

UNIVERSIDADE DE BRASÍLIA

Faculdade do Gama

Sistemas de Banco de Dados 2

**Tecnologias de Banco de Dados (TI-BD)**

**Bancos de Dados Paralelos**

**Pedro Vitor Augusto de Jesus - 20/0073249**

Brasília, DF

2023

## **A) Definição Bancos de Dados Paralelos**

O Paralelismo em Bancos de Dados é um conceito fundamental que se insere no contexto de sistemas distribuídos. Nesse cenário, diversos nós compartilham recursos, como processadores, memória e discos, conectados por uma rede de alta velocidade. Essa abordagem visa otimizar o desempenho dos sistemas, especialmente no que diz respeito ao processamento de entradas e saídas (E/S). À medida que as máquinas paralelas se tornam cada vez mais comuns, desempenham um papel essencial no avanço dos estudos relacionados aos sistemas de bancos de dados paralelos.

Um banco de dados paralelo se refere a um sistema em que várias operações de consulta ou processamento de dados podem ser executadas simultaneamente, geralmente usando múltiplos processadores ou nós de processamento. A ideia é dividir uma única tarefa em partes menores que podem ser processadas em paralelo para melhorar o desempenho. Essas operações paralelas podem ocorrer em um único servidor ou em um cluster de servidores. (M. Tamer Özsu & Patrick Valduriez, “Principles of Distributed Database Systems”).

Para os casos onde as operações ocorrem em cluster temos de entender sobre banco de dados distribuídos

O que é um sistema de bancos de dados distribuído?

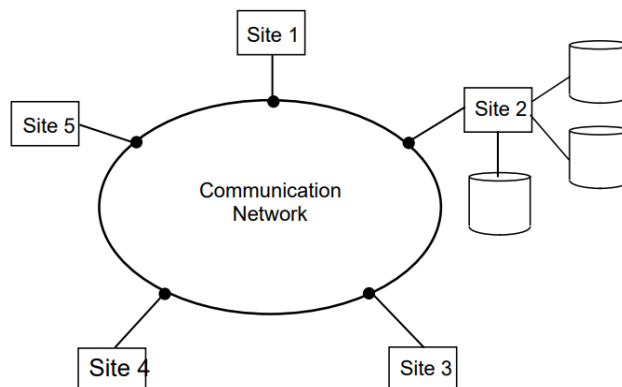
Definimos um banco de dados distribuído como uma coleção de vários bancos de dados logicamente inter-relacionados, distribuídos por uma rede de computadores. Um sistema de gerenciamento de banco de dados distribuído (SGBD distribuído) é então definido como o sistema de software que permite o gerenciamento do banco de dados distribuído e torna a distribuição transparente para os usuários.

Um SGBDD não é uma “coleção de arquivos” que pode ser armazenada

individualmente em cada nó de uma rede de computadores. Para formar um SGBDD , os arquivos não devem apenas estar relacionados logicamente, mas devem estar estruturados entre os arquivos, e o acesso deve ser feito através de uma interface comum. Devemos observar que tem havido muita atividade recente no fornecimento de funcionalidades de SGBD sobre dados semi estruturados que são armazenados em arquivos na Internet (como páginas da Web). À luz desta atividade, o requisito acima pode parecer desnecessariamente rigoroso. No entanto, é importante fazer uma distinção entre um SGBDD onde este requisito é atendido e sistemas de gerenciamento de dados distribuídos mais gerais que fornecem um acesso “semelhante ao SGBD” aos dados.

