

UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO UNIDADE ACADÊMICA DE
EDUCAÇÃO A DISTÂNCIA E TECNOLOGIA BACHARELADO EM SISTEMAS DE
INFORMAÇÃO

BACHARELADO EM SISTEMAS DE INFORMAÇÃO

ALEXSANDRO MATIAS DE ALMEIDA

**ANÁLISE DE DESEMPENHO APÓS ALTERAÇÕES NAS
VARIÁVEIS NO BANCO DE DADOS MYSQL UTILIZANDO O
SISTEMA LINUX DEBIAN**

PALMARES, 2021

ALEXSANDRO MATIAS DE ALMEIDA

**ANÁLISE DE DESEMPENHO APÓS ALTERAÇÕES NAS VARIÁVEIS NO BANCO
DE DADOS MYSQL UTILIZANDO O SISTEMA LINUX DEBIAN**

Trabalho de Conclusão de Curso apresentado à
Coordenação do Curso de Bacharelado em
Sistemas de informação, pelo aluno
ALEXSANDRO MATIAS DE ALMEIDA,
sob orientação da professora Dra. Juliana
Regueira Basto Diniz, para conclusão do Curso
de Sistemas de Informação.

PALMARES, 2021

AGRADECIMENTOS

Agradeço, primeiramente, a Deus pela oportunidade dessa formação e por ter me dado saúde e força para superar as dificuldades.

A minha mãe, irmãs e esposa, por todo amor, paciência, incentivo e apoio incondicional que me deram durante a minha jornada no curso.

E a todos que direta ou indiretamente fizeram parte da minha formação, o meu muito obrigado.

RESUMO

Em momentos atuais, os Bancos de Dados são ferramentas importantíssimas que auxiliam as empresas na tomada de decisão. Desta forma, os dados são os elementos básicos que subsidiam tal ferramenta. Por isso, se faz necessário o estudo minucioso na forma como estes dados são gerados, armazenados e manipulados para que se obtenha um menor tempo de resposta a eles, acarretando numa maior produtividade das instituições. Baseado neste contexto, o presente estudo mensura o desempenho quando são alteradas as variáveis de configuração do sistema de Banco de Dados MySQL utilizando o Sistema Operacional Linux Debian Buster. Para realização deste teste será utilizado um padrão internacional para avaliação de benchmark chamado TPC-H que serve para medição de carga de trabalho auxiliando no suporte à decisão. Para isso, serão criadas quatro bases de dados: As duas primeiras serão criadas seguindo o modelo de criação e população das tabelas indicado pelo TPC-H, tendo estas os tamanhos de 1 GB e outra de 10 GB sem alteração das variáveis. A partir deste momento serão realizadas as 22 consultas propostas pelo teste de Benchmark e realizadas as respectivas medições do tempo de médio de cada consulta. Na etapa seguinte, foram realizadas as alterações nos valores das variáveis de configuração do MySQL como memória cache e memória *buffer* com o intuito de tornar as consultas ao banco mais rápidas. A partir dos dados coletados, foram realizadas as análises com relação ao tempo médio nas consultas nessas bases de dados supostamente otimizadas.

Palavras-Chave: MySQL, Desempenho, TPC-H.

ABSTRACT

Today, databases are extremely important tools that assist companies in decision making. In this way, data are the basic elements that support such a tool. For this reason, it is necessary to conduct a detailed study of how these data are stored. Based on this context, this study measures performance when the configuration variables of the MySQL Database system are changed using the Debian Buster Linux Operating System. To perform this test, an international standard for benchmark evaluation called TPC-H will be used, which serves to measure workload, helping to support the decision. For this, four databases will be created: The first two will be created following the model of creation and population of the tables indicated by TPC-H, these having the sizes of 1 GB and another of 10 GB without changing the variables. From this moment on, the 22 queries proposed by the Benchmark test will be carried out and the respective measurements of the average time of each consultation will be carried out. In the next step, changes were made to the values of the MySQL configuration variables such as cache memory and buffer memory in order to make queries to the bank faster. Based on the data collected, analyzes were performed regarding the average time spent in queries in these supposedly optimized databases.

Keywords: MySQL, Performance, TPC-H.

LISTA DE FIGURAS

Figura 1 - Modelo simplificado Memória Buffer	16
Figura 2 - Modelo detalhado da Memória Buffer.....	17
Figura 3 – Esquema Banco De Dados Benchmark TPC-H.....	20

LISTA DE TABELAS

Tabela 1 - Características do innodb_buffer_pool_size	14
Tabela 2 - Características do innodb_buffer_pool_instances.....	15
Tabela 3 - Características do innodb_buffer_pool_chunk_size	16
Tabela 4 - Características do key_buffer_size.....	18
Tabela 5 - Valores Padrão da Variáveis do MySQL	18
Tabela 6 - Tamanho Estimado Banco de Dados Fator Escala 10 GB (em tuplas).....	21
Tabela 7 - Resultado de Saída da Consulta 06 do Modelo TPC-H	21
Tabela 8 – Tempos de consultas da Query 01 da Base de Dados de 10 GB	24
Tabela 9 - Tempo Médio das Consultas da Base de Dados não Alterada de tamanho 10 GB .	25
Tabela 10 - Tempos de consulta da Query 01 da Base de Dados de 1 GB	25
Tabela 11 - Tempo Médio das Consultas da Base de Dados não Alterada de tamanho 1 GB .	26
Tabela 12 - Valores das Variáveis padrão Após Alteração	27
Tabela 13 – Relação entre as médias de todas as bases de dados	27

LISTA DE SIGLAS

BD – Banco de Dados.

DDL - Data Definition Language.

DML - Data Manipulation Language.

DCL - Data Control Language

SGBD – Sistema de Gerenciamento de Banco de Dados.

SGBDR – Sistema de Gerenciamento de Banco de Dados Relacional.

SQL – Structured Query Language.

TPC – Transaction Performance Council.

