

OTIMIZAÇÃO, MONITORAMENTO E OPERAÇÃO EM BANCOS DE DADOS



PUC Minas
Virtual



PUC Minas
Virtual

UNIDADE IV – SEGURANÇA E RECUPERAÇÃO DE FALHAS



PUC Minas
Virtual

CONTINUIDADE DE OPERAÇÕES



PUC Minas
Virtual

IMPLEMENTAÇÕES DE ALTA DISPONIBILIDADE

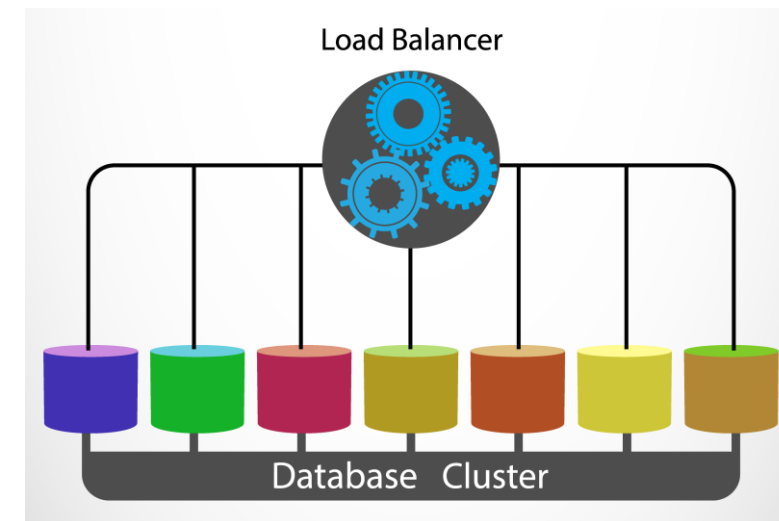
A NECESSIDADE É...

- **Operação contínua** (*Continuous operation*), também conhecida como **disponibilidade contínua** (*continuous availability*) ou **disponibilidade sempre ativa** (*always-on availability*), são termos utilizados para descrever um sistema de banco de dados projetado para operar continuamente sem interrupção.
- Significa que haverá disponibilidade do serviço de banco de dados **mesmo em caso de falha ou interrupção**.



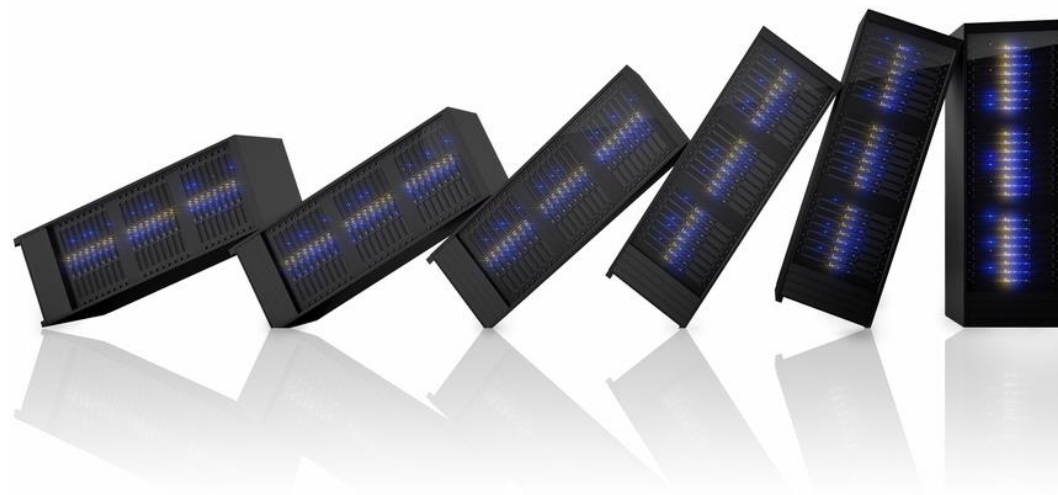
A NECESSIDADE É...