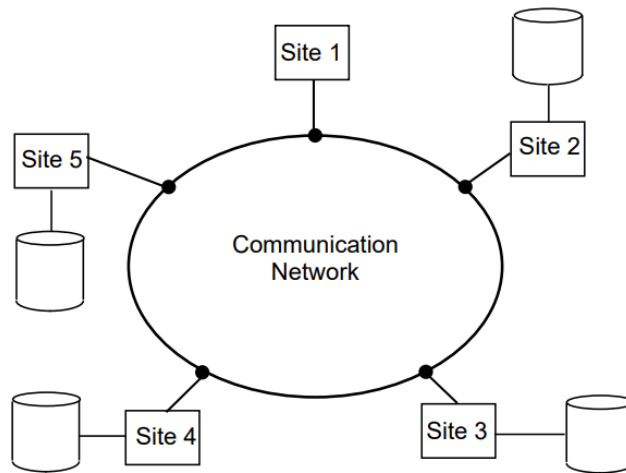
Um SGBDD também não é um sistema onde, apesar da existência de uma rede, o banco de dados reside em apenas um nó da rede (Figura 1). Nesse caso, os problemas de gerenciamento de banco de dados não são diferentes dos problemas encontrados em um ambiente de banco de dados centralizado. O banco de dados é gerenciado centralmente por um sistema de computador e todas as solicitações são roteadas para esse site. A única consideração adicional tem a ver com atrasos na transmissão. É óbvio que a existência de uma rede de computadores ou de uma coleção de “arquivos” não é suficiente para formar um sistema de banco de dados distribuído. O que nos interessa é um ambiente onde os dados são distribuídos entre vários sites (Figura 2).

Fig 1: Banco de dados central em uma rede:



Fonte: Principles of Distributed Database Systems

Fig 2: Sistemas de banco de dados distribuídos:



Fonte: Principles of Distributed Database Systems

Para uma compreensão mais aprofundada desse tema, é importante familiarizar-se com os seguintes conceitos:

- **Localização de Dados:** Esse conceito envolve a distribuição de dados de um mesmo banco de dados em diversos nós. Essa estratégia tem como objetivo melhorar o desempenho e a disponibilidade do banco de dados.
- **Balanceamento de Carga:** O balanceamento de carga é uma técnica que divide o processamento entre dois ou mais servidores de banco de dados. Isso não apenas aumenta a tolerância a falhas, mas também contribui para melhorar o desempenho do sistema.
- **Cluster de Banco de Dados:** Um cluster de banco de dados é um conjunto de nós independentes que estão interconectados para compartilhar recursos e funcionar como um único sistema coeso. Essa configuração é valiosa para escalabilidade e redundância.

## **B) Objetivos principais dos Bancos de Dados Paralelos**

O processamento paralelo explora computadores multiprocessadores para executar programas aplicativos usando vários processadores cooperativamente, a fim de melhorar o desempenho. Seu uso proeminente tem sido há muito tempo na computação científica, melhorando o tempo de

resposta de aplicações numéricas [Kowalik, 1985; Afiado, 1987]. Os desenvolvimentos em computadores paralelos de uso geral usando microprocessadores padrão e técnicas de programação paralela [Osterhaug, 1989] permitiram que o processamento paralelo invadisse o campo de processamento de dados. (M. Tamer Özsu & Patrick Valduriez, “Principles of Distributed Database Systems”).

Os sistemas de banco de dados paralelos combinam gerenciamento de banco de dados e processamento paralelo para aumentar o desempenho e a disponibilidade. Observe que o desempenho também era o objetivo das máquinas de banco de dados nas décadas de 70 e 80 [Hsiao, 1983]. O problema enfrentado pelo gerenciamento convencional de banco de dados é conhecido há muito tempo como “gargalo de E/S” [Boral e DeWitt, 1983], induzido pelo alto tempo de acesso ao disco em relação ao tempo de acesso à memória principal (normalmente centenas de milhares de vezes mais rápido). (M. Tamer Özsu & Patrick Valduriez, “Principles of Distributed Database Systems”).