SUMÁRIO

1. INTRODUÇÃO.....	7
1.1 OBJETIVOS	7
1.1.1 OBJETIVO GERAL	8
1.1.2 OBJETIVOS ESPECÍFICOS	8
1.2 JUSTIFICATIVA	8
2. FUNDAMENTAÇÃO TEÓRICA	10
2.1 SISTEMA DE GERENCIAMENTO DE BANCO DE DADOS	10
2.2 BANCO DE DADOS MySQL	10
2.2.1 DDL DATA DEFINITION LANGUAGE	11
2.2.2 DML DATA MANIPULATION LANGUAGE	12
2.2.3 DCL DATA CONTROL LANGUAGE	12
2.3 CHAVES	12
2.3.1 CHAVE PRIMÁRIA	12
2.3.2 CHAVE ESTRANGEIRA	12
2.3.3 INTEGRIDADE REFERENCIAL	13
2.4 VARIÁVEIS DE SISTEMA DO SERVIDOR MYSQL.....	13
2.4.1 INNODB_BUFFER_POOL_SIZE.....	14
2.4.2 INNODB_BUFFER_POOL_INSTANCES	15
2.4.3 INNODB_BUFFER_POOL_CHUNK_SIZE.....	16
2.4.4 KEY_BUFFER_SIZE.....	17
2.4.5 VALORES PADRAO DA VARIÁVEIS	18
2.5 TPC-H.....	18
2.5.1 ENTIDADES E RELACIONAMENTOS DO BANCO DE DADOS	19
2.5.2 GERAÇÃO DOS DADOS E POPULAÇÃO DAS TABELAS	20
3. PROCEDIMENTOS METODOLÓGICOS	22
3.1 METODOLOGIA.....	22

3.2	AMBIENTE DE TESTES	22
3.3	COLETA DE DADOS	23
4.	ANÁLISE DOS RESULTADOS DA PESQUISA	24
5.	CONSIDERAÇÕES FINAIS	29
6.	REFERÊNCIAS	30

1. INTRODUÇÃO

A utilização dos bancos de dados é indispensável em Sistemas de Informação para aplicações no campo empresarial. Com o uso dessa tecnologia é possível auxiliar as empresas no apoio à tomada de decisão o que subsidia o planejamento estratégico destas corporações a agirem de forma célere no que tange as mais diversas áreas administrativas e financeiras da empresa. Então, o investimento nessa ferramenta implica na valorização, por parte dessas empresas, nos seus ativos mais valiosos representados principalmente no tempo, e posteriormente, na segurança da informação.

Desta maneira, para melhor escolha do Banco de Dados dentre os diversos oferecidos pelo mercado, se faz necessário um estudo de viabilidade técnica, juntamente com aplicação de testes de performance alinhados com a necessidade da corporação.

Observando-se a aplicação desta tecnologia ao redor do mundo, verifica-se que os cinco Bancos de Dados mais utilizados nos últimos cinco anos, quatro deles são relacionais. Estes são representados em ordem crescente em Oracle, MySQL, Microsoft SQL Server, PostgreSQL e MongoDB (DB-ENGINES, 2020). Uma vez que o banco MySQL está contido nesta lista, se torna interessante o conhecimento mais específico quanto à performance deste Banco de Dados aplicado no que diz respeito ao tempo gasto nas transações realizadas com os dados quando aplicada determinada carga no sistema.

Então, para realização desse teste de performance, também chamado de *Benchmark*, a instituição internacional *Transaction Processing Performance Council* (TPC) apresenta e valida todo o conjunto de metodologias, métricas e relatórios para realização de testes em Banco de Dados. Dentre toda gama de testes possíveis propostos pelo TPC, um deles, faz referência à tomada de decisão representado pela sigla TPC-H que será realizado neste trabalho.

1.1 OBJETIVOS

Serão apresentados os objetivos que nortearam este trabalho final de curso.

1.1.1 OBJETIVO GERAL

Mensurar a variação de desempenho após alterações dos valores das variáveis de configuração no banco de dados MySQL utilizando como referência de Benchmark o padrão TPC-H.

1.1.2 OBJETIVOS ESPECÍFICOS

- Realizar medição de consultas aos registros sem otimizações;
- Analisar os resultados obtidos;
- Realizar as alterações nas variáveis de sistema no mesmo banco de dados;
- Analisar o tempo gasto nessas consultas depois das otimizações.

1.2 JUSTIFICATIVA

O aumento crescente da utilização dos bancos de dados é atribuído aos avanços nas tecnologias de sistemas de informação que consideram os dados de qualquer instituição como um bem intangível e extremamente valioso. Assim, com o crescimento exponencial, a disponibilidade, integridade, confidencialidade e forma de armazenamento desses dados são imprescindíveis neste processo de competitividade empresarial. Outra aplicação direta do uso de banco de dados é a manipulação de grandes volumes de dados visando também promover subsídios para a tomada de decisão.

Visando a otimização ao acesso dessa informação, os profissionais que trabalham com Banco de Dados procuram realizar configurações rápidas que já impactam positivamente no desempenho desta tecnologia, dentre elas, a alteração das variáveis do sistema como por exemplo, memória de acesso ao disco e memória compartilhada entre o sistema de Gerenciamento de Banco de Dados e o sistema operacional.

Por conseguinte, na seção 2 deste trabalho serão apresentados os conceitos básicos, e aplicação desses bancos de dados, assim como serão explanados os comandos que viabilizam os testes. Na seção 3 será apresentada a metodologia e preparação do ambiente para o teste de

carga, indicando quais os softwares (com suas respectivas versões) e hardware foram utilizados, além de tratar de algumas métricas preestabelecidas juntamente com as configurações específicas dos bancos de dados estudados. Já na seção 4 serão mostrados os resultados obtidos no teste de carga, para que se possa concluir se houve ganho do tempo de consultas ao Banco de Dados após as alterações.

2. FUNDAMENTAÇÃO TEÓRICA

Nesta seção são abordados conceitos básicos necessários para a compreensão dos principais temas abordados nesta pesquisa. De uma forma geral serão explanados os conceitos fundamentais em torno de Sistemas de Gerenciamento de Banco de Dados, os comandos SQL e suas particularidades.