- Para obter **operação contínua**, um sistema de banco de dados deve ser **projetado** para fornecer **alta disponibilidade e tolerância a falhas**.
- Envolve o uso de uma combinação de **hardware, software e tecnologias de rede** para criar uma infraestrutura de banco de dados altamente disponível e **resiliente**.
- A técnica que você vai utilizar vai depender do tipo de banco que você está usando.



TÉCNICAS PARA ALTA DISPONIBILIDADE

- Termo comum que iremos ouvir no mercado é a utilização de ferramentas de **HA**, que significa *High Availability*.
- Estas ferramentas e técnicas evitaram queda de um ou mais nós dentro do mesmo Datacenter ou em localidades distintas.
- Para sistemas de banco de dados SQL, a **operação contínua** pode ser obtida por meio de técnicas, como *clustering*, *replicação* e *clustering failover*.



TÉCNICAS PARA ALTA DISPONIBILIDADE

- **Replicação:** envolve a combinação de vários servidores em uma única unidade lógica, neste processo se cria várias cópias do mesmo banco de dados que são mantidas sincronizadas em tempo real.
- A ideia básica da replicação é criar cópias do bancos de dados que sejam idênticas ao banco de dados primário.
- Há uma padronização nas etapas para implementar os tipos de replicação, observe:



PROCESSO GENÉRICO PARA REPLICAÇÃO

1. **Captura:** o sistema de replicação captura as alterações feitas no banco de dados principal. Envolve o **registro de alterações no log de transações** ou o **monitoramento de *updates*** (tabelas ou colunas) específicas.
2. **Transporte:** as alterações **capturadas** são transportadas **para os bancos de dados de réplica**. O envio das alterações ocorrer por meio de uma conexão de rede, possibilita armazenamento em um dispositivo ou uso de um sistema de mensagens.