Os objetivos dos sistemas de banco de dados paralelos são cobertos pelos dos SGBD distribuídos (desempenho, disponibilidade, extensibilidade). Idealmente, um sistema de banco de dados paralelo deveria oferecer as seguintes vantagens:

**1. Alto desempenho:** Isto pode ser obtido através de diversas soluções complementares: suporte a sistemas operacionais orientados a banco de dados, gerenciamento paralelo de dados, otimização de consultas e balanceamento de carga. Ter o sistema operacional restrito e “consciente” dos requisitos específicos do banco de dados (por exemplo, gerenciamento de buffer) simplifica a implementação de funções de banco de dados de baixo nível e, portanto, diminui seu custo. Por exemplo, o custo de uma mensagem pode ser significativamente reduzido para algumas centenas de instruções através da especialização do protocolo de comunicação. O paralelismo pode aumentar o rendimento, usando paralelismo entre consultas, e diminuir os

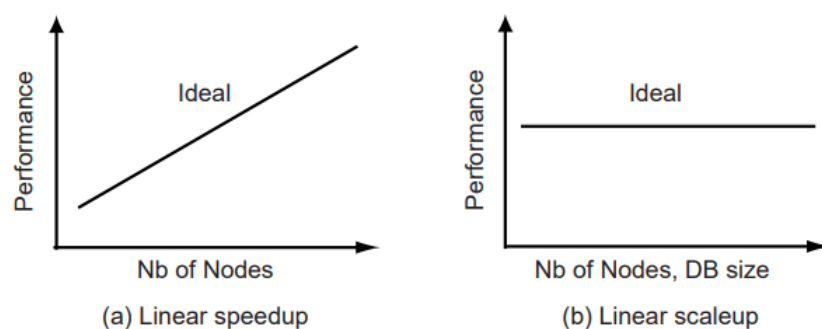
tempos de resposta das transações, usando paralelismo intra-consulta. No entanto, diminuir o tempo de resposta de uma consulta complexa através do paralelismo em larga escala pode muito bem aumentar o seu tempo total (por meio de comunicação adicional) e prejudicar o rendimento como efeito colateral. Portanto, é crucial otimizar e paralelizar as consultas para minimizar a sobrecarga do paralelismo, por exemplo, restringindo o grau de paralelismo da consulta. O balanceamento de carga é a capacidade do sistema de dividir uma determinada carga de trabalho igualmente entre todos os processadores. Dependendo da arquitetura do sistema paralelo, isso pode ser alcançado estaticamente pelo design físico apropriado do banco de dados ou dinamicamente em tempo de execução. (M. Tamer Özsu & Patrick Valduriez, “Principles of Distributed Database Systems”).

**2. Alta disponibilidade:** Como um sistema de banco de dados paralelo consiste em muitos componentes redundantes, ele pode aumentar a disponibilidade de dados e a tolerância a falhas. Em um sistema altamente paralelo com muitos nós, a probabilidade de falha de um nó a qualquer momento pode ser relativamente alta. A replicação de dados em vários nós é útil para suportar failover, uma técnica de tolerância a falhas que permite o redirecionamento automático de transações de um nó com falha para outro nó que armazena uma cópia dos dados. Isso fornece serviço ininterrupto aos usuários. No entanto, é essencial que uma falha no nó não crie desequilíbrio de carga, por exemplo, duplicando a carga na cópia disponível. As soluções para este problema requerem o particionamento de cópias de forma que também possam ser acessadas em paralelo. (M. Tamer Özsu & Patrick Valduriez, “Principles of Distributed Database Systems”).

**3. Extensibilidade:** Em um sistema paralelo, acomodar tamanhos crescentes de bancos de dados ou demandas crescentes de desempenho (por exemplo, taxa de transferência) deveria ser mais fácil. Extensibilidade é a capacidade de expandir o sistema sem problemas, adicionando capacidade de processamento

e armazenamento ao sistema. Idealmente, o sistema de banco de dados paralelo deveria demonstrar duas vantagens de extensibilidade [DeWitt e Gray, 1992]: aceleração linear e aumento de escala linear (Figura 3). A aceleração linear refere-se a um aumento linear no desempenho para um tamanho de banco de dados constante, enquanto o número de nós (ou seja, poder de processamento e armazenamento) aumenta linearmente. O aumento de escala linear refere-se a um desempenho sustentado para um aumento linear no tamanho do banco de dados e no número de nós. Além disso, a extensão do sistema deverá exigir uma reorganização mínima da base de dados existente. (M. Tamer Özsu & Patrick Valduriez, “Principles of Distributed Database Systems”).

