

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
ESPECIALIZAÇÃO EM BANCO DE DADOS**

GILVANO PIONTKOSKI

**COMPARATIVO DE DESEMPENHO DE CONSULTAS *SQL* ENTRE
BANCO DE DADOS EM MEMÓRIA *RAM* E EM *SSD***

MONOGRAFIA DE ESPECIALIZAÇÃO

**PATO BRANCO
2017**

GILVANO PIONTKOSKI

**COMPARATIVO DE DESEMPENHO DE CONSULTAS *SQL* ENTRE
BANCO DE DADOS EM MEMÓRIA *RAM* E EM *SSD***

Trabalho de Conclusão de Curso, apresentado ao II Curso de Especialização em Banco de Dados, da Universidade Tecnológica Federal do Paraná, campus Pato Branco, como requisito parcial para obtenção do título de Especialista.

Orientador(a): Prof(a). Fabiano Luiz Carniel.

PATO BRANCO

2017



TERMO DE APROVAÇÃO

COMPARATIVO DE DESEMPENHO DE CONSULTAS SQL ENTRE BANCO DE DADOS EM MEMÓRIA RAM E EM DISCO SSD.

por

GILVANO PIONTKOSKI

Este Trabalho de Conclusão de Curso foi apresentado em 24 fevereiro de 2017 como requisito parcial para a obtenção do título de Especialista em Banco de Dados. O(a) candidato(a) foi arguido(a) pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

Fabiano Carniel
Prof.(a) Orientador(a)

Ives Rene Venturini Pola
Membro titular

Bruno Cesar Ribas
Membro titular

“O Termo de Aprovação assinado encontra-se na Coordenação do Curso”

AGRADECIMENTOS

Agradeço em especial a minha amada esposa, que me apoiou desde o início desta especialização e sempre me lembrou de como era importante a conclusão de mais esta etapa da minha vida.

RESUMO

PIONTKOSKI, Gilvano. **Comparativo de desempenho de consultas *sql* entre banco de dados em memória *ram* e em *ssd***. 2017. 28 f. Monografia (II Curso de Especialização em Banco de Dados) - Universidade Tecnológica Federal do Paraná. Pato Branco, 2012.

O principal objetivo deste trabalho foi realizar um comparativo de desempenho de consultas SQL em duas novas tecnologias que estão se sobre saindo em relação ao modelo tradicional, que são o Banco de Dados em Memória (In-memory Database) e o Banco de Dados em Disco Sólidos (SSD). Para simular essas consultas foram criados bancos de dados idênticos para cada tecnologia, estes bancos simulavam o controle de vendas por produto e cliente. Nestes cenários foram executadas consultas que retornam uma grande quantidade de resultados com os dados das vendas de vários meses, consulta que no cenário tradicional utilizando Disco Rígido, pode demorar alguns minutos. A mesma consulta foi executada nos outros dois ambientes, onde foi medido desempenho destas consultas e ao final verificou-se que a melhor alternativa foi a In-memory para um volume menor de informações e a SSD para um volume maior de dados.

Palavras-chave: Banco de Dados em Memória. Banco de Dados em *SSD*. Consulta *SQL*.

ABSTRACT

PIONTKOSKI, Gilvano. **Comparing performance of *sql* queries in databases using *ram* and *ssd* memories**. 2017. 28 p. Monography (II Specialization Course in Database) - Federal University of Technology - Parana. Pato Branco, 2017.

The main objective of this work was to compare performance of SQL queries in two new technologies that are about to come out in relation to the traditional model, which are the In-memory Database and the Disk Database Solids (SSD). To simulate these queries, identical databases were created for each technology, these banks simulated sales control by product and customer. In these scenarios queries were executed that return a large amount of results with sales data of several months, which in the traditional scenario using Hard Disk can take a few minutes. The same query was performed in the other two environments, where the performance of these queries was measured and at the end it was verified that the best alternative was the In-memory for a smaller volume of information and the SSD for a larger volume of data.

Keywords: In-memory Database. Databases in SSD. SQL queries.

LISTA DE SIGLAS

HDD	<i>Hard Disk Drive</i>
PL	<i>Procedural Language</i>
RAM	<i>Random Access Memory</i>
RPM	Rotações por minuto
SGBD	Sistema de Gerenciamento de Banco de Dados
SQL	<i>Structured Query Language,</i>
SSD	<i>Solid State Drive</i>

LISTA DE FIGURAS

Figura 1: Comparativo Disco Rígido x <i>SSD</i>	10
Figura 2: Funcionamento - Banco de dados em memória.....	12

LISTA DE QUADROS

Quadro 1: Vantagens da utilização do banco de dados.....	9
Quadro 2: Resultados de estudos anteriores comparativos.	13
Quadro 3: Ferramentas e tecnologias utilizadas	14
Quadro 4: Recursos consumidos na execução das consultas	25

LISTA DE SCRIPTS

Script 1: Criação dos tablespaces	15
Script 2: Criação da tabela TABELA_HDD	15
Script 3: Bloco anônimo PL/SQL para geração e inserção dos dados.....	16
Script 4: Função que testa a execução da consulta SQL na tabela_hdd	17
Script 5: Script que executa dez vezes os testes na tabela tabela_hdd e grava os resultados.....	18
Script 6: Consulta na tabela v\$sqlarea	19

LISTA DE GRÁFICOS

Gráfico 1 - Resultado de cada consulta com 1.000.000 de linhas.....	20
Gráfico 2 - Tempo médio da consulta com 1.000.000 de linhas.....	21
Gráfico 3 - Resultado de cada consulta com 2.500.000 linhas.....	21
Gráfico 4 - Tempo médio da consulta com 2.500.000 de linhas.....	22
Gráfico 5 - Resultado de cada consulta com 5.000.000 linhas.....	22
Gráfico 6 - Tempo médio da consulta com 5.000.000 de linhas.....	23
Gráfico 7 - Resultado de cada consulta com 10.000.000 linhas.....	23
Gráfico 8 - Tempo médio da consulta com 10.000.000 de linhas.....	24