2.1 SISTEMA DE GERENCIAMENTO DE BANCO DE DADOS

Originalmente da sigla em inglês DBMS (Data Base Management System), ou Sistema de Gerenciamento de Banco de Dados (SGBD), é um pacote de softwares projetado cuja função é gerenciar uma base de dados. Algumas das vantagens do seu uso, de acordo com o que é proposto por (RAMAKRISHNAN e GEHRKE, 2009):

- Independência de Dados: Os programas aplicativos não devem, idealmente, ser expostos aos detalhes de representação e armazenamento de dados.
- Acesso Eficiente aos Dados: Um SGBD utiliza uma variedade de técnicas sofisticadas para armazenar e recuperar dados eficientemente.
- Integridade e Segurança dos Dados: Se os dados são sempre acessados através do SGBD, ele pode forçar restrições de integridade.
- Acesso Concorrente e Recuperação de Falha: Um SGBD planeja o acesso concorrente aos dados de maneira tal que os usuários podem achar que os dados estão sendo acessados por apenas um único usuário de cada vez. Além disso, o SGBD protege os usuários dos efeitos de falhas de sistema.

2.2 BANCO DE DADOS MYSQL

O MySQL é um sistema de gerenciamento de banco de dados (SGBD), que utiliza a linguagem SQL (*Structured Query Language* ou Linguagem de Consulta Estruturada). É atualmente um dos sistemas de gerenciamento de bancos de dados mais populares da Oracle

Corporation, com mais de 10 milhões de instalações pelo mundo. Segundo (CABRAL e MURPHY, 2009), o MYSQL possui algumas características relevantes como por exemplo:

- É um Software Livre com base na GPL;
- Alta portabilidade já que suporta praticamente qualquer plataforma atual;
- Excelente desempenho e estabilidade;
- Pouco exigente quanto a recursos de novos hardwares;
- Facilidade no manuseio;
- Contempla a utilização de vários Storage Engines como MyISAM, InnoDB, Falcon, BDB, Archive, Federated, CSV, Solid;
- Suporta controle transacional;
- Suporta Triggers;
- Suporta Stored Procedures e Functions;
- Replicação facilmente configurável;

No entanto, cada SGBD tem suas particularidades dentro da própria linguagem, também tendo implementações diferentes. Neste contexto, a linguagem SQL tem algumas divisões, que facilitam o entendimento da mesma, categorizando seus comandos.

2.2.1 DDL DATA DEFINITION LANGUAGE

Linguagem de Definição de Dados, é a parte da Linguagem SQL que trata, como o próprio nome diz, da definição da estrutura dos dados, cujos efeitos se dão sobre objetos. Esses comandos são utilizados para a criação de bancos de dados, tabelas, views, triggers (ATANAZIO, 2019). Exemplos de comandos: CREATE (criação), ALTER (alteração), DROP (remoção).

2.2.2 DML DATA MANIPULATION LANGUAGE

Linguagem de Manipulação de Dados, é a parte da Linguagem SQL que não altera a estrutura, mas sim os registros de uma base de dados, cujos efeitos se darão sobre registros (ATANAZIO, 2019). São comandos que fazem consultas, inserem, alteram ou apagam registros. Exemplos de comandos: SELECT (consulta), INSERT (inserção), UPDATE (alteração), DELETE (remoção).

2.2.3 DCL DATA CONTROL LANGUAGE

Linguagem de Controle de Dados, é a parte da linguagem SQL referente ao controle de acesso a objetos por usuários e seus respectivos privilégios (ATANAZIO, 2019). Os principais comandos são:

- GRANT: Garante (permite) acesso dado a um usuário;
- REVOKE: Revoga (retira) direitos dados a um usuário. Os direitos dados a um usuário podem ser: ALL, CREATE, EXECUTE, REFERENCES, SELECT, TRIGGER, USAGE, CONNECT, DELETE, INSERT, RULE, TEMPORARY, UPDATE, etc.

2.3 CHAVES

2.3.1 CHAVE PRIMÁRIA

Uma chave primária é uma ou mais colunas que identificam exclusivamente uma linha. Nenhuma das colunas que fazem parte da chave primária pode ser anulável, ou seja, com ausência de algum valor (TPC, 2018).

2.3.2 CHAVE ESTRANGEIRA

Uma chave estrangeira é uma coluna ou combinação de colunas usada para estabelecer a relação entre os dados em duas tabelas. Isso é possível adicionando a coluna ou colunas que

contêm os valores de chave primária de uma tabela à outra tabela. Esta coluna se torna uma chave estrangeira na segunda tabela (TPC, 2018).

2.3.3 INTEGRIDADE REFERENCIAL

Integridade referencial é uma propriedade de dados pela qual uma chave estrangeira em uma tabela tem uma chave primária correspondente em uma tabela diferente (TPC, 2018).

2.4 VARIÁVEIS DE SISTEMA DO SERVIDOR MYSQL

O servidor MySQL mantém muitas variáveis de sistema que configuram sua operação. Cada variável do sistema possui um valor padrão. As variáveis do sistema podem ser definidas na inicialização do servidor usando opções na linha de comando ou em um arquivo de configuração (MYSQL, 2020). A maioria delas pode ser alterada dinamicamente em tempo de execução usando a instrução SET que permite modificar a operação do servidor sem precisar interrompê-lo e reiniciá-lo. Também é possível realizar a alteração dos valores dessas variáveis através ou de um arquivo de configuração no Debian localizado no diretório `/etc/mysql/mysql.cnf`. Porém neste último caso, é necessário reiniciar o servidor (serviço em execução do MySQL).