Fig3: Métricas de Extensibilidade



Fonte: Principles of Distributed Database Systems

### C) Vantagens dos Bancos de Dados Paralelos

Desempenho Aprimorado:

Bancos de Dados Paralelos: Essa tecnologia aproveita o processamento paralelo para melhorar o desempenho das consultas e transações. O processamento é distribuído entre vários nós, o que permite consultas mais rápidas e respostas mais ágeis, especialmente para consultas complexas que podem ser divididas em tarefas menores executadas em paralelo.

Bancos de Dados Relacionais: Os bancos de dados relacionais tradicionais em sistemas distribuídos podem enfrentar problemas de desempenho devido ao

gargalo de E/S e à necessidade de comunicação entre nós, o que pode resultar em latência. Consultas complexas podem ser mais lentas devido à falta de paralelismo eficaz.

#### Alta Disponibilidade e Tolerância a Falhas:

**Bancos de Dados Paralelos:** Devido à redundância de dados e à capacidade de failover, os sistemas de bancos de dados paralelos oferecem maior disponibilidade de dados e tolerância a falhas. Em caso de falha de um nó, as transações podem ser redirecionadas para outros nós, garantindo um serviço contínuo.

**Bancos de Dados Relacionais:** A alta disponibilidade e a tolerância a falhas podem ser desafiadoras de implementar em sistemas de bancos de dados relacionais distribuídos, uma vez que tradicionalmente dependem de nós centrais para o gerenciamento. A falha em um nó central pode resultar em interrupções.

#### Extensibilidade Eficiente:

**Bancos de Dados Paralelos:** Os bancos de dados paralelos são mais facilmente escaláveis. Eles demonstram aceleração linear e aumento de escala linear, o que significa que, à medida que mais capacidade de processamento e armazenamento é adicionada, o desempenho permanece eficiente. Além disso, a reorganização da base de dados existente é mínima.

**Bancos de Dados Relacionais:** A escalabilidade em sistemas de bancos de dados relacionais distribuídos pode ser mais complexa, e a reorganização da base de dados pode ser necessária à medida que o sistema cresce, o que pode ser um processo trabalhoso.

Conquanto, a escolha entre bancos de dados paralelos e relacionais em um ambiente distribuído deve ser baseada nas necessidades específicas do projeto, na complexidade das consultas e na capacidade de escalabilidade



requerida.

## **D) Desvantagens dos Bancos de Dados Paralelos**

### **Complexidade de Implementação:**

**Bancos de Dados Paralelos:** A implementação de um sistema de banco de dados paralelo é mais complexa em comparação com um banco de dados relacional tradicional. Requer expertise em programação paralela, distribuição de dados, balanceamento de carga e gerenciamento de nós. A complexidade aumenta à medida que o número de nós e a escala do sistema aumentam.

**Bancos de Dados Relacionais:** Os sistemas de bancos de dados relacionais tendem a ter uma implementação mais simples e direta, especialmente em ambientes com um único nó central. A complexidade geralmente aumenta quando há a necessidade de distribuir dados em vários nós.

### **Consistência e Concorrência:**

**Bancos de Dados Paralelos:** Lidar com consistência e concorrência em sistemas de bancos de dados paralelos pode ser desafiador. A distribuição de dados e o processamento paralelo introduzem a necessidade de controlar o acesso concorrente aos dados e garantir a consistência, o que pode resultar em complexidade adicional no design e na implementação.

**Bancos de Dados Relacionais:** Os sistemas de bancos de dados relacionais geralmente têm mecanismos de controle de concorrência e transações bem estabelecidos, o que simplifica a garantia da consistência dos dados. No entanto, a distribuição de dados em ambientes distribuídos pode introduzir complexidade na garantia da consistência.

### **Custo e Recursos de Hardware:**

**Bancos de Dados Paralelos:** Implementar um sistema de banco de dados

paralelo muitas vezes requer hardware especializado, como clusters de servidores e redes de alta velocidade. Isso pode resultar em custos significativos de infraestrutura e manutenção.