SUMÁRIO

1 INTRODUÇÃO	5
1.1 CONSIDERAÇÕES INICIAIS	5
1.2 OBJETIVOS	6
1.2.1 Objetivo Geral	6
1.2.2 Objetivos Específicos	6
1.3 JUSTIFICATIVA.....	6
1.4 ORGANIZAÇÃO DO TEXTO	7
2 REFERENCIAL TEÓRICO	8
2.1 BANCO DE DADOS.....	8
2.2 DISCO RÍGIDO.....	9
2.3 SSD	10
2.4 BANCO DE DADOS EM MEMÓRIA.....	11
2.5 ESTUDOS RELACIONADOS AO TEMA.....	12
3 MATERIAIS E MÉTODO	14
3.1 FERRAMENTAS E TECNOLOGIAS	14
3.2 MÉTODO	14
4 RESULTADOS E DISCUSSÃO	20
5 CONCLUSÃO	23
REFERÊNCIAS	27

1 INTRODUÇÃO

Este capítulo apresenta as considerações iniciais, com uma visão geral do trabalho, os objetivos e a justificativa do mesmo e a organização do texto.

1.1 CONSIDERAÇÕES INICIAIS

No modelo tradicional de Banco de Dados a estrutura de armazenamento predominantemente utilizada são os Discos Rígidos, que são estruturas seguras, confiáveis e que suportam grandes volumes de dados, porém estão há muito tempo sem grande evolução em questão de desempenho, com tempos de leitura e gravação muito altos. Com o constante crescimento do volume de informações armazenados, a busca e recuperação dessas informações tendem a ser mais lentas. Em contra partida as empresas necessitam de consultas e relatórios para visualizar seus resultados de maneira mais ágil, não aceitando que a geração dessas informações demande uma quantidade elevada de tempo.

Com o surgimento do *SSD (Solid State Drive)* o cenário mudou completamente, visto que os mesmos possuem baixo tempo de gravação e leitura. A redução da leitura/escrita em disco melhora o desempenho de maneira considerável (ELMASRI; NAVATHE, 2011, p. 26).

Em paralelo ao surgimento dos *SSD*, as grandes empresas desenvolvedoras de sistemas gerenciadores de banco de dados (SGDB) buscaram desenvolver alternativas para contornar o problema da baixa velocidade dos Discos Rígidos. Uma dessas soluções é o Banco de Dados em Memória (*In-memory Database*), na qual os dados não vão para o disco, e sim para a memória *RAM (Random Access Memory)*, que tem tempo de leitura e gravação muito menores que às dos discos rígidos.

Conforme relatam Elmasri e Navathe (2011, p.26) “o banco de dados e o catálogo do SGBD, habitualmente, são armazenados em disco. O acesso ao disco é controlado, em especial, pelo sistema operacional(SO), que escalona a leitura/escrita em disco.”

1.2 OBJETIVOS

1.2.1 Objetivo Geral

Realizar um comparativo do desempenho de consultas *SQL (Structured Query Language)* em Banco de Dados em Memória *RAM* e Banco de Dados utilizando *SSD*.

1.2.2 Objetivos Específicos

Para auxiliar o alcance do objetivo geral, traçaram-se os seguintes objetivos específicos:

- Criar o Banco de Dados no Oracle 12c;
- Gerar as estruturas do Banco de Dados em Disco Rígido, em Banco de Dados em *SSD* e em Banco de Dados em Memória;
- Aferir o desempenho de consulta *SQL* Banco de Dados em Disco Rígido, em Banco de Dados em *SSD* e em Banco de Dados em Memória;
- Comparar os resultados dos testes realizados;
- Apresentar a melhor alternativa ao Banco de Dados em Disco Rígido.

1.3 JUSTIFICATIVA

Muitas empresas precisam fazer consultas detalhadas sobre as movimentações de seus produtos e/ou clientes, algumas dessas consultas quando executadas em banco de dados armazenados em Discos Rígidos tendem a ser lentas, podendo levar de alguns minutos à várias horas para serem executadas.

Visando encontrar a melhor solução para a otimização de desempenho das consultas, foi realizado um comparativo de desempenho de consultas *SQL* em duas novas tecnologias que estão se sobre saindo em relação ao modelo tradicional, que são o Banco de Dados em Memória (*In-memory Database*) e o Banco de Dados em Disco de Estado Sólido (*SSD*).

1.4 ORGANIZAÇÃO DO TEXTO

Este texto está organizado em capítulos, este é o primeiro capítulo e apresenta a ideia e o contexto do comparativo, incluindo os objetivos e a justificativa.

O Capítulo 2 contém o referencial teórico que fundamenta a proposta conceitual da pesquisa. O referencial teórico está centrado em apresentar uma breve explanação do que é banco de dados, uma visão do que é o banco de dados em memória, a evolução na velocidade de gravação e leitura com os *SSD*.

No Capítulo 3 estão os materiais e o métodos empregados no desenvolvimento deste trabalho.

O Capítulo 4 mostra os resultados obtidos e uma breve discussão sobre os mesmos.

E por último o Capítulo 5 apresenta a conclusão com as considerações finais.

2 REFERENCIAL TEÓRICO

Este capítulo apresenta os conceitos básicos e outros estudos relacionados ao tema proposto neste trabalho. A revisão da literatura subdividiu-se em cinco itens: (2.1) banco de dados, (2.2) disco rígido, (2.3) *SSD*, (2.4) banco de dados em memória, e (2.5) estudos relacionados ao tema.