O MySQL necessita da maior quantidade de memória para cache disponível, para que se tenham operações mais rápidas uma vez que é menor a quantidade de acessos ao disco, que é muito mais lento do que o acesso aos dados na memória. Além disso, a memória alocada para o sistema operacional também deve ser considerada, mesmo que ele possa armazenar em cache alguns dados do MySQL. A seguir estão as memórias mais importantes a serem consideradas para a maioria das instalações (SCHWARTZ, TKACHENKO e ZAITSEV, 2012):

- O pool de buffer InnoDB;
- Caches de chave;
- O cache de consulta;

Levando em consideração as memórias mencionadas anteriormente, os parâmetros que fazem com que seus valores sejam alterados serão explicados a seguir:

2.4.1 INNODB_BUFFER_POOL_SIZE

Este é o parâmetro de configuração do MySQL que especifica a quantidade de memória alocada para o MySQL que determina o tamanho do buffer que o mecanismo de armazenamento InnoDB usa para armazenar dados e índices em cache que implica um acesso aos dados nas tabelas. Esta é uma das configurações mais importantes na configuração de MySQL e deve ser definida com base na RAM do sistema disponível (SCALEGRID, 2018). Este espaço contém muitas estruturas de dados em memória como por exemplo buffers, caches, índices e até mesmo dados de linha. Isso ocorre porque o mecanismo de armazenamento InnoDB armazena seus dados usados com frequência na memória já que armazena os dados dos registros, o índice de hash adaptável, o buffer de inserção, bloqueios e outras estruturas internas.

Em um servidor de banco de dados dedicado usando principalmente tabelas InnoDB, esta deve ser uma porcentagem significativa da memória total disponível para o processo do MySQL (mysqld). Recomenda-se ter cuidado com essa configuração pois se ela for adotada para usar muita memória, poderá causar a troca pelo sistema operacional, degradando o desempenho do mysqld. Na pior das hipóteses, em determinados contextos, usar muita memória fará com que o mysqld trave. (CABRAL e MURPHY, 2009).

De acordo com a documentação da linguagem oficial da tecnologia (MYSQL, 2020), as características dessa variável estão representadas na tabela 1:

Tabela 1 - Características do *innodb_buffer_pool_size*

COMMAND-LINE FORMAT	--INNODB-BUFFER-POOL-SIZE=#
SYSTEM VARIABLE	INNODB_BUFFER_POOL_SIZE
SCOPE	GLOBAL
DYNAMIC	YES
SET_VAR HINT APPLIES	NO
TYPE	INTEGER
DEFAULT VALUE	134217728
MINIMUM VALUE	5242880
MAXIMUM VALUE (64-BIT PLATFORMS)	2**64-1

MAXIMUM VALUE (32-BIT PLATFORMS)	$2^{32}-1$
----------------------------------	------------

Fonte: (MYSQL, 2020)

Caso de um Banco de Dados que foi projeto para conter dados que não crescerão rapidamente, não é necessário alocar memória no buffer pool. Não há problemas em planejar no momento da concepção do projeto um banco de dados em rápido crescimento, é claro, mas às vezes observa-se enormes pools de buffer com uma pequena quantidade de dados (SCHWARTZ, TKACHENKO e ZAITSEV, 2012).

2.4.2 INNODB_BUFFER_POOL_INSTANCES

Essa variável representa o número de regiões, ou segmentos, nas quais o buffer pool do InnoDB é dividido. Este parâmetro é provavelmente uma das maneiras mais importantes de melhorar a escalabilidade do MySQL em máquinas com vários núcleos com uma carga de trabalho altamente simultânea. Para sistemas com mais memória, dividindo o *pool de buffer* em instâncias separadas pode melhorar a simultaneidade, reduzindo a contenção à medida que diferentes *threads* leem e gravam nas páginas em cache. Cada página que é armazenada em ou lido do *buffer pool* é atribuído a uma das instâncias do *buffer pool* aleatoriamente, usando uma função de *hash* (MYSQL, 2020).

De acordo com a documentação oficial (MYSQL, 2020) as características desta variável estão representadas na tabela 2:

Tabela 2 - Características do *innodb_buffer_pool_instances*

COMMAND-LINE FORMAT	--INNODB-BUFFER-POOL-INSTANCES=#
SYSTEM VARIABLE INNODB_BUFFER_POOL_INSTANCES	INNODB_BUFFER_POOL_INSTANCES
SCOPE	GLOBAL
DYNAMIC	NO
SET_VAR	HINT APPLIES NO
TYPE	INTEGER
DEFAULT VALUE	(OTHER) 8 (OR 1 IF INNODB_BUFFER_POOL_SIZE < 1GB)
DEFAULT VALUE (WINDOWS, 32 - BIT PLATFORMS)	(AUTOSIZED)
MINIMUM VALUE	1
MAXIMUM VALUE	64

Fonte: (MYSQL, 2020)

2.4.3 INNODB_BUFFER_POOL_CHUNK_SIZE

Esta variável define o tamanho do bloco para redimensionamento do *buffer pool* do InnoDB operações. Ela serve para evitar a cópia de todas as páginas do *buffer pool* durante as operações de redimensionamento, deixando as transações sendo executadas por partes. De acordo com a documentação do MySQL (MYSQL, 2020), as características dessa variável estão representadas na tabela 3:

Tabela 3 - Características do *innodb_buffer_pool_chunk_size*

COMMAND-LINE FORMAT	--INNODB-BUFFER-POOL-CHUNK-SIZE=#
SYSTEM VARIABLE	INNODB_BUFFER_POOL_CHUNK_SIZE
SCOPE	GLOBAL
DYNAMIC	NO
SET_VAR	NO
TYPE	INTEGER
DEFAULT VALUE	134217728
MINIMUM VALUE	1048576
MAXIMUM VALUE	INNODB_BUFFER_POOL_SIZE / INNODB_BUFFER_POOL_INSTANCES

Fonte: (MYSQL, 2020)

A representação da interação entre as o *innodb_buffer_pool_size*, *innodb_buffer_pool_instances* e *innodb_buffer_pool_chunk_size* está representada de forma simplificada na figura 1:

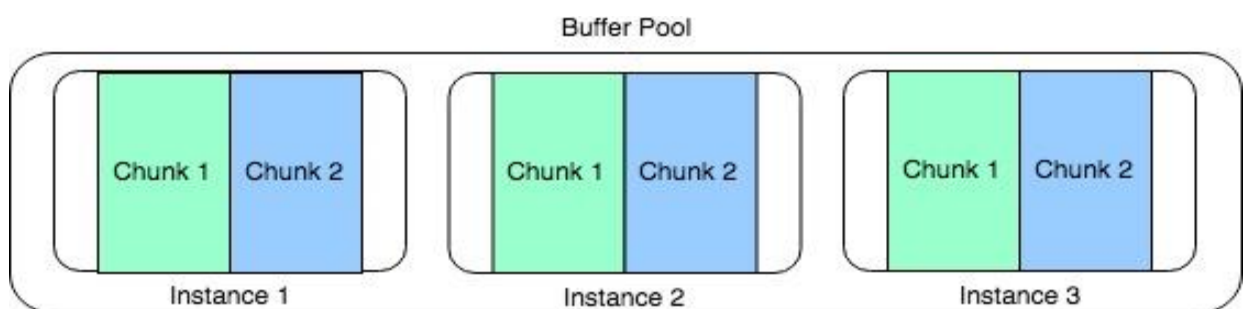


Figura 1 - Modelo simplificado Memória Buffer

Fonte: (DAVID DUCOS, 2018)

Uma forma mais detalhada desta ideia pode ser compreendida na figura 02:

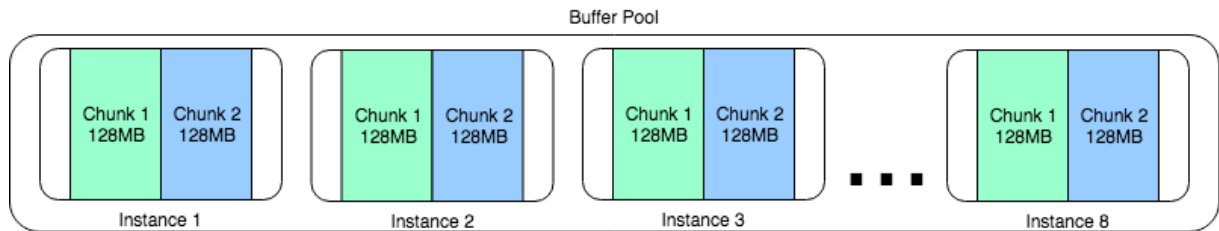


Figura 2 - Modelo detalhado da Memória Buffer
Fonte: (DAVID DUCOS, 2018)

Seguindo o exemplo da figura 02, um pool de buffer de 2 GB com 8 instâncias e blocos no valor padrão (128 MB), obteremos 2 blocos por instância, que acarreta, de acordo com o exemplo, 16 blocos de memória. Dentre os vários benefícios de ter várias instâncias são (DAVID DUCOS, 2018):

- Em um servidor virtual, é possível adicionar memória dinamicamente;
- Para um servidor físico, você pode querer reduzir o uso de memória do banco de dados para abrir caminho para outros processos;

2.4.4 KEY_BUFFER_SIZE

Esta variável determina o tamanho do buffer usado para blocos de índice. O buffer de chave também é conhecido como cache de chaves (MYSQL, 2020).

Esse valor pode aumentar para obter melhor tratamento dos índices para todas as leituras e várias gravações. No entanto, caso o valor seja muito grande (por exemplo, mais de 50% da memória total da máquina), o sistema pode começar a pagnar e se tornar extremamente lento. Isso ocorre porque o MySQL depende da operação sistema para executar o cache do sistema de arquivos para leituras de dados. **então, é necessário** algum espaço para o arquivo cache do sistema. As características dessa variável estão representadas na tabela abaixo 4 (MYSQL, 2020):

Tabela 4 - Características do *key_buffer_size*

COMMAND-LINE FORMAT	--KEY-BUFFER-SIZE=#
SYSTEM VARIABLE	KEY_BUFFER_SIZE
SCOPE	GLOBAL
DYNAMIC	YES
SET_VAR HINT APPLIES	NO
TYPE	INTEGER
MAXIMUM VALUE (64-BIT PLATFORMS)	OS_PER_PROCESS_LIMIT
MAXIMUM VALUE (32-BIT PLATFORMS)	4294967295
DEFAULT VALUE	8388608
MINIMUM VALUE	8

Fonte: (MYSQL, 2020)

2.4.5 VALORES PADRAO DA VARIÁVEIS

A partir de todas as variáveis anteriormente denominadas, se torna necessário exibir os valores iniciais (*default* ou padrão) que servirão de ponto de partida para este trabalho.

Tabela 5 - Valores Padrão da Variáveis do MySQL

VARIÁVEL	VALOR PADRÃO
INNODB_BUFFER_POOL_SIZE	128 MB
INNODB_BUFFER_POOL_INSTANCES	8
INNODB_BUFFER_POOL_CHUNK_SIZE	16 MB
KEY_BUFFER_SIZE	1 MB

Fonte: (MYSQL, 2020)

2.5 TPC-H

O TPC-H é um *benchmark* de suporte à decisão que consiste em um conjunto de consultas orientadas para os negócios e modificações de dados simultâneas (TPC, 2018). As consultas e os dados que povoam o banco de dados foram escolhidos para ter ampla relevância em todo o setor, mantendo um grau suficiente de facilidade na sua implementação. Alguns tópicos que ilustram esse sistema são (TPC, 2018):

- Possibilidade de examinar grandes volumes de dados;
- Variar grau de complexidade das consultas;
- Dar respostas a perguntas de negócios do mundo real;

- Inclui uma ampla variedade de operadores e restrições de seletividade;
- Gerar atividade intensiva por parte do componente servidor de banco de dados do sistema em teste;
- São executados em um banco de dados em conformidade com a população específica e requisitos de escala;

Apesar do fato de que este *benchmark* oferecer um ambiente relativamente representativo, essas métricas não refletem toda a gama de requisitos de suporte à decisão, já que o desempenho relativo deste sistema em produção provavelmente varia de acordo com esses e outros fatores como por exemplo, a carga de trabalho, requisitos de aplicativos específicos e da implementação de sistemas (TPC, 2018).