Bancos de Dados Relacionais: Em comparação, sistemas de bancos de dados relacionais podem ser implantados em hardware convencional e não exigem necessariamente investimentos em infraestrutura de alto desempenho. Isso pode torná-los mais acessíveis em termos de custo.

## **E) Exemplo de uso de Bancos de Dados Paralelos**

Postgress: O PostgreSQL pode criar planos de consulta que podem aproveitar múltiplas CPUs para responder às consultas com mais rapidez. (Chapter 15. Parallel Query, [PostgreSQL: Documentation: 16: Chapter 15. Parallel Query](#))

MySQL NDB Cluster: MySQL Cluster possui um mecanismo de consulta paralelo exclusivo. Ele fornece uma visão transacional consolidada consistente de todo o conjunto de dados particionados distribuídos. ( MySQL NDB Cluster: Parallel Query Engine, [MySQL NDB Cluster: Parallel Query Engine](#))

### **História de Sucesso: A Implementação do TAO pelo Facebook**

O Facebook, uma das maiores redes sociais do mundo, enfrentou um desafio monumental à medida que sua base de usuários crescia exponencialmente. O desafio estava em manter um sistema robusto e escalável que pudesse lidar com bilhões de usuários e trilhões de interações de dados em tempo real. Para superar esse desafio, o Facebook desenvolveu com sucesso uma tecnologia interna denominada TAO, abreviação de "The Associations and Objects."

O TAO é um sistema de armazenamento de gráfico distribuído que combina características de um banco de dados paralelo e um banco de dados distribuído. Sua função principal é permitir que o Facebook armazene e acesse eficientemente conexões entre usuários e os dados associados, incluindo fotos,

mensagens, curtidas e outras informações relacionadas.

O sucesso do TAO transformou o Facebook, permitindo que a plataforma atendesse a bilhões de usuários em todo o mundo, mantendo tempos de resposta rápidos e fornecendo uma experiência de usuário excepcional. A arquitetura paralela e distribuída do TAO foi fundamental para lidar com a intensa carga de trabalho e as operações simultâneas exigidas por uma rede social global de tal magnitude.

No entanto, é crucial observar que, ao longo de sua jornada, o Facebook também enfrentou desafios críticos relacionados à privacidade e à segurança dos dados. Várias violações de privacidade e preocupações com o uso indevido de dados levaram a escrutínio regulatório e questões éticas em várias ocasiões.

Esta história de sucesso ilustra como a implementação de um banco de dados paralelo e distribuído pode ser essencial para alcançar a escalabilidade e o desempenho em empresas de tecnologia em crescimento. Ao mesmo tempo, destaca a importância de uma governança sólida dos dados e medidas rigorosas de segurança para garantir a integridade e a confiança dos usuários em um ambiente digital em constante evolução.

## **G) Base de Dados**

Essa base de dados foi criada para armazenar informações relacionadas a funcionários, vendas, produtos e clientes de uma empresa. Ela possibilita o acompanhamento das atividades corporativas de forma organizada:

A tabela "EMPLOYEE" contém dados sobre os funcionários, incluindo seus nomes e identificações exclusivas.

A tabela "SALE" registra informações sobre as vendas, tais como identificação única da venda, o cliente associado e a quantidade de produtos vendidos.

