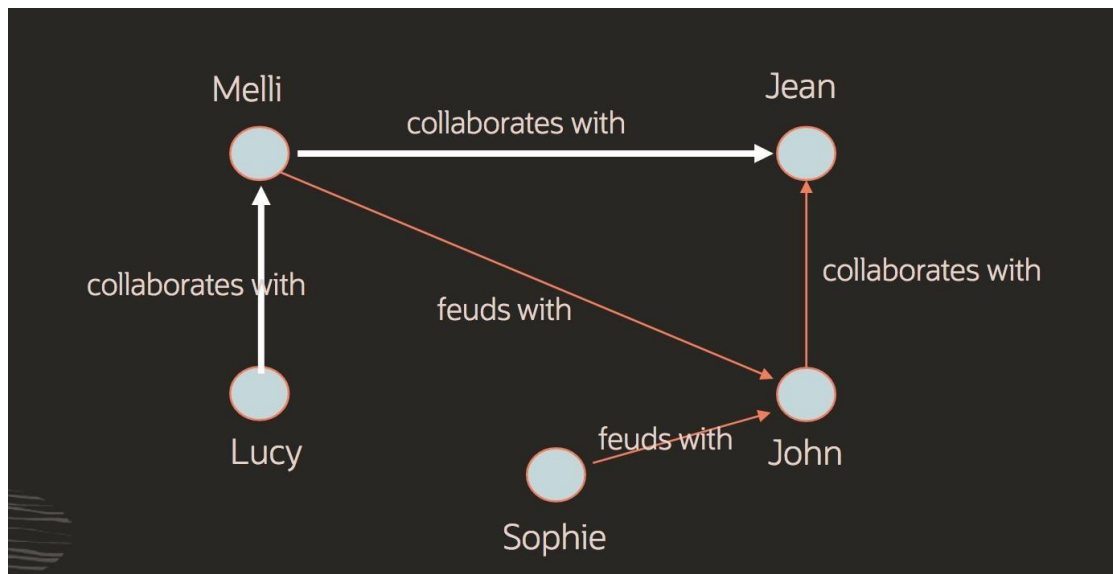
Grafos

O modelo baseado em grafos consiste em armazenar os dados de modo gráfico, possui 3 componentes essenciais:

- Nós: que são os vértices do grafo
- Relacionamento: são as arestas
- Propriedades: que são os atributos tanto dos vértices quanto das arestas

Nesse conceito, o banco de dados pode ser visto como um multigrafo que possui suas placas e direções onde cada nó pode possuir uma relação com uma ou mais arestas.

Figura 2. Modelo Banco de dados grafo



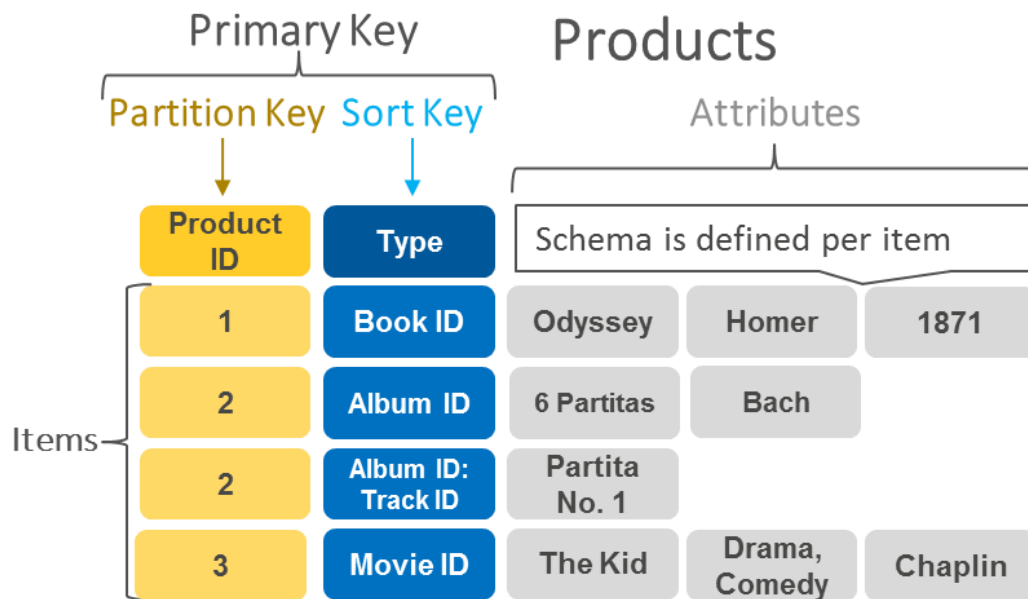
Fonte: Oracle, 2021.