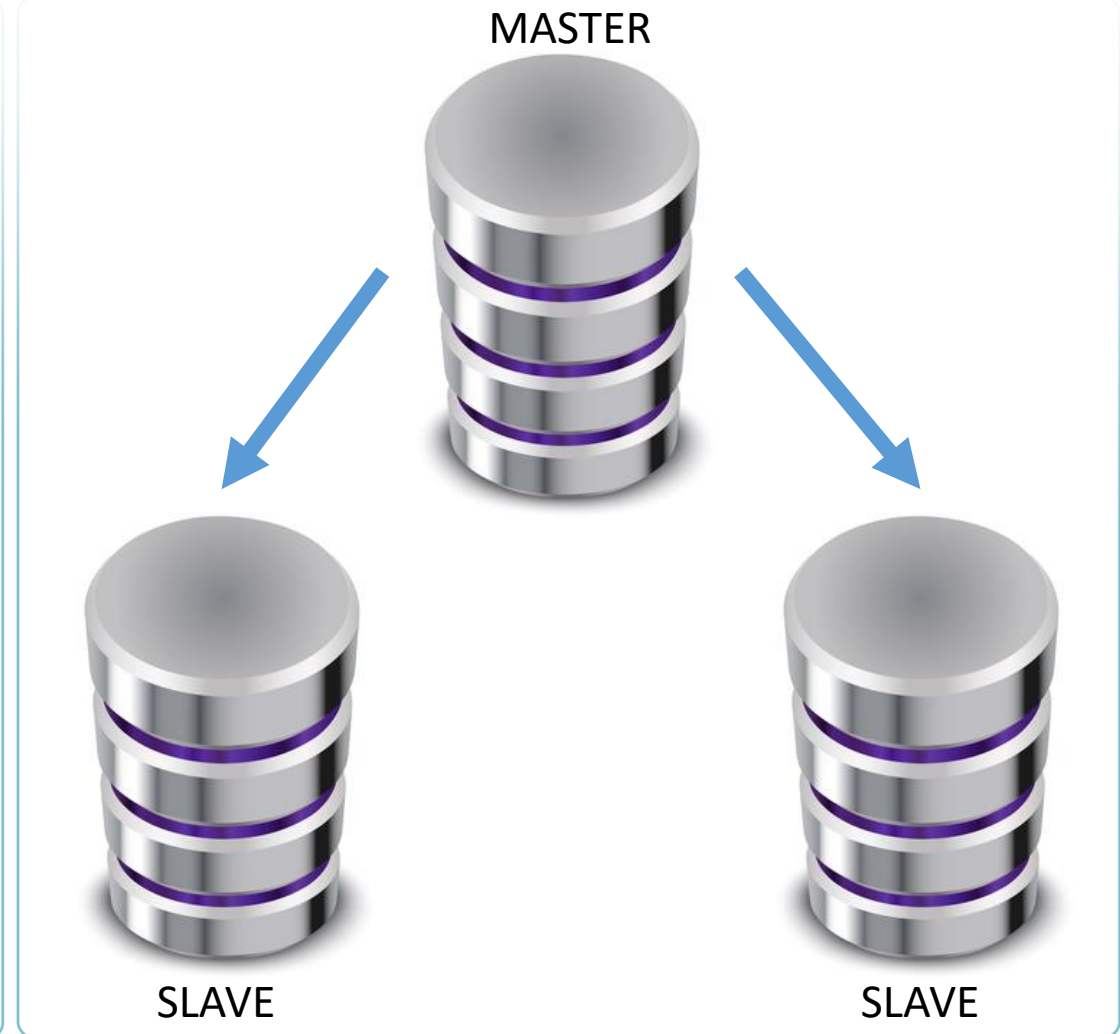
PROCESSO GENÉRICO PARA REPLICAÇÃO

3. **Aplicação:** As alterações são aplicadas aos bancos de dados de **réplica**, garantindo que todas as cópias dos dados sejam **consistentes e atualizadas**.
- Isso pode envolver a execução de **instruções SQL** ou a aplicação de **transformações de dados** às alterações capturadas.