2.1 BANCO DE DADOS

Os sistemas de bancos de dados vieram para proporcionar um domínio mais eficiente perante as informações das empresas, auxiliando no controle de cadastros de produtos, clientes, fornecedores, movimentações de compras e vendas, empréstimos, saques, enfim tudo que um dia foi controlado por meio de anotações em cadernos e planilhas, hoje são facilmente gravados em tabelas de banco de dados.

Elmasri e Navathe (2011) citam que um banco de dados é uma coleção de dados relacionados, ou seja, o termo “dados” significa fatos conhecidos que podem ser registrados, que possuem significado implícito e podem ser interpretados.

Já na visão de Vinhas (2016) um banco de dados pode ser definido como uma série de dados organizados de uma forma que um computador possa armazená-los e recuperá-los de maneira eficiente. O autor ainda complementa que os bancos de dados transformaram os computadores em repositórios de informações.

Date (2003) afirma que um banco de dados é uma coleção de dados persistentes, usada pelos sistemas de aplicação de uma determinada empresa. Nesse contexto, o termo “empresa” é utilizado como uma definição genérica para qualquer organização comercial, científica, técnica ou autônoma.

Por sua vez Ferreira *et al.* (2005) definem como uma coleção de dados relacionados, com valores armazenados e que servem para a obtenção de informações. Com isso, os dados podem ser compilados e processados de acordo com as consultas e análises executadas.

Matsumoto (2006) ressalta que a partir das informações coletadas dos bancos de dados é possível extrair informações mais completas, sucintas, resumidas, mas de maior inteligibilidade, facilitando assim as interpretações e as tomadas de decisões.

Perante isso, Batista (2006) complementa que um banco de dados deve ser estruturado de acordo com as necessidades informacionais dos usuários ou dos grupos de usuários que

utilizarão esses dados. Pois, o banco associado ao sistema computacional conseguirá suprir essas necessidades de uma maneira mais correta.

Date (2003) cita algumas vantagens na utilização dos bancos de dados ao invés de anotações ou métodos tradicionais, conforme demonstrado no Quadro 1:

Vantagem	Justificativa
Densidade	Não há necessidade de arquivos em papel, possivelmente volumosos.
Velocidade	A máquina pode obter e atualizar dados com rapidez muito maior que o ser humano.
Menos trabalho monótono	Grande parte do empenho em manter arquivos à mão é eliminada. As tarefas mecânicas são sempre feitas com melhor qualidade por máquinas.
Atualidade	Informações precisas e atualizadas estão disponíveis a qualquer momento sob consulta.
Proteção	Os dados podem ser mais bem protegidos contra perda não intencional e acesso ilegal.

Quadro 1: Vantagens da utilização do banco de dados.

Fonte: Adaptado de Date (2003)

É notável que são variadas as vantagens em utilizar um banco de dados na operacionalização das informações organizacionais. Há uma evolução contínua nessa tecnologia, e em alguns casos sua utilização torna-se indispensável.

Gehrke e Ramakrishnan (2008) ressaltam que a quantidade de informações das organizações é infinita, gerando a necessidade de um armazenamento mais seguro, pois os dados passam a ter um valor e ser considerado um ativo organizacional, fato este amplamente conhecido.

Nos tópicos seguintes desta seção são conceituadas as diferentes formas e opções de armazenamento para bancos de dados, conforme definição e observações de publicações e estudos anteriores.

2.2 DISCO RÍGIDO

Os discos rígidos, segundo DEVMEDIA (2016) foram os primeiros dispositivos de armazenamento em larga escala empregados. Esse tipo de *hardware* de armazenamento é composto por discos magnéticos (ou discos rígidos) que armazenam as informações, onde a leitura e gravação dos dados são efetuadas por uma agulha mecânica.

Os discos magnéticos são amplamente utilizados por aplicações de bancos de dados, pois fornecem um acesso harmônico aos dados gravados. Contudo, por se tratar de uma

estrutura de armazenamento que possui partes mecânicas, pode haver um menor desempenho devido a fatores de “latência rotacional” e “tempo de busca” (GEHRKE; RAMAKRISHNAN, 2008).

Nesse sentido, Oliveira (2014) afirma que o desempenho de um disco rígido é afetado diretamente pela latência na busca de informação. O autor também destaca que a latência rotacional é um dos principais entraves para o acesso, leitura e retorno das informações, pois há atrasos de tempo provocados pela falta de sincronia entre o setor desejado e o cabeçote, isso ocorre nas transferências de dados, na busca de informações e na meia rotação.

A Figura 1 demonstra as diferenças estruturais entre o Disco Rígido e o SSD.



Figura 1: Comparativo Disco Rígido x SSD.

Fonte: TECHTUDO (2013)

Ao observar a figura anterior nota-se as diferenças estruturais entre os dois tipos de discos, no caso do Disco Rígido a latência é provocada pelo tempo necessário entre a leitura das informações e o retorno dos dados.

2.3 SSD

A tecnologia *SSD* avançou rapidamente nos últimos anos, pois o tempo de transferência de dados entre o *SSD* e a memória *RAM* é muito menor que os *hard disk* tradicionais. Conforme citado em CEAVI/UEDESC (2013), um *SSD* pode ser compreendido como um “*HDD*” que não utiliza discos magnéticos para o armazenamento de dados, mas sim, *chips* de memória *flash*.

O *SSD* tem uma resposta de velocidade muito maior se comparada a tradicional arquitetura de *hard disk*. Onde um dos motivos desse ganho de velocidade e desempenho, é o fato dos *SSD* não possuírem estrutura mecânica, ou seja, não ter uma leitura de disco (OLIVEIRA, 2014). Em concordância, Geier (2015) expõe que justamente essa ausência de uma estrutura mecânica torna os *SSD* mais resistentes a quedas e exposição a temperaturas elevadas quando comparados com os *hard disk*.