Na figura 2, pode-se observar um exemplo do banco de dados grafo, onde cada vértice é uma pessoa como (Lucy / Melli / Jean) e seus relacionamentos estão descritos em cima de cada aresta, essa é uma forma gráfica de representar um banco de dados em grafo.

Par de chave valor

Bancos de dados onde o modelo é baseado em pares de chave-valor apresentam uma estrutura definida onde as chaves são mapeadas em seu objeto e não obrigatoriamente precisa possuir dados fortemente definidos e estruturais, o interessante é que neste modelo as consultas só podem ser realizadas por intermédio do campo chave.

Figura 3. Modelo Banco de dados Chave-valor



Fonte: AWS, 2021.

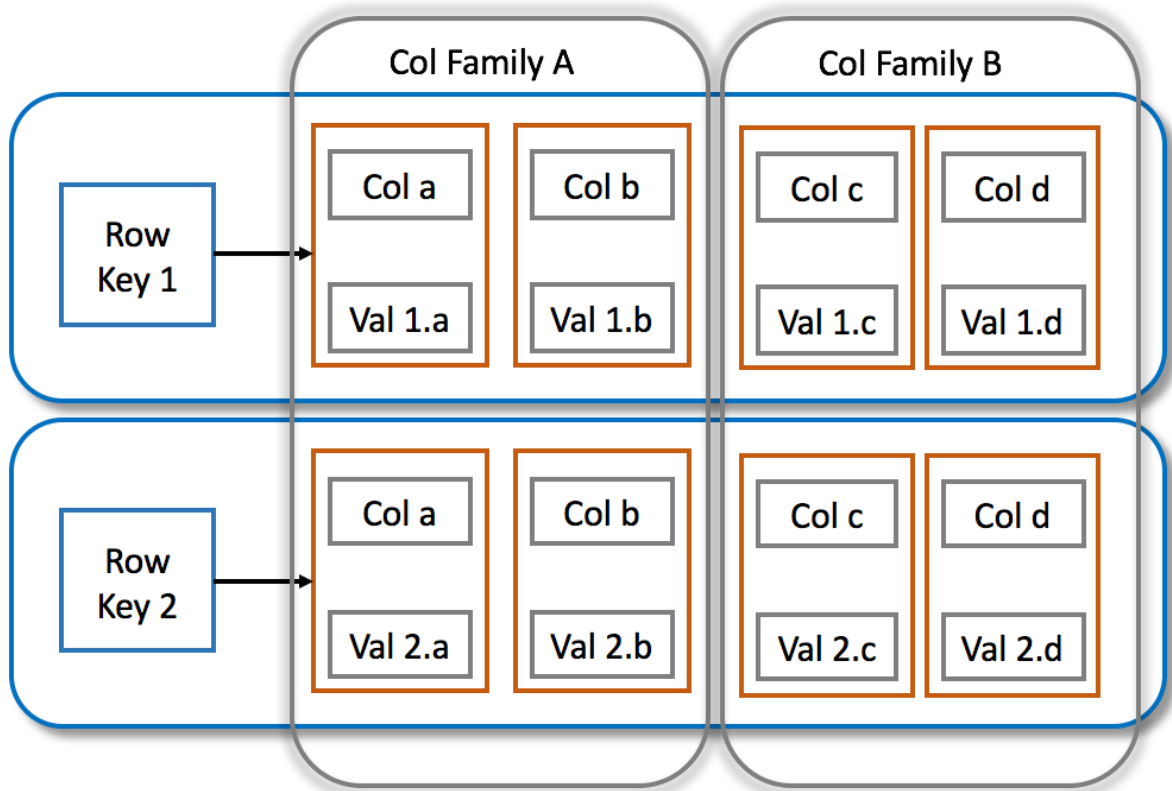
Na figura 3, observa-se um exemplo gráfico de um banco de dados que trabalha com chave-valor, no caso em questão o Amazon DynamoDB, onde os pares de chave e valor cada dado está vinculado a uma chave e para pesquisar o valor é necessário obrigatoriamente possuir a chave.

Colunas grandes

Este modelo de banco de dados possui uma estrutura parecida com um banco de dados relacional onde possui linhas e colunas, mas com grandes colunas é possível armazenar uma quantidade muito maior de dados, o que auxilia em sistemas muito grandes.

Como em outros modelos de banco NoSQL não é necessário possuir uma grande gama de dados estruturados.