2.5.1 ENTIDADES E RELACIONAMENTOS DO BANCO DE DADOS

As tabelas e relacionamentos que compõem o esquema do banco de dados do *benchmark* TPC-H são representados por CUSTOMER, NATION, PART, PARTSUPP, REGION, SUPPLIER, ORDERS e LINEITEM. Estas tabelas, assim como os seus relacionamentos são ilustrados na figura abaixo (TPC, 2018).

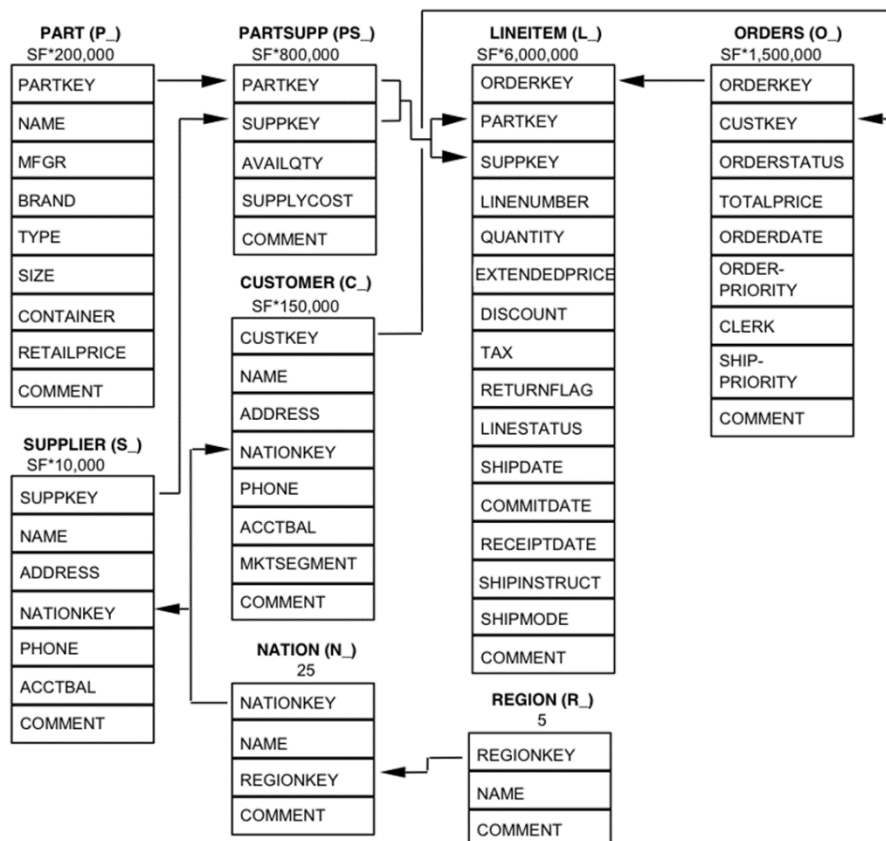


Figura 3 – Esquema Banco De Dados Benchmark **TPC-H**
 Fonte: (TPC, 2018)

2.5.2 GERAÇÃO DOS DADOS E POPULAÇÃO DAS TABELAS

A geração dos dados para popular a base do *benchmark* TPC-H, é realizada através da ferramenta escrita em ANSI C denominada de DBGEN (*Database Generator*). Este programa, depois de compilado e executado, tem a função de automatizar a criação dos dados para população das tabelas.

Caso o tamanho do banco de dados seja definido com referência ao *Scale Factor* (fator de escala) igual a 1, sua base de dados representará um tamanho de aproximadamente 1 GB. Assim, os espaços de armazenamento aproximados dos fatores de escala geralmente utilizados são 1 GB, 10 GB, 30 GB, 100 GB, 300 GB, 1000 GB, 3000 GB, 10.000 GB, 30000 GB ou 100000 GB.

O fator de escala escolhido para a geração do banco de dados, é multiplicado pelo tamanho de cada tabela, como se pode ver na tabela 6.

Tabela 6 - Tamanho Estimado Banco de Dados Fator Escala 10 GB (em tuplas)

TABELA	TUPLAS	TUPLAS * FS
SUPPLIER	10000	100000
PART	200000	2000000
PARTSUPP	800000	8000000
CUSTOMER	150000	1500000
ORDERS	1500000	15000000
LINEITEM	6001215	59986052
NATION	25	25
REGION	5	5
TOTAL	8661245	86586132

Fonte: (TPC, 2018)

Os valores do resultante do produto $TUPLAS * FS$ podem ser encontrados na parte superior de cada tabela, de acordo com a figura 3 (TPC, 2018).

2.5.3 CONSULTAS

O documento do TPC-H especifica os exemplos de saída (*result set*) de uma linha das 22 (vinte e duas) consultas propostas pelo modelo, embora as consultas geralmente produzam várias linhas (TPC, 2018).

Assim, como por exemplo, a consulta 06 (Q06) é representada no código SQL abaixo:

```
SELECT SUM (L_EXTENDEDPRICE * L_DISCOUNT) AS REVENUE
FROM LINEITEM
WHERE L_SHIPDATE >= DATE '[DATE]' AND L_SHIPDATE < DATE '[DATE]' + INTERVAL '1' YEAR
AND L_DISCOUNT BETWEEN [DISCOUNT] - 0.01 AND [DISCOUNT] + 0.01
AND L_QUANTITY < [QUANTITY];
```

Esta consulta apresenta um exemplo saída (*result set*) de acordo com a tabela 7:

Tabela 7 - Resultado de Saída da Consulta 06 do Modelo TPC-H

REVENUE
123141078.23

Fonte: (TPC, 2018)

3. PROCEDIMENTOS METODOLÓGICOS

3.1 METODOLOGIA

A metodologia de elaboração deste trabalho está dividida nas seguintes etapas:

- Etapa 1: Utilização do Modelo Entidade Relacionamento (MER) indicado pelo padrão TPC-H;
- Etapa 2: Instalação e configuração dos bancos de dados;
- Etapa 3: Criação e população das tabelas de uma base de dados de tamanho de 10 GB e outra de 1 GB;
- Etapa 4: Execução das consultas nestas bases recém-criadas utilizando as configurações que vêm por padrão (default) no MySQL depois de instalado;
- Etapa 5: Proposição de novos valores das variáveis com base nos estudos bibliográfico;
- Etapa 6: Alteração dos valores das variáveis de configuração do MYSQL proposta neste trabalho;
- Etapa 6: Execução das consultas propostas pelo *benchmark* TPC-H com os valores das variáveis alteradas;
- Etapa 7: Coleta dos dados e análise de resultados.