Em Devmedia (2016) há um destaque para o custo ainda elevado da tecnologia *SSD*, isso pode ser justificado quando observado que se trata de uma tecnologia relativamente nova e que oferece, na maioria das vezes, maiores vantagens que o armazenamento tradicional em discos rígidos.

2.4 BANCO DE DADOS EM MEMÓRIA

Os bancos de dados do modelo tradicional já tem a funcionalidade de carregar os dados para a memória *RAM*, que é o *cache* de dados. Segundo a ORACLE (2015) o banco de dados em memória é uma nova funcionalidade que foi implementada na versão 12c. O banco de dados em memória é similar ao *cache buffer* do banco de dados tradicional, com a diferença que todos os dados ficam na memória e não somente os mais acessados recentemente.

A implementação da Oracle permite escolher quais tabelas vão ser carregadas em memória, assim não necessitando carregar todo o banco de dados para memória como acontece em um banco de dados puramente em memória.

Essa forma de busca de dados é importante, pois com o constante crescimento do volume de informações de algumas empresas, a tarefa de consultar esses dados muitas vezes não é executada rapidamente no modelo tradicional, utilizando somente o disco rígido, com o banco de dados em memória o retorno das informações pode ser efetuada em um menor tempo.

Segundo informações da ORACLE (2015) os bancos de dados em memória se diferenciam pelo fato de armazenarem os dados em linhas, porém quando ocupam a parte *in memory* da SGA (System Global Area) são agrupados por blocos de colunas, que otimizam o retorno das consultas e essas podem ser varridas rapidamente, através de técnicas como o SIMD processamento de vetor. A Figura 2 demonstra o agrupamento descrito anteriormente.

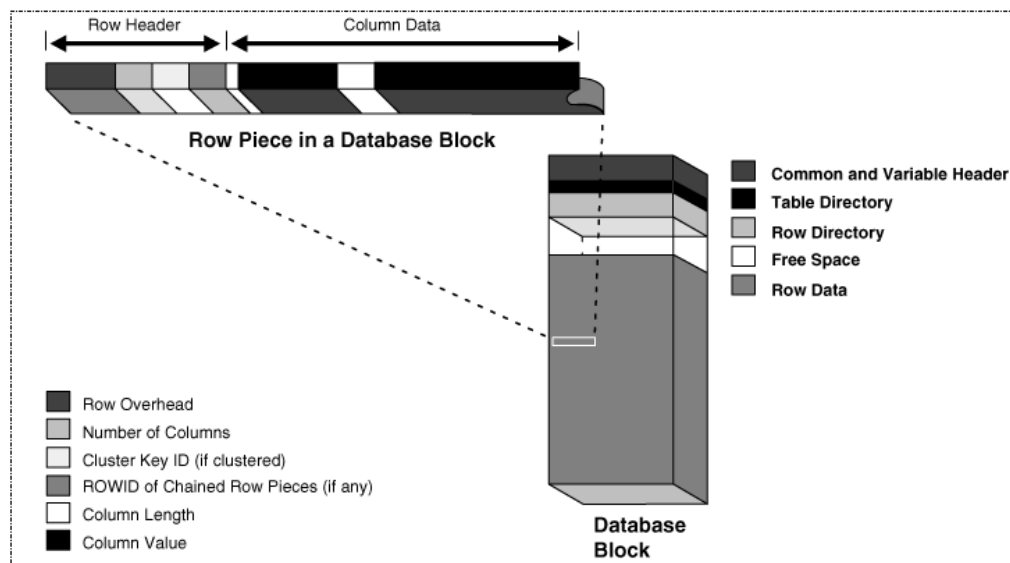


Figura 2: Funcionamento - Banco de dados em memória.

Fonte: ORACLE (2015).

Segundo artigo da PROGRESS (2016) esse tipo de arquitetura carrega os bancos de dados para a memória, com isso há uma obtenção mais rápida perante uma grande quantidade de dados. Contudo, os dados em memória são uma parte da solução para diminuição da velocidade do retorno dos dados, destaca-se que o conjunto do *hardware* pode auxiliar na maximização de desempenho.

2.5 ESTUDOS RELACIONADOS AO TEMA

Na literatura é possível encontrar estudos que realizaram comparação de desempenho entre o retorno de consultas em diferentes meios de armazenamento de banco de dados. No Quadro 02, são expostos tais estudos que realizaram essa comparação e seus principais achados.

PUBLICAÇÃO	AUTOR	OBJETIVO	RESULTADO
<i>HD, SSD</i> ou disco híbrido, qual o melhor para sua empresa?	DIEGO MACÊDO (2013)	Comparação empírica entre Discos rígidos (<i>HD</i>), <i>SSD</i> e discos híbridos.	<i>HDs</i> podem ser utilizados no armazenamento de grande quantidade de memória, <i>SSD</i> apresentam maior velocidade e retorno das informações, os discos híbridos possuem baixo custo, alta capacidade e gerenciamento.