Figura 4. Modelo Banco de dados Wide Column



Fonte: Akshay Pore.

Analisando a figura 4, observa-se um exemplo da estrutura de um banco de dados com grandes colunas ou *Wide Column*, onde cada linha está relacionada a uma grande quantidade de dados e cada coluna possui diversas informações.



Videoaula 3

Utilize o QR Code para assistir!

Assista ao vídeo para entender melhor o assunto.



Utilizando de um banco de dados relacional ou não relacional, para se desenvolver um sistema é necessário conhecer a necessidade do cliente, analisar os recursos e com base nessas informações decidir qual a melhor forma de banco de dados irá atender de maneira mais adequada às necessidades do sistema que precisa ser desenvolvido.

Banco de dados conceitos e fundamentos

Olá, aluno, vamos começar a nossa segunda aula desta unidade? Nesta aula vamos conhecer e se aprofundar melhor nos conceitos e fundamentos dos bancos de dados relacionais, para que mais à frente na disciplina você seja capaz de colocar em prática e reconhecer esses conceitos e fundamentos.



Videoaula 1

Utilize o QR Code para assistir!

Assista ao vídeo para entender melhor o assunto.



Os bancos relacionais utilizam de comandos para realizar operações dentro da sua estrutura, esses comandos são na linguagem SQL (Structured Query Language), em português, Linguagem de Consulta Estruturada, é a responsável por fazer as operações entre o Sistema Gerenciador de Banco de Dados (SGBD) e o Banco de Dados (BD).

Os comandos SQL consistem em quatro grupos, sendo eles:

DML (Data Manipulation Language)

São os comandos relacionados a manipulação do conteúdo das tabelas no banco de dados, são esses comandos:

UPDATE

Figura 5. Comando UPDATE

```
1 UPDATE aluno SET nome = 'Aluno 1 teste' WHERE id_aluno = 1;
```

Fonte: O autor, 2021.

O comando UPDATE é utilizado para atualizar um registro no banco de dados, na figura 5 o texto 'aluno' faz referência a tabela onde estão os registros que serão alterados, SET indica quais colunas devem ser alteradas e o valor e o WHERE indica qual registro da tabela deve ser alterado, sendo uma clausura muito importante dentro da operação UPDATE.

DELETE

Figura 6. Comando DELETE

```
1 DELETE FROM aluno WHERE id_aluno = 2;
```

Fonte: O autor, 2021.

O comando DELETE é utilizado para realizar a exclusão de um registro na tabela, no caso da figura 6 o script em questão busca realizar a exclusão do registro com 'id_aluno' 2 na tabela 'aluno'.

INSERT

Figura 7. Comando INSERT

```
1 insert into aluno (id_aluno, nome, idade, email)
2 values (2, 'Aluno 2', 25, 'aluno.2@unifil.com.br');
```

Fonte: O autor, 2021.

Para inserir um novo registro em uma tabela utiliza-se do comando INSERT, conforme demonstra a figura 7, na tabela 'aluno' serão incluídos os dados presentes dentro dos parênteses após a sinalização VALUES, na figura em questão são especificados a ordem das colunas que serão inseridos os valores.

SELECT

Figura 8. Comando SELECT

```
1 SELECT * FROM aluno;
2
3 SELECT * FROM aluno WHERE id_aluno = 1;
```

Fonte: O autor, 2021.

O comando SELECT é utilizado para buscar os dados de uma tabela, no caso da figura 8 o '*' indica que o retorno dessa query deve ser todas as colunas da tabela aluno, a segunda query presente na figura indica que a busca na tabela 'aluno' é somente para onde o 'id_aluno' for 1;

DDL (Data Definition Language)

São os comandos relacionados a definição e criação das estruturas das tabelas e de cada banco de dados.

CREATE

Figura 9. Comando CREATE

```
1 create table aluno (  
2     id_aluno serial primary key,  
3     nome varchar(255),  
4     idade int,  
5     email varchar(255)  
6 );
```

Fonte: O autor, 2021.

O comando CREATE é utilizado para criar um banco de dados e para criar uma tabela, a figura 9 demonstra como é aplicado o comando para a criação de uma tabela aluno, sendo suas colunas:

- **id_aluno:** que é a chave primária da tabela
- **nome:** tipo texto (varchar)
- **idade:** tipo inteiro (int)
- **email:** tipo texto (varchar)

ALTER

Figura 10. Comando ALTER

```
8 ALTER TABLE aluno ALTER COLUMN email TYPE varchar(250);
```

Fonte: O autor, 2021.