3.2 AMBIENTE DE TESTES

Para a realização dos testes foi utilizado o sistema com as seguintes características no que tange aos softwares utilizados:

- Sistema Operacional: Debian GNU/Linux 10 (Buster) x86_64;
- Kernel: 4.19.0-10-amd64;
- MySQL: 8.0.21 for Linux on x86_64 (MySQL Community Server - GPL);
- Engine utilizado nas tabelas: InnoDB;

- Shell: bash 5.0.3;
- DE: XFCE4.

Já no que se refere ao hardware, o teste foi executado nas seguintes configurações:

- CPU: AMD Ryzen 3 2200G (4) @ 3.500GHz;
- GPU: AMD ATI Radeon Vega Series / Radeon Vega Mobile Seri;
- Memória do Sistema: 15026 MB;
- Disk /dev/sda: 111.8 GB, 120034123776 bytes, 234441648 sectors;
- Disk Model: KINGSTON SA400S3;
- Units: sectors of 1 * 512 = 512 bytes.

3.3 COLETA DE DADOS

Para medição do tempo médio de cada consulta foi implementado um script (arquivo) SQL que automatiza a repetição de uma determinada consulta.

Posteriormente, foram realizadas 30 (trinta) repetições da mesma consulta objetivando obter o tempo médio de determinada consulta. O resultado ao final da execução do script, foi possível obter os tempos médios das consultas executadas.

4. ANÁLISE DOS RESULTADOS DA PESQUISA

Como exemplo, os valores de retorno do tempo médio em segundos da primeira consulta (Query 01) quando executadas na base de dados de tamanho de 10 GB são apresentados na tabela 8:

Tabela 8 – Tempos de consultas da Query 01 da Base de Dados de 10 GB

Iteração	Tempo Médio das consultas (segundos)
1	138,45
2	138,58
3	138,51
4	138,47
5	138,56
6	138,61
7	138,60
8	138,54
9	138,66
10	138,57
11	138,69
12	138,52
13	138,59
14	138,61
15	138,63
16	138,75
17	138,52
18	138,58
19	138,62
20	138,65
21	138,60
22	138,58
23	138,59
24	138,61
25	138,51
26	138,52
27	138,59
28	138,74
29	138,51
30	138,72

Depois de concluído o teste da base dados de 10 GB, os valores dos tempos gastos de todas as consultas no banco de dados não alterado são representados na tabela 9:

Tabela 9 - Tempo Médio das Consultas da Base de Dados não Alterada de tamanho 10 GB

Consulta	Média do Tempo das Consultas (s)
1	138,59
2	138,41
3	57,55
4	12,93
5	37,32
6	21,37
7	21,02
8	21,85
9	269,60
10	21,85
11	4,91
12	34,83
13	157,14
14	31,47
15	46,29
16	5,78
17	11,87
18	33,10
19	3,85
20	11,12
21	78,84
22	2,16

Da mesma forma que realizada na base dados de tamanho de 10 GB, os valores obtidos do retorno da primeira consulta (Query 01) quando executadas as consultas na base de dados de tamanho de 1 GB estão demonstradas na tabela 10:

Tabela 10 - Tempos de consulta da Query 01 da Base de Dados de 1 GB

Iteração	Tempo Absoluto (segundos)
1	13,44
2	13,48
3	13,6
4	13,7
5	13,52
6	13,58
7	13,56
8	13,55
9	13,61
10	13,58

11	13,64
12	13,66
13	13,6
14	13,53
15	13,61
16	13,63
17	13,55
18	13,41
19	13,61
20	13,43
21	13,56
22	13,61
23	13,55
24	13,55
25	13,52
26	13,65
27	13,55
28	13,58
29	13,54
30	13,57

Nesta etapa, depois de concluído o teste da base dados de 1 GB, os valores dos tempos gastos das consultas no banco de dados não alterado são representados na tabela 11:

Tabela 11 - Tempo Médio das Consultas da Base de Dados não Alterada de tamanho 1 GB

Consulta	Média do Tempo Absoluto (s)
1	13,57
2	0,11
3	1,11
4	0,58
5	0,86
6	1,78
7	0,77
8	2,39
9	1,65
10	1,15
11	0,12
12	2,6
13	2,88
14	1,77
15	3,78
16	0,32

17	0,27
18	2,66
19	0,17
20	0,81
21	0,76
22	0,21

A próxima etapa do estudo consiste na alteração dos valores das variáveis padrão de acordo com a tabela 12:

Tabela 12 - Valores das Variáveis padrão Após Alteração

VARIÁVEL	VALOR PADRÃO	VALOR DEPOIS DA ALTERAÇÃO
INNODB_BUFFER_POOL_SIZE	128 MB	8.8 GB
INNODB_BUFFER_POOL_INSTANCES	8	2
INNODB_BUFFER_POOL_CHUNK_SIZE	16 MB	32 MB
KEY_BUFFER_SIZE	1 MB	16 MB

Depois de todos os testes, os valores obtidos de todos os bancos criados estão resumidos na tabela 13:

Tabela 13 – Relação entre as médias de todas as bases de dados

1GB				10GB		
Consulta	Média do Tempo (segundos) – Sem alteração	Média do Tempo (segundos) – Depois da alteração	Variação de Percentual do Tempo Médio das consultas	Média do Tempo (segundos) – Sem alteração	Média do Tempo (segundos) – Depois da alteração	Variação de Percentual do Tempo Médio das consultas
1	13,83	13,5657	-1,91%	138,5893	135	-2,59%
2	0,106	0,105	-0,94%	138,4093	1,368	-99,01%
3	4,4657	1,1135	-75,06%	57,549	89,5033	55,53%
4	0,58	0,5827	0,46%	12,9313	7,305	-43,51%
5	2,7257	0,8626	-68,35%	37,322	11,4407	-69,35%
6	2,08	1,7807	-14,39%	21,37	20,2969	-5,02%
7	1,7817	0,7714	-56,70%	21,0213	9,077	-56,82%
8	6,5713	2,3937	-63,57%	21,8473	30,7483	40,74%