Comparando o desempenho do disco no SGBD em <i>HD</i> , <i>SSD</i> , <i>Ramdrive</i> e <i>Cloud</i> .	PICHILIANI (2015)	Comparar o desempenho do disco no SGBD em <i>HD</i> , <i>SSD</i> , <i>Ramdrive</i> e <i>Cloud</i> .	<i>Ramdrive</i> e <i>SSD</i> apresentaram melhor desempenho em relação as demais opções. As soluções <i>Cloud</i> deixaram a desejar no quesito velocidade.
Guia: conheça as diferenças entre os dispositivos de armazenamento.	OLHAR DIGITAL (2015)	Comparar o desempenho de <i>HD</i> , <i>SSD</i> e <i>SSHDs</i> .	<i>SSD</i> se destacam por possuir maior desempenho no retorno das informações.
Testando o desempenho do <i>HD</i> com banco de dados.	DEVMEDIA (2016)	Comparação técnica de desempenho de operações entre tecnologias de armazenamento (<i>HDs</i> mecânicos, <i>SSD</i> , <i>ramdrivers</i> (disco em memória) e soluções na nuvem.	A opção de disco em memória (<i>ramdrive</i>) é muito superior às demais. Os dados mostram que a utilização de um <i>drive SSD</i> é superior aos <i>drives</i> mecânicos e às opções de armazenamento de dados em nuvem testadas.
Comparação entre um <i>HD</i> Sata em um <i>Solid State Drive (SSD)</i> na instalação, manuseio e acesso de uma base de dados <i>MySQL</i> .	COSTA e PADOVANI (2016)	Comparar o desempenho do <i>HD</i> e um <i>SSD</i> no manuseio de uma base de dados <i>MySQL</i> .	O <i>SSD</i> se mostrou melhor desempenho no retorno das consultas em relação ao <i>HD</i> .

Quadro 2: Resultados de estudos anteriores comparativos.

Ao observar o Quadro 2 é possível verificar que estudos e comparativos sobre o meio de armazenamento mais eficiente estão sendo elaborados com frequência nos últimos anos. Com isto, pode-se perceber a importância do tema e seus resultados no meio tecnológico e empresarial.

Nota-se que os autores encontraram resultados similares em seus experimentos, nem todos os trabalhos fizeram comparativos com banco de dados em memória por ser uma tecnologia relativamente nova. Os estudos mostram que o banco em memória e o *SSD* se destacam perante as demais tecnologias, porém apresentam um custo mais elevado. Devmedia (2016) utilizou a tecnologia *ramdriver* para simular o banco em memória, essa solução não é a mesma apresentada neste estudo, visto que *ramdriver* é um software que faz uso da memória *RAM* para simular um disco de armazenamento, a tecnologia aqui utilizada foi *in-memory data base* da Oracle disponível na versão 12c *Enterprise*.

3 MATERIAIS E MÉTODO

Este capítulo apresenta os materiais e o métodos utilizados na realização deste trabalho. Os materiais se referem às tecnologias e ferramentas para criação e configuração do banco de dados, e também as usadas para a aferição dos testes. O método contém as etapas com os principais procedimentos utilizados para o desenvolvimento do sistema.

3.1 FERRAMENTAS E TECNOLOGIAS

O Quadro 3 contém informações sobre as tecnologias e ferramentas utilizadas.

NOME	VERSÃO	LOCALIZAÇÃO	DESCRIÇÃO DE USO
Oracle Database 12c	12.1.0.2	https://www.oracle.com	Sistema gerenciador de banco de dados.
SQL Developer	4.2	http://www.oracle.com/technetwork/developer-tools/sql-developer/downloads/index.html	Ambiente de desenvolvimento PL/SQL, utilizado para realizar consultas ao banco de dados
SQLPlus	12.1.0.2	https://www.oracle.com	Ferramenta de linha comando para execução de comandos SQL.

Quadro 3: Ferramentas e tecnologias utilizadas

3.2 MÉTODO

Por se tratar de um estudo comparativo de desempenho de consultas SQL, foi necessário criar um banco de dados com uma tabela em cada modelo proposto, execução e mensuração das consultas. Para isso foi utilizado um notebook Dell Inspiron 5437, com Windows 10 Pro, processador intel core i5 4200U 2300 MHz, 12gb de memória RAM com 1600 MHz de frequência, disco rígido(HDD) Seagate ST1000LM024 de 5400 RPM com 1 terabyte de capacidade de armazenamento e um disco de estado sólido (SSD) Kingston SV300S37A240G com taxas de gravação e leitura de 450 Mb/s capacidade de armazenamento de 250 gigabytes.

Foi feita uma instalação padrão do Oracle 12c Enterprise, com somente duas alterações nos parâmetros de memória, o `sga_target` foi alterado para 6 gigabytes e o `inmemory_size` para 5 gigabytes, foram criados três *tablespaces* para armazenar as tabelas, `tbs_hdd`, `tbs_ssd` e `tbs_mem`, um para cada modelo de armazenamento, os três com tamanho inicial de 1 gigabyte, o `tbs_hdd` foi criado no disco rígido, o `tbs_ssd` no disco de estado sólido e o `tbs_mem` foi criado

inicialmente no disco rígido e posteriormente transferido para memória conforme comando contido no *script 1*.

```
CREATE TABLESPACE TBS_HDD
DATAFILE 'D:\DADOS\TABLESPACE\TBS_HDD.DBS'
SIZE 1000M AUTOEXTEND ON NEXT 50M
DEFAULT NO INMEMORY
ONLINE;

CREATE TABLESPACE TBS_SSD
DATAFILE 'C:\DADOS\TABLESPACE\TBS_SSD.DBS'
SIZE 1000M AUTOEXTEND ON NEXT 50M
DEFAULT NO INMEMORY
ONLINE;

CREATE TABLESPACE TBS_MEM
DATAFILE 'D:\DADOS\TABLESPACE\TBS_MEM.DBS'
SIZE 1000M AUTOEXTEND ON NEXT 50M
ONLINE;

ALTER TABLESPACE TBS_MEM DEFAULT INMEMORY;
```