O comando ALTER é utilizado para realizar alterações na estrutura do banco de dados ou das tabelas, na figura 10 é demonstrada uma alteração na coluna 'e-mail' dentro da tabela 'aluno' onde o tipo do campo continua sendo varchar, porém com tamanho diferente.

DROP

Figura 11. Comando DROP

```
10 DROP TABLE aluno;
```

Fonte: O autor, 2021.

Para realizar a exclusão de uma tabela ou banco de dados sem chances de recuperação a não ser que exista um backup é utilizado o comando DROP, na figura 11 se demonstra a exclusão da tabela 'aluno'.



Videoaula 2

Utilize o QR Code para assistir!

Assista ao vídeo para entender melhor o assunto.



DCL (Data Control Language)

São os comandos relativos às permissões de cada usuário dentro do banco de dados, como o próprio nome diz, é sobre o controle de acesso aos dados, sendo os comandos mais utilizados:

- GRANT
- REVOKE
- DENY

DQL (Data Query Language)

Alguns autores vão pontuar que no DQL se restringe somente ao comando SELECT porém já foi pontuado nessa aula que este pertence aos comandos DML. Sendo assim, os comandos DQL são relativos a tudo que permite filtrar e selecionar os dados que deseja buscar, sendo eles:

- WHERE
- WHEN
- AND
- IF

Figura 12. Comandos DQL

```
1 SELECT * FROM aluno WHERE email = 'alunos.1@unifil.com.br';  
2  
3 SELECT * FROM aluno WHERE nome = 'aluno teste' and idade = 25;|
```

Fonte: O autor, 2021.

Na figura 12, pode-se observar a sintaxe de utilização de alguns comandos DQL em uma query onde se executa uma busca dentro da tabela 'aluno'.

FORMAS NORMAIS

Aplicar a normalização em um banco de dados relacional auxilia na organização do BD e em sua estrutura, facilitando futuras alterações ou possíveis manutenções que possam chegar a surgir com o tempo.

Primeira Forma Normal

Essa forma normal defende que os atributos, ou seja, cada coluna possua valores atômicos, não sendo multivalorados.

Figura 13. Tabela fora da primeira forma normal

nome	telefone	endereço
Aluno 1	43998092955, 4333281966	Av. Rio Branco, 123 - Centro
Aluno 2	55998098522, 4333281965	Av. Rio Branco, 1552 - Centro

Fonte: O autor, 2021.

A figura 13 representa uma tabela divergente com a primeira forma normal, onde na coluna telefone e na coluna endereço os dados são multivalorados, além de não possuir uma chave primária na tabela.

Para ficar de acordo com primeira forma normal a tabela 'Aluno' deve ficar assim:

Figura 13.1. Tabela primeira forma normal

id_aluno	nome	telefone1	telefone2	endereco	id_bairro
1	Aluno 1	43998092955	4333281966	Av. Rio Branco, 123	1
2	Aluno 2	55998098522	4333281965	Av. Rio Branco, 1552	1

Fonte: O autor, 2021.

Na figura 13.1, é possível visualizar a mesma tabela relatada na figura 13, porém de acordo com a primeira forma normal.

Segunda Forma Normal

Na segunda forma normal defende que todos os atributos de uma tabela, devem ser dependentes somente da chave primária da tabela de modo completo e não devem existir atributos dependentes de outros atributos.

Figura 14. Tabela fora da segunda forma normal

id_produto	nome	valor	fornecedor	endereco_fornecedor	contato_fornecedor
1	Celular	890,00	Motorola	Av. Paulista - SP	11998296535

Fonte: O autor, 2021.

A figura 14 demonstra como é uma tabela em um banco de dados relacional sem a aplicação da segunda forma normal, onde se possui atributos relacionados ao fornecedor, porém a chave primária é do produto, ou seja, existem atributos que não são dependentes só da chave primária.

Para ficar de acordo com a segunda forma normal a tabela 'Produto' ficaria assim:

Figura 14.1 Tabela segunda forma normal

id_produto	nome	valor	id_fornecedor
1	Celular	890,00	36

Fonte: O autor, 2021.

Na figura 14.1 se visualiza como fica a tabela 'Produto' aplicando a segunda forma normal, onde os dados do fornecedor vão para um tabela chamada 'Fornecedor' e assim dentro da tabela 'Produto' o fornecedor se torna uma chave estrangeira, criando assim um relacionamento dentro do banco de dados.

Terceira Forma Normal

Para implementar a terceira forma normal a tabela já precisa estar na primeira e na segunda forma normal.

Desta forma é defendido que cada atributo não chave que é determinante na relação dos dados deve se tornar uma tabela apartada e assim existir um relacionamento entre essas tabelas.