9	7,5917	1,6473	-78,30%	269,6007	377,461	40,01%
10	1,762	1,152	-34,62%	21,8473	14,1997	-35,01%
11	0,4513	0,1237	-72,60%	4,9117	1,9287	-60,73%
12	3,015	2,5993	-13,79%	34,8333	132,0593	279,12%
13	1,515	2,8777	89,94%	34,8333	35,5354	2,02%
14	1,5233	1,7743	16,48%	31,4677	105,8877	236,50%
15	4,4237	3,7787	-14,58%	46,294	39,0483	-15,65%
16	0,5923	0,315	-46,82%	5,7837	3,6267	-37,29%
17	1,2233	0,266	-78,26%	11,867	2,3441	-80,25%
18	3,009	2,6623	-11,52%	33,1013	129,46	291,10%
19	0,3039	0,172	-43,40%	3,8523	2,284	-40,71%
20	0,81	0,2207	-72,76%	11,1193	6,9877	-37,16%
21	0,81	5,199	541,85%	78,8357	151,268	91,88%
22	0,2087	0,1633	-21,73%	2,162	1,76	-18,59%

A partir desta última tabela, **observa-se** que a alteração nas variáveis do MySQL implicou na melhora da performance de 18 consultas da base de dados menor (1 GB), sendo representada por uma redução no tempo total do somatório das médias das consultas de 25,69%. Já na base de dados de maior (10 GB), 14 consultas tiveram uma redução de tempo, porém houve um aumento de 25,88% no somatório dos tempos médios das consultas.

Outra interpretação representada explicitada na tabela é que uma base dados menor pode sofrer uma maior interferência positiva quando ajustados os valores das variáveis de configuração alteradas neste estudo.

5. CONSIDERAÇÕES FINAIS

Avaliando este estudo foi possível a instalação, configuração e preparação do Banco de Dados que abrangeu desde o estudo inicial da modelagem dos dados até a implementação do ambiente de testes. Tais experimentos, mesmo sendo criteriosos, foram limitados, uma vez que foram realizados numa máquina de uso pessoal, tendo em vista que em contextos comerciais, são aplicadas em máquinas com o poder computacional bastante superior àquela utilizada nos testes.

Desta forma, dentre as contribuições desse trabalho chega-se à constatação de que o simples aumento de memória disponível para o SGBD não determinará um ganho de performance nas consultas.

Como trabalhos futuros, inicialmente poderiam ser pesquisados a análise de performance do banco além de análise dos parâmetros deste estudo, como por exemplo, a inserção de índices juntamente com o particionamento de tabelas. Outra linha de estudo seria medição de performance quando executados comandos propostos neste trabalho utilizando Bancos de Dados diferentes como, por exemplo, Oracle e PostgreSQL.

6. REFERÊNCIAS

ALMEIDA, A. M. D. SempreUpdate. **Como instalar do LAMP no Ubuntu**, 2019. Disponível em: <<https://sempreupdate.com.br/como-instalar-do-lamp-no-ubuntu/>>. Acesso em: 19 Julho 2020.

ALMEIDA, A. M. D. GitHub. **TPCH**, 2020. Disponível em: <<https://github.com/alexsandro-matias/TPC-H.git>>. Acesso em: 01 Agosto 2020.

ATANAZIO, J. **PostgreSQL - SQL Básico**. [S.l.]: [s.n.], 2019. Disponível em: <https://github.com/juliano777/pgsql_fs2w/blob/master/postgresql_sql_basico.pdf>. Acesso em: 10 Outubro 2020.

CABRAL, S.; MURPHY, K. **MySQL Administrator's Bible**. Indianapolis: Wiley, 2009.

DAVID DUCOS. Percona. **InnoDB Buffer Pool Resizing: Chunk Change**, 2018. Disponível em: <<https://www.percona.com/blog/2018/06/19/chunk-change-innodb-buffer-pool-resizing/>>. Acesso em: 08 Janeiro 2021.

DB-ENGINES. **DB-Engines Ranking**, 14 Junho 2020. Disponível em: <<https://db-engines.com/en/ranking>>. Acesso em: 14 Junho 2020.

MYSQL. **MySQL 8.0 Reference Manual**, 09 Setembro 2020. Disponível em: <<https://downloads.mysql.com/docs/refman-8.0-en.pdf>>. Acesso em: 2020.

RAMAKRISHNAN, R.; GEHRKE, J. **Sistemas de Gerenciamento Sistemas de Gerenciamento de Banco de Dados**. São Paulo: McGraw-Hill, 2009.

SCALEGRID. **What is an InnoDB Buffer Pool?**, 2018. Disponível em: <<https://scalegrid.io/blog/calculating-innodb-buffer-pool-size-for-your-mysql-server/>>.

Acesso em: 08 Janeiro 2020.

SCHWARTZ, B.; TKACHENKO, ; ZAITSEV,. **High Performance MySQL**. Third Edition. ed. Sebastopol: O'Reilly Media, 2012.

TARGETTRUST. **Performance e Otimização de Banco de Dados MySQL**, 2017. Disponível em: <<http://materiais.targettrust.com.br/ebook-otimizacao-banco-de-dados-mysql-lp>>. Acesso em: 19 Setembro 2020.

TPC. **BENCHMARK (Decision Support) Standard Specification Revision**, 2018. Disponível em: <http://www.tpc.org/tpc_documents_current_versions/pdf/tpc-h_v2.18.0.pdf>.

WIKIPEDIA. **David DeWitt**, 14 Julho 2020. Disponível em: <https://en.wikipedia.org/wiki/David_DeWitt>. Acesso em: 01 Setembro 2020.