Script 1: Criação dos tablespaces

Em cada *tablespace* foi criada uma tabela semelhante à uma tabela de um sistema real que faz controle de vendas por produto e cliente, esta tabela foi criada com as colunas *id*, *id_nota*, *id_cliente*, *id_produto* do tipo inteiro, *valor1*, *valor2*, *valor3*, *valor4* e *valor5* do tipo decimal, *caracter1*, *caracter2*, *caracter3*, *caracter4* e *caracter5* do tipo *string*, e *data1* e *data2* do tipo *date*. A coluna *id* foi criada como chave primária e foi criado um índice para essa coluna. No *script 2* é mostrado o comando de criação da tabela *tabela_hdd*, um *script* quase idêntico foi montado para a criação das outras duas tabelas, somente com as diferenças de nome da tabela e *tablespace* de armazenamento, ficando *tabela_ssd* armazenada no *tablespace* *tbs_ssd* e *tabela_mem* armazenada no *tablespace* *tbs_mem*.

```
CREATE TABLE TABELA_HDD(
  ID NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,
  ID_NOTA NUMBER,
  ID_CLIENTE NUMBER,
  ID_PRODUTO NUMBER,
  VALOR1 NUMBER(38,2),
  VALOR2 NUMBER(38,2),
  VALOR3 NUMBER(38,2),
  VALOR4 NUMBER(38,2),
  VALOR5 NUMBER(38,2),
  CHARACTER1 VARCHAR2(1),
  CHARACTER2 VARCHAR2(2),
  CHARACTER3 VARCHAR2(3),
  CHARACTER4 VARCHAR2(4),
  CHARACTER5 VARCHAR2(5),
  DATA1 DATE,
  DATA2 DATE,
  CONSTRAINT PK_TABELA_HDD PRIMARY KEY(ID)
)
TABLESPACE TBS_HDD;
```

Script 2: Criação da tabela TABELA_HDD

Criadas as tabelas, então foi necessário gerar e inserir dados nas mesmas, para fazer este trabalho foi criado o bloco anônimo PL/SQL mostrado no *script 3*, que gera 20.000.000 de linhas, visto que para o teste de desempenho eram necessárias muitas linhas em cada tabela. Cada linha foi gerada da seguinte maneira: o campo *id* foi gerado automaticamente pelo banco

de dados por se tratar de um campo *identity*, o campo *id_nota* recebeu uma valor sequencial de 1 a 20.000.000 conforme a iteração do loop executado, o campo *id_cliente* recebeu números aleatórios entre 1 e 10000, o campo *id_produto* números aleatórios entre 10000 e 30000, os campos de valor receberam números aleatórios entre 1 e 100.000 com precisão de duas casas decimais, os campos caracteres receberam caracteres aleatórios, por final os campos *datas* receberam *datas* aleatórias entre os anos 2010 e 2020.

```

DECLARE
  I NUMBER;
  ID_CLI NUMBER;
  ID_PROD NUMBER;
  N1 NUMBER;
  N2 NUMBER;
  N3 NUMBER;
  N4 NUMBER;
  N5 NUMBER;
  S1 VARCHAR2(1);
  S2 VARCHAR2(2);
  S3 VARCHAR2(3);
  S4 VARCHAR2(4);
  S5 VARCHAR2(5);
  DATA1 DATE;
  DATA2 DATE;
BEGIN
  FOR I IN 1..20000000
  LOOP
    SELECT
      /* Gera números aleatórios entre 1 e 10000 para código do cliente */
      ROUND(DBMS_RANDOM.VALUE (1, 10000)) ,
      /* Gera números aleatórios entre 10000 e 30000 para código do produto */
      ROUND(DBMS_RANDOM.VALUE (10000, 30000)) ,
      /* Gera números aleatórios para as colunas numéricas */
      ROUND(DBMS_RANDOM.VALUE (1, 100000), 2) ,
      ROUND(DBMS_RANDOM.VALUE (1, 100000), 2) ,
      ROUND(DBMS_RANDOM.VALUE (1, 100000), 2) ,
      ROUND(DBMS_RANDOM.VALUE (1, 100000), 2) ,
      ROUND(DBMS_RANDOM.VALUE (1, 100000), 2) ,
      /* Gera as caracteres aleatórias para os campos texto */
      DBMS_RANDOM.STRING('U', 1) ,
      DBMS_RANDOM.STRING('U', 2) ,
      DBMS_RANDOM.STRING('U', 3) ,
      DBMS_RANDOM.STRING('U', 4) ,
      DBMS_RANDOM.STRING('U', 5) ,
      /* Gera as datas aleatórias para os campos data */
      TO_DATE(TRUNC(DBMS_RANDOM.VALUE(2455196,2458846)), 'J') ,
      TO_DATE(TRUNC(DBMS_RANDOM.VALUE(2455196,2458846)), 'J')
    INTO ID_CLI, ID_PROD,
         N1, N2, N3, N4, N5,
         S1, S2, S3, S4, S5,
         DATA1, DATA2
  FROM DUAL;
  /* Insere uma linha na tabela em cada loop */
  INSERT INTO TABELA_HDD
  (
    ID, ID_NOTA, ID_CLIENTE, ID_PRODUTO,
    VALOR1, VALOR2, VALOR3, VALOR4, VALOR5,
    CARACTER1, CARACTER2, CARACTER3, CARACTER4, CARACTER5,
    DATA1, DATA2)
  VALUES(
    NULL, I, ID_CLI, ID_PROD,
    N1, N2, N3, N4, N5,
    S1, S2, S3, S4, S5,
    DATA1, DATA2);
  IF (MOD(I,10000)=0)THEN      /* Commit a cada 10000 linhas */
    COMMIT;
  END IF;
  END LOOP;
END;