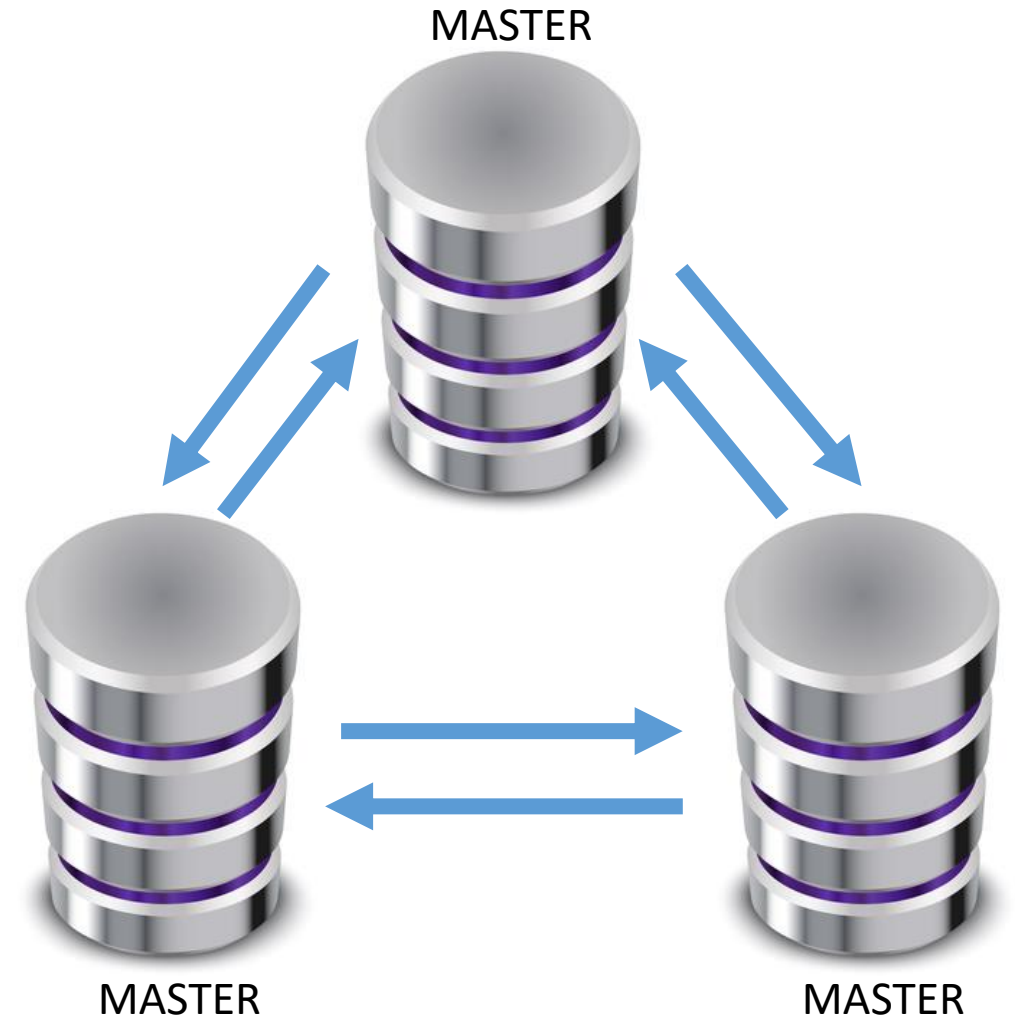
AS TOPOLOGIAS DE REPLICAÇÃO

- **Replicação *Master-slave*:** determina que um dos nós será o banco de dados primário (*master*) e todas as alterações são feitas nesse banco de dados.
- Podem haver vários bancos de dados secundários (*slaves*) que recebem a replica das alterações feitas no *master*.
- **Este procedimento garante que todas as cópias dos dados sejam consistentes.**