Figura 15. Tabela fora da terceira forma normal

id_venda	id_produto	nota_fiscal	valor	codigo_vendedor	vendedor
1	856	192235666	890,00	1572169	Marcus

Fonte: O autor, 2021.

A figura 15 demonstra a tabela 'Venda' que está fora da terceira normalização, pois os atributos 'codigo_vendedor' e 'vendedor' são atributos determinantes no relacionamento de dados, dessa forma, devem estar presentes em outra tabela.

Figura 15.1 Tabela terceira forma normal

id_venda	id_produto	nota_fiscal	valor	id_vendedor
1	856	192235666	890,00	1572169

Fonte: O autor, 2021.

Como pode-se observar, a figura 15.1 demonstra a tabela 'Venda' aplicando a terceira forma normal, onde foram reconhecidos os atributos que são determinantes no relacionamento dos dados. Então, criando uma tabela 'Vendedor' onde a sua chave primária se torna chave estrangeira dentro da tabela 'Venda'.

A boa aplicação da normalização em banco de dados relacionais é essencial para a construção de um sistema bem organizado e com as funcionalidades corretas.

Desse modo, essas práticas devem ser consideradas ainda no momento de modelagem de um banco de dados, e até mesmo em suas diagramações. Tornar esses relacionamentos e mapear os atributos corretamente ainda na modelagem possibilitam o desenvolvimento mais claro e bem definido entre todas as partes interessadas.



Videoaula 3

Utilize o QR Code para assistir!

Descrição do Professor...



Videoaulas

Prezado(a) aluno(a)!

Você também poderá encontrar todas as videoaulas, clicando em “**Módulos**” no “**Menu Lateral**” e acessar a página de vídeos.

Encerramento

Chegamos ao final da nossa primeira unidade, onde você pode relembrar alguns conceitos e fundamentos de banco de dados. Com certeza, muito do que foi mostrado aqui você já possuía um conhecimento prévio ou até mesmo já havia aplicado. Espero que você tenha aprendido um pouco mais sobre todo esse contexto de banco de dados.

No mundo dos softwares há muitos bancos de dados e seus SGBDs que podem ser melhores ou piores, no final tudo depende da demanda e do produto que deve ser entregue. Na aula 1, visualizamos alguns conceitos entre banco de dados relacionais e não relacionais e assim demonstrando um comparativo entre esses.

Na aula 2, foi relembrado alguns fundamentos e conceitos sobre o banco de dados relacionais, com seus comandos e práticas e a importância desses fundamentos no momento de modelar um software.

Juntando as duas aulas teremos um embasamento básico e teórico sobre banco de dados. Caro aluno, pense um pouco em como os conhecimentos desta unidade podem te auxiliar em seu dia a dia. Até a próxima.

Referências

CUNHA, J.L.M. **Consultoria Organizacional** [Livro eletrônico]. Curitiba: Intersaberes, 2013. (Disponível na biblioteca digital).

OLIVEIRA, D.P.R. **Manual de consultoria empresarial**: conceitos, metodologia e práticas. São Paulo: Atlas, 2004.

SOUZA, O.G.S (org.). **Consultoria Empresarial** [Livro eletrônico]. São Paulo: Pearson Education, 2016. (Disponível na biblioteca digital)

BLOCK, P. **Consultoria: o desafio da liberdade**. São Paulo: McGraw-Hill, 1991.

CUNHA, J.L.M. **Consultoria Organizacional** [Livro eletrônico]. Curitiba: Intersaberes, 2013.

OLIVEIRA, D.P.R. **Manual de consultoria empresarial**: conceitos, metodologia e práticas. São Paulo: Atlas, 2004.

SOUZA, O.G.S (org.). **Consultoria Empresarial** [Livro eletrônico]. São Paulo: Pearson Education, 2016.

CHIAVENATO, Idalberto. **Administração de recursos humanos**. 6. ed. SÃO PAULO: Atlas, 2006.

CROCCO, L.; GUTTMANN, E. **Consultoria empresarial**. São Paulo, Ed. Saraiva, 2005.

CUNHA, J.L.M. **Consultoria Organizacional** [Livro eletrônico]. Curitiba: Intersaberes, 2013.

OLIVEIRA, D.P.R. **Manual de consultoria empresarial**: conceitos, metodologia e práticas. São Paulo: Atlas, 2004.

SOUZA, O.G.S (org.). **Consultoria Empresarial** [Livro eletrônico]. São Paulo: Pearson Education, 2016.



UNIFIL.BR