```

Script 3: Bloco anônimo PL/SQL para geração e inserção dos dados

Para testar o desempenho das consultas foram criadas as funções de banco de dados `testa_tabela_hdd`, `testa_tabela_ssd` e `testa_tabela_mem`, essas funções executam consultas em suas respectivas tabelas com base nos parâmetros passados, o primeiro parâmetro é número de execução, o segundo é a linha inicial e o terceiro é o linha final, para facilitar os chamada dessas funções, os parâmetros de número de linhas são multiplicados por 1.000.000, sendo assim se for chamada à função `testa_tabela_hdd` com os parâmetros (1, 1, 2), ela irá executar uma consulta na tabela `tabela_hdd` recuperando as linhas entre 1.000.000 e 2.000.000 e com o retorno “EXEC: 1 - INI: 1000000 - FIM: 2000000”. Para medir somente o processo de consulta no banco de dados não foi retornado todas as linhas selecionadas, visto que dependendo do programa que é chamada os tempos de retorno e exibição dos dados poderiam variar muito. No *script* 4 é exibido o comando de criação da função `testa_tabela_hdd`, as funções `testa_tabela_ssd` e `testa_tabela_mem` são semelhantes, com as diferenças de nome e tabela em qual a consulta é executada.

```
CREATE OR REPLACE FUNCTION TESTA_TABELA_HDD
(
  NEXEC IN NUMBER,
  NINI IN NUMBER,
  NFIM IN NUMBER
) RETURN VARCHAR2 AS
  N1 NUMBER;
  VINI NUMBER;
  VFIM NUMBER;
BEGIN
  VINI := NINI * 1000000;
  VFIM := NFIM * 1000000;
  N1 := 0;
  FOR I IN(SELECT ID AS HD,
            ID_NOTA,
            ID_CLIENTE,
            ID_PRODUTO,
            VALOR1,
            VALOR2,
            VALOR3,
            VALOR4,
            VALOR5,
            CARACTER1,
            CARACTER2,
            CARACTER3,
            CARACTER4,
            CARACTER5,
            DATA1,
            DATA2
           FROM TABELA_HDD
           WHERE ID BETWEEN VINI AND VFIM)
  LOOP
    N1 := I.VALOR1;
  END LOOP;

  RETURN 'EXEC: ' || NEXEC || ' - INI: ' || VINI || ' - FIM: ' || VFIM;
END;
```

Script 4: Função que testa a execução da consulta SQL na tabela_hdd

Para chamar as funções que executam as consultas no banco de dados foram criados *scripts* executam a função dez vezes com parâmetros diferentes, assim selecionando linhas diferentes em cada execução e evitando assim o *cache* de banco de dados, esses *scripts* também

medem o tempo de cada execução de cada chamada e por fim gravam os resultados dos testes em um arquivo de texto. No script 5 é exibido o comando que executa o teste na tabela `tabela_hdd`, esse *script* foi salvo com o nome `testa_1kk_hdd.sql` e posteriormente chamado pela ferramenta SQLPLUS, cada uma das dez execuções seleciona 1.000.000 de linhas no banco de dados.

```

set timing on
column dt new_val X
select to_char(sysdate,'yyyymmddhh24mmss') dt from dual;

spool c:\dados\resultado_1kk_hdd&X..txt

select testa_tabela_hdd(1, 1, 2) from dual;
select testa_tabela_hdd(2, 2, 3) from dual;
select testa_tabela_hdd(3, 3, 4) from dual;
select testa_tabela_hdd(4, 4, 5) from dual;
select testa_tabela_hdd(5, 5, 6) from dual;
select testa_tabela_hdd(6, 6, 7) from dual;
select testa_tabela_hdd(7, 7, 8) from dual;
select testa_tabela_hdd(8, 8, 9) from dual;
select testa_tabela_hdd(9, 9, 10) from dual;
select testa_tabela_hdd(10, 10, 11) from dual;

spool off

```

Script 5: Script que executa dez vezes os testes na tabela `tabela_hdd` e grava os resultados

Foram criados scripts para consultar 1.000.000, 2.500.000, 5.000.000 e 10.000.000 de linhas em cada uma das três tabelas relacionadas ao teste.

Para aferição dos resultados foram analisadas duas medidas diferentes, a primeira foi o tempo de execução individual de cada chamada de função, que foi registrado pelo SQLPLUS e gravado em arquivo de texto, o registro de tempo no SQLPLUS é ativado pelo comando “*set timing on*” que é mostrado na *script* 5. A segunda e mais importante medida analisada foi tempo de execução que é gravado internamente pelo próprio SGBD, que fica na tabela `v$sqlarea`, essa tabela armazena várias estatísticas das últimas consultas executadas, para este estudo foram analisadas apenas as colunas com informações mais relevantes, as duas mais importantes são a *elapsed_time* que armazena em microssegundos o tempo decorrido na análise, execução e busca de todas as execuções de uma mesma consulta, e a coluna *executions* que como o próprio nome já diz armazena o total de vezes que a consulta foi executada, com essas duas informações é possível chegar no tempo médio de execução de cada consulta, também foram listadas algumas colunas que mostram a quantidade de buscas realizadas no banco de dados, o número de linhas processadas, o tempo de *CPU* gasto, as leituras de disco e o tempo de espera de entrada e saída de dados para o usuário. No *script* é listado a consulta *SQL* na tabela `v$sqlarea`.

```
select  round(elapsed_time / 1000000, 3) tempo_total ,
        executions,
        round( (elapsed_time / executions) / 1000000, 3) as avg_ela_time,
        fetches,
        rows_processed,
        cpu_time,
        disk_reads,
        user_io_wait_time
from v$sqlarea v
where sql_text like '%TABELA\_HDD%' escape '\;
```

Script 6: Consulta na tabela v\$sqlarea

4 RESULTADOS E DISCUSSÃO

Este Capítulo apresenta os resultados obtidos através dos testes realizados, foram feitos testes de consultas *SQL* com quatro número de linhas: um milhão de linhas, dois milhões e quinhentas mil linhas, cinco milhões de linhas e dez milhões. As consultas foram executadas e repetidas dez vezes em cada uma das tecnologias analisadas neste estudo.