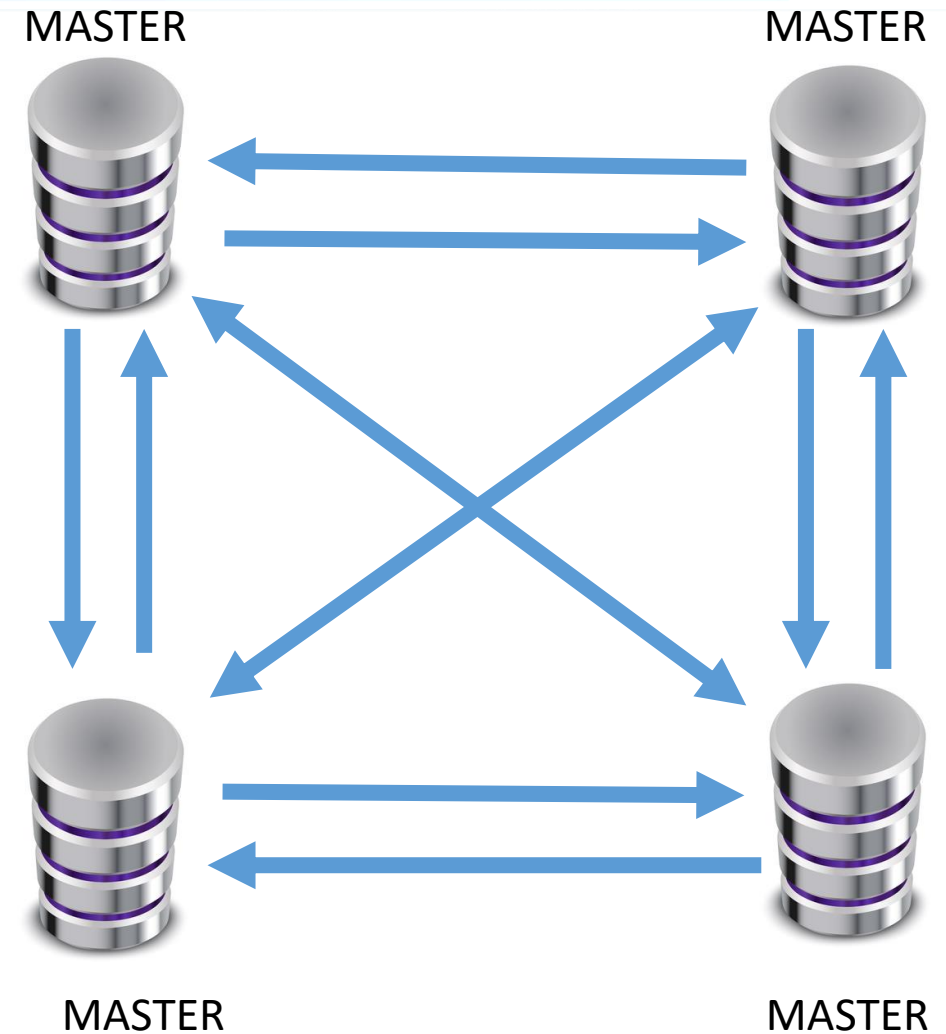
AS TOPOLOGIAS DE REPLICAÇÃO

- **Replicação *Multi-Master*:** vários nós bancos de dados são designados como *master* e as alterações podem ser feitas em qualquer um desses bancos de dados.
- É semelhante à topologia *Master-Slave*, com a diferença que quando **os nós são master** e há **replicação circular** entre todos que podem ser graváveis.



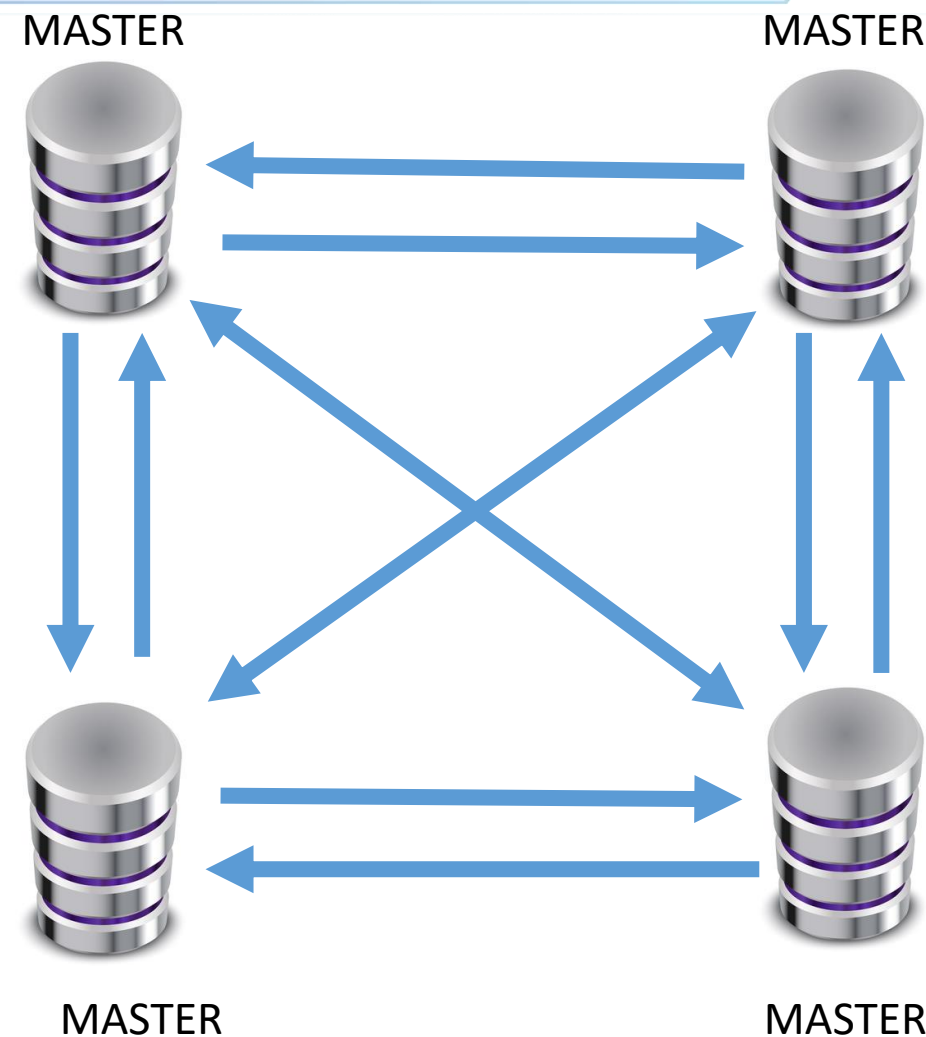
AS TOPOLOGIAS DE REPLICAÇÃO

- **Replication *Peer-to-peer*** : vários nós de bancos de dados são designados como *master* e alterações podem ser feitas em qualquer um desses bancos de dados.
- Mesmo mecanismo da topologia *multi-master* porém com vários nós distribuídos em diversas localidades.
- Existem basicamente três papéis que tornam esta replicação possível:



AS TOPOLOGIAS DE REPLICAÇÃO

- ✓ **Publisher:** É o banco de dados de origem, que contém os dados a serem replicados.
- ✓ **Subscriber:** Este é o banco de dados de destino e pode haver muitos deles para um único *Publisher*.
- ✓ **Distribuidor:** utilizado para distribuir as transações para o banco de dados do *subscriber*.



AS TOPOLOGIAS DE REPLICAÇÃO

- Vivemos um momento de transformação das cargas de trabalho tradicionais.
- Este novo processo enfatiza **arquitetura, automação, engenharia de software, integração e entrega contínuas** e habilidades de **instrumentação de sistemas**, acima de tudo.
- A importância e o valor dos dados que administramos o tempo todo aumentaram **estão aumentando e não irão parar.**



BANCOS NoSQL

- Os bancos NoSQL, são a mesma coisa?
- **PARECIDO**
- As técnicas são diferentes como *sharding* e **implantação de vários datacenters**, porém na maior parte das vezes a **técnica de replicação** é utilizada.
- Nos bancos de dados NoSQL, o fundamento é possuir localização geográfica distribuída e, desta forma há requisitos diferentes na técnica de replicação.



BANCOS NoSQL

- ✓ **Sharding:** envolve o particionamento dos dados em vários servidores para aumentar o desempenho e a escalabilidade.
- ✓ **Replicação:** replicação envolve a criação de várias cópias dos dados e sua manutenção sincronizada em tempo real.



BANCOS NoSQL

- ✓ **Vários Datacenters:** envolve a implantação do banco de dados em vários locais geográficos para garantir a redundância e reduzir a latência.
- Com a distribuição global os clientes estão espalhados e os bancos de dados devem ser mantidos sincronizados em vários sites.
- Isto impulsiona inclusive a utilização das tecnologias em Nuvem e o NoSQL é uma das melhores tecnologias para atender esta demanda.



SHARDING

- **Sharding:** é uma técnica usada em bancos de dados NoSQL para **particionar horizontalmente** os dados em **vários servidores ou nós**.
- Na prática cada nó armazena apenas uma parte dos dados, em vez de todo o conjunto de dados.
- Quando dividimos os dados em partes menores e os distribuímos em vários nós, o banco de dados cria habilidade para lidar com grandes volumes de dados.

Original Table

CUSTOMER ID	FIRST NAME	LAST NAME	FAVORITE COLOR
1	TAEKO	OHNUKI	BLUE
2	O.V.	WRIGHT	GREEN
3	SELDA	BAGCAN	PURPLE
4	JIM	PEPPER	AUBERGINE

Vertical Partitions

VP1

CUSTOMER ID	FIRST NAME	LAST NAME
1	TAEKO	OHNUKI
2	O.V.	WRIGHT
3	SELDA	BAGCAN
4	JIM	PEPPER

VP2

CUSTOMER ID	FAVORITE COLOR
1	BLUE
2	GREEN
3	PURPLE
4	AUBERGINE

Horizontal Partitions

HP1

CUSTOMER ID	FIRST NAME	LAST NAME	FAVORITE COLOR
1	TAEKO	OHNUKI	BLUE
2	O.V.	WRIGHT	GREEN

HP2

CUSTOMER ID	FIRST NAME	LAST NAME	FAVORITE COLOR
3	SELDA	BAGCAN	PURPLE
4	JIM	PEPPER	AUBERGINE

Fonte: DRAKE, 2022

SHARDING

- O processo de fragmentação normalmente envolve a divisão dos dados em *shards* ou **partições**.
- Com base em uma chave de fragmentação ou função de *hash* os fragmentos são então distribuídos em vários nós.
- Cada nó é responsável por armazenar e servir um subconjunto dos dados.
- Um dos nós atua como nó de roteamento, nele fica a localização de cada registro.