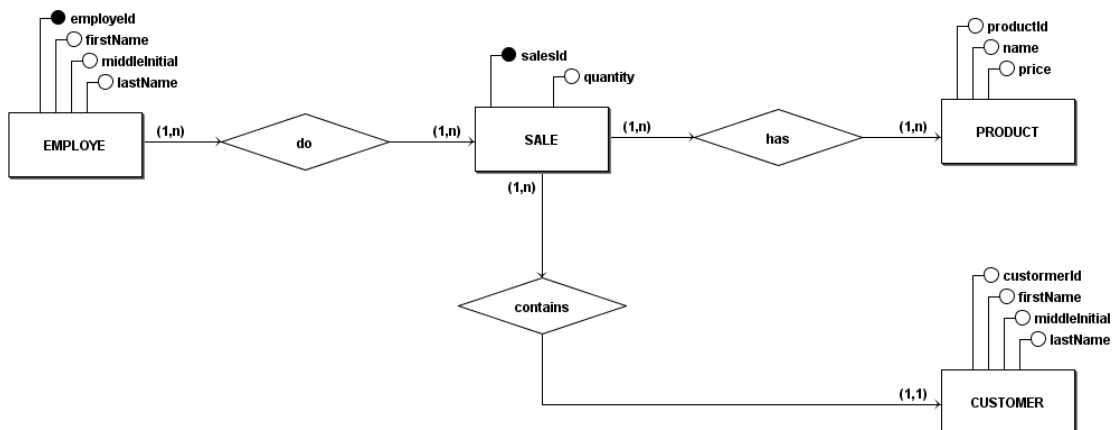
A tabela "PRODUCT" mantém informações sobre os produtos comercializados, como nomes e preços.

A tabela "CUSTOMER" registra informações dos clientes, incluindo seus nomes.

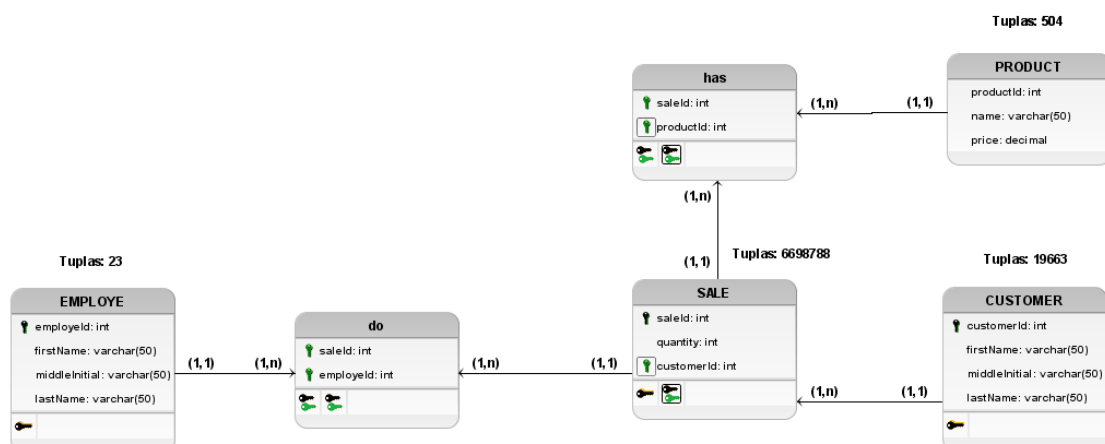
Além disso, há tabelas de relacionamento, como "do" e "has", que conectam essas entidades. A tabela "do" relaciona os funcionários às vendas, enquanto a tabela "has" relaciona as vendas aos produtos vendidos.

Essa base de dados é fundamental para uma gestão eficiente dos recursos humanos, o controle de vendas e o monitoramento de produtos e clientes, proporcionando uma visão detalhada das operações empresariais.

## DER



## DLD



## DICIONÁRIO DE DADOS

<b>Entidade: EMPLOYE</b>				
<b>Descrição:</b> Tabela que armazena informações sobre funcionários da empresa.				
Atributo	Propriedades do atributo	Tipo de dado	Tamanho	Descrição
employeId	Chave primária Obrigatória	INT	-	Identificação única do funcionário
firstName	Obrigatório	VARCHAR	50	Primeiro nome do funcionário
middleInitial		VARCHAR	50	Nome do meio do funcionário
lastName	Obrigatório	VARCHAR	50	Último nome do funcionário

<b>Entidade: do</b>				
<b>Descrição:</b> Tabela que realiza relacionamento entre funcionário e venda				
Atributo	Propriedades do atributo	Tipo de dado	Tamanho	Descrição
saleId	Obrigatório	INT	-	Identificação única da venda
employeId	Obrigatório	INT	-	Identificação única do funcionário