4.1 APRESENTAÇÃO DOS RESULTADOS

Os resultados adiante estão separados pelo número de linhas consultadas, e em ordem crescente, ou seja, iniciando pela consulta com um milhão de linhas e finalizando com a de dez milhões de linhas. Nos gráficos a seguir cada tecnologia estudada será representada da seguinte maneira, banco de dados em disco rígido como *HDD*, banco de dados em *SSD* e banco de dados em memória com *MEM*. Para cada montante de linhas são exibidos dois gráficos, um representando os tempos de execução de cada consulta e outro o tempo médio de execução em cada ambiente, quanto menor o tempo, melhor foi o desempenho.

No Gráfico 1 são exibidos os tempos individuais de execução de cada consulta com 1.000.000 de linhas obtidos através da ferramenta *SQLPLUS*.

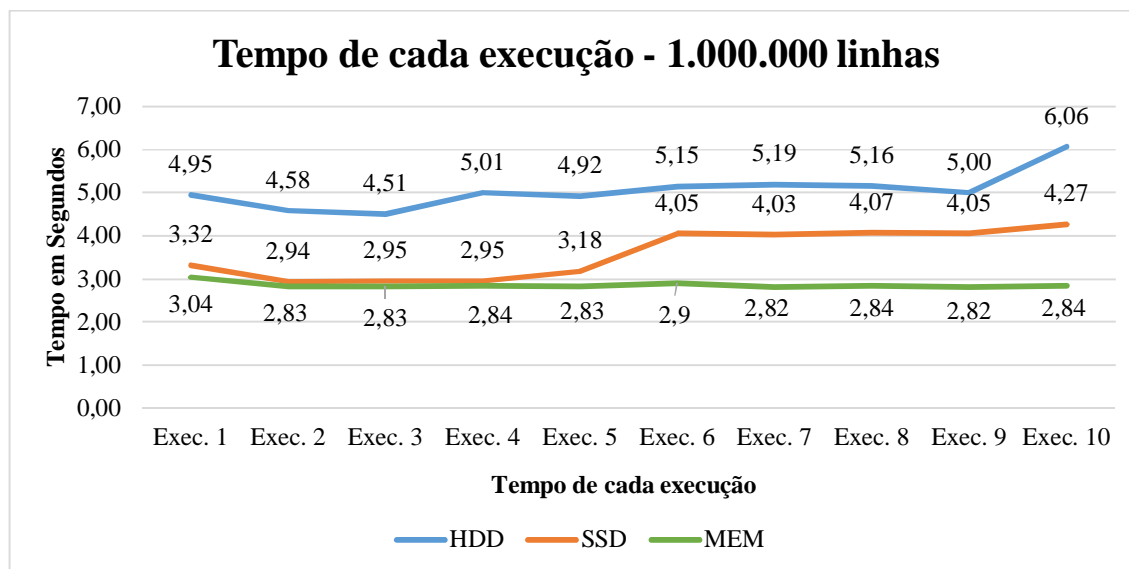


Gráfico 1 - Resultado de cada consulta com 1.000.000 de linhas.

No gráfico 2 são exibidos os tempos médios de execução da consulta com 1.000.000 de linhas obtidos por meio de consulta à tabela *v\$sqlarea*.

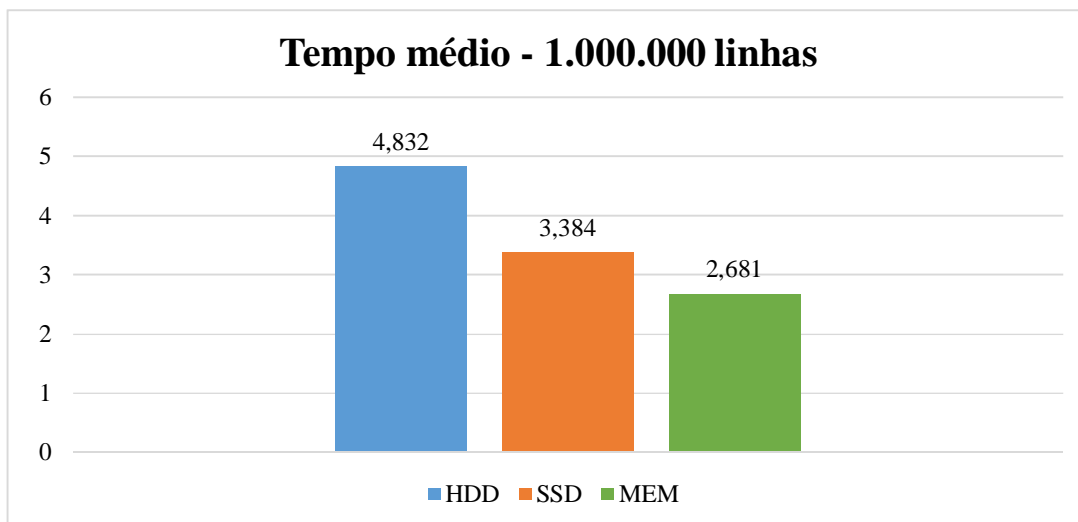


Gráfico 2 - Tempo médio da consulta com 1.000.000 de linhas.

Nos testes com 1.000.000 de linhas a consulta em disco rígido (*HDD*) comprovou que este ambiente não é o mais eficiente no quesito velocidade e ficou bem para trás e relação às demais, a consulta *SSD* teve significativas variações de tempo de execução, e na média das execuções apresentou resultados 30% melhores que o *HDD*, já a consulta em banco de dados em memória praticamente não teve variações e seus resultados foram 45% melhores que o modelo tradicional (*HDD*).

No gráfico 3 são exibidos os tempos individuais de execução de cada consulta com 2.500.000 de linhas obtidos através da ferramenta SQLPLUS.