Original Table

CUSTOMER ID	FIRST NAME	LAST NAME	FAVORITE COLOR
1	TAEKO	OHNUKI	BLUE
2	O.V.	WRIGHT	GREEN
3	SELDA	BAGCAN	PURPLE
4	JIM	PEPPER	AUBERGINE

Vertical Partitions

VP1

CUSTOMER ID	FIRST NAME	LAST NAME
1	TAEKO	OHNUKI
2	O.V.	WRIGHT
3	SELDA	BAGCAN
4	JIM	PEPPER

VP2

CUSTOMER ID	FAVORITE COLOR
1	BLUE
2	GREEN
3	PURPLE
4	AUBERGINE

Horizontal Partitions

HP1

CUSTOMER ID	FIRST NAME	LAST NAME	FAVORITE COLOR
1	TAEKO	OHNUKI	BLUE
2	O.V.	WRIGHT	GREEN

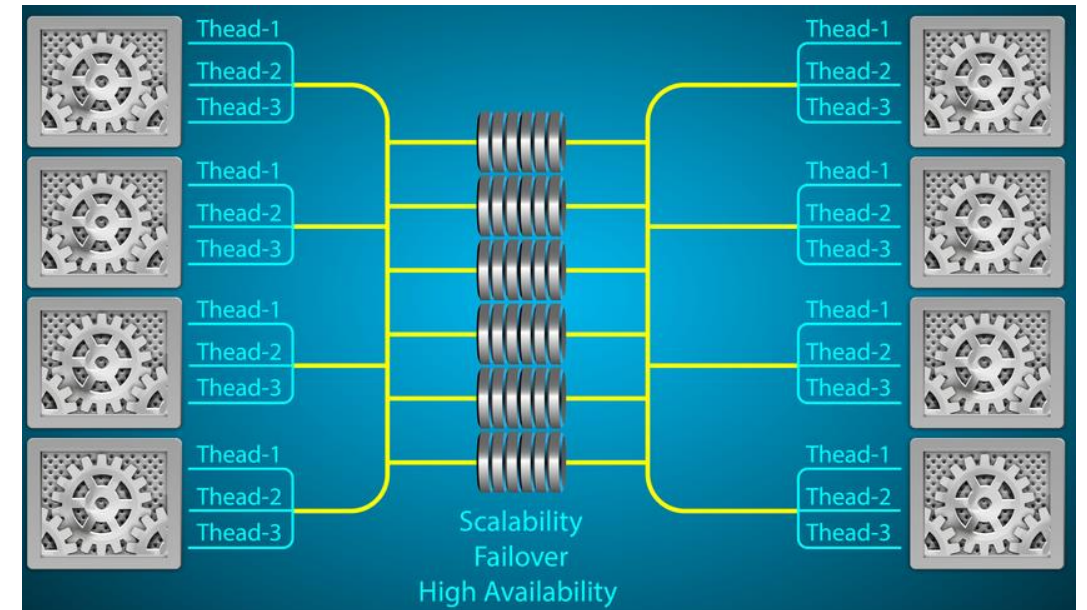
HP2

CUSTOMER ID	FIRST NAME	LAST NAME	FAVORITE COLOR
3	SELDA	BAGCAN	PURPLE
4	JIM	PEPPER	AUBERGINE

Fonte: DRAKE, 2022

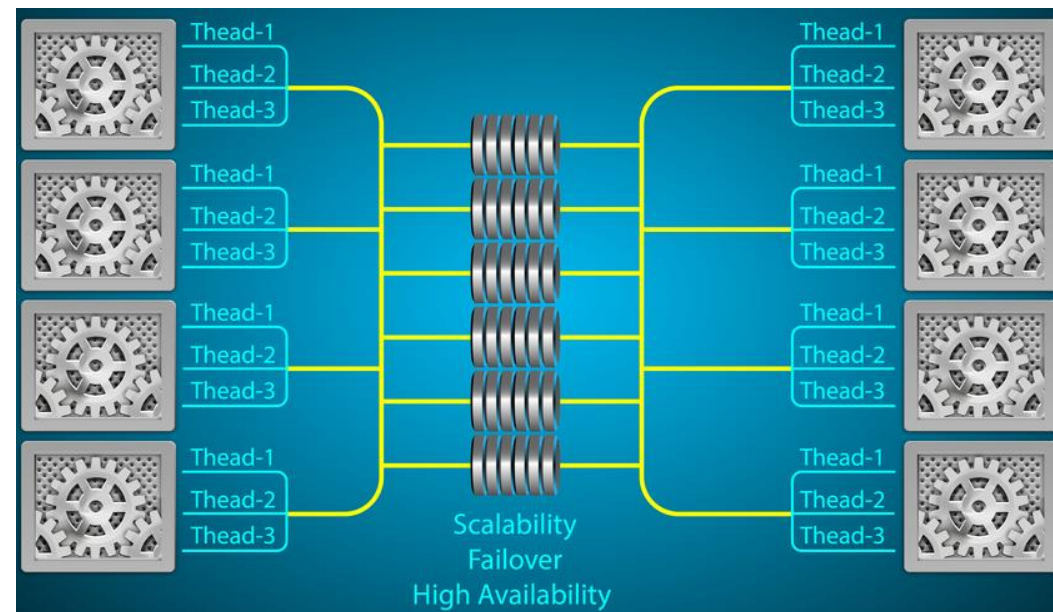
SHARDING

- O processo de fragmentação normalmente envolve a divisão dos dados em *shards* ou **partições**.
- Com base em uma chave de fragmentação ou função de *hash* os fragmentos são então distribuídos em vários nós.
- Cada nó é responsável por armazenar e servir um subconjunto dos dados.
- Um dos nós atua como nó de roteamento, nele fica a localização de cada registro.



O FOCO É...

- Existem diversas técnicas, diversos tipos de SGBD e diversas naturezas de dados em diversos modelos de negócios distintos.
- A escolha, como sempre é com reforçar, se dá ao avaliarmos as necessidades do negócio por dois motivos básicos: 1 – viabilidade financeira e 2 – manutenção e contibuidade das operações de negócio disponíveis.
- **PREPARE-SE PARA FAZER A MELHOR ESCOLHA!**





PUC Minas
Virtual

TECNICAS PARA OPERAÇÃO, GERALMENTE TUDO DÁ CERTO, E QUANDO ALGO DÁ ERRADO?

■ REFERENCIAS BIBLIOGRÁFICAS

GRUPTA, Rajenda, 2016. Peer to peer replication. Disponível em: <<https://www.sqlshack.com/peer-to-peer-replication/>>. Acesso em: 30 Abr. 2023.

SHARIF, Ashraf, 2016. The Difference Between MySQL Multi-Master and Multi-Source Replication. Disponível em: <<https://severalnines.com/blog/difference-between-mysql-multi-master-and-multi-source-replication/>>. Acesso em: 30 Abr. 2023.

LOBZAKOV, Vadym, 2020. PostgreSQL Auto-Clustering with Asynchronous Master-Slave Replication. Disponível em: <<https://www.virtuozzo.com/company/blog/postgresql-auto-clustering-master-slave-replication/>>. Acesso em: 30 Abr. 2023.

DRAKE, Mark, 2022. Understanding Database Sharding. Disponível em: <<https://www.digitalocean.com/community/tutorials/understanding-database-sharding>>. Acesso em: 30 Abr. 2023.

COOPER B. F., **SILBERSTEIN**, A., E. Tam, **RAMAKRISHNAN** R. , **SEARS** R. . Benchmarking cloud serving systems with YCSB. ACM Symposium on Cloud Computing 2010 (ACM SOCC 2010), Indianapolis, IN, USA.



PUC Minas
Virtual