<b>Entidade: SALE</b>				
<b>Descrição:</b> Tabela que armazena informações de compras				
<b>Atributo</b>	<b>Propriedades do atributo</b>	<b>Tipo de dado</b>	<b>Tamanho</b>	<b>Descrição</b>
saleId	Obrigatório	INT	-	Identificação única da venda
customerId	Obrigatório	INT	-	Identificação única de cliente
quantity	Obrigatório	INT	-	Quantidade de produtos

<b>Entidade: has</b>				
<b>Descrição:</b> Tabela que realiza relacionamento entre venda e produto				
<b>Atributo</b>	<b>Propriedades do atributo</b>	<b>Tipo de dado</b>	<b>Tamanho</b>	<b>Descrição</b>
saleId	Obrigatório	INT	-	Identificação única da venda
productId	Obrigatório	INT	-	Identificação única de produto

<b>Entidade: PRODUCT</b>				
<b>Descrição:</b> Tabela que realiza relacionamento entre venda e produto				
<b>Atributo</b>	<b>Propriedades do atributo</b>	<b>Tipo de dado</b>	<b>Tamanho</b>	<b>Descrição</b>
productId	Obrigatório	INT	-	Identificação única da venda
name	Obrigatório	VARCHAR	50	Nome do produto
price	Obrigatório	DECIMAL	-	Preço do produto

Entidade: CUSTOMER				
Descrição: Tabela que realiza relacionamento entre venda e produto				
Atributo	Propriedades do atributo	Tipo de dado	Tamanho	Descrição
customerId	Obrigatório	INT	-	Identificação única do consumidor
firstName	Obrigatório	VARCHAR	50	Primeiro nome do funcionário
middleName	Obrigatório	VARCHAR	50	Nome do meio do funcionário
lastName	Obrigatório	VARCHAR	50	Último nome do funcionário

Acesso da base disponível em: [SalesDB](#)

## F) REFERÊNCIAS

Chang, F., Dean, J., Ghemawat, S., Hsieh, W. C., Wallach, D. A., Burrows, M., Chandra, T., Fikes, A., & Gruber, R. E. (2006). Bigtable: A Distributed Storage System for Structured Data. Google, Inc. [Online]. Disponível em: <http://research.google.com/archive/bigtable.html>. Acesso em 17/09/2023

Facebook Research. (2013). Tao: Facebook's Distributed Data Store for the Social Graph. [Online]. Disponível em: <https://research.facebook.com/publications/tao-facebooks-distributed-data-store-for-the-social-graph/>.

Lira, Márcio Angelo B. de, "Sistemas de Banco de Dados Paralelos", Pós-Graduação em Ciência da Computação, Universidade Federal de Pernambuco (UFPE), 2023. Disponível em: [https://www.cin.ufpe.br/~if694/aulas\\_pdf/12%20-%20Sistemas%20de%20banco%20de%20dados%20Paralelos.pdf](https://www.cin.ufpe.br/~if694/aulas_pdf/12%20-%20Sistemas%20de%20banco%20de%20dados%20Paralelos.pdf). Acesso em 17/09/2023.

MySQL Documentation. (2023). MySQL NDB Cluster: Parallel Query Engine. In MySQL NDB Cluster. [Online]. Disponível em: <https://dev.mysql.com/doc/mysql-cluster/en/mysql-cluster-parallel-query-engine.html>. Acesso em 17/09/2023.

ÖZSU, M. Tamer et al. **Principles of distributed database systems**. Englewood Cliffs: Prentice Hall, 1999.

PostgreSQL Documentation. (2023). Chapter 15. Parallel Query. In PostgreSQL 16. [Online]. Disponível em: [PostgreSQL: Documentation: 16: Chapter 15. Parallel Query](#). Acesso em 17/09/2023.

Facebook Research. (2013). Tao: Facebook's Distributed Data Store for the Social Graph. [Online]. Disponível em: <https://research.facebook.com/publications/tao-facebooks-distributed-data-store-for-the-social-graph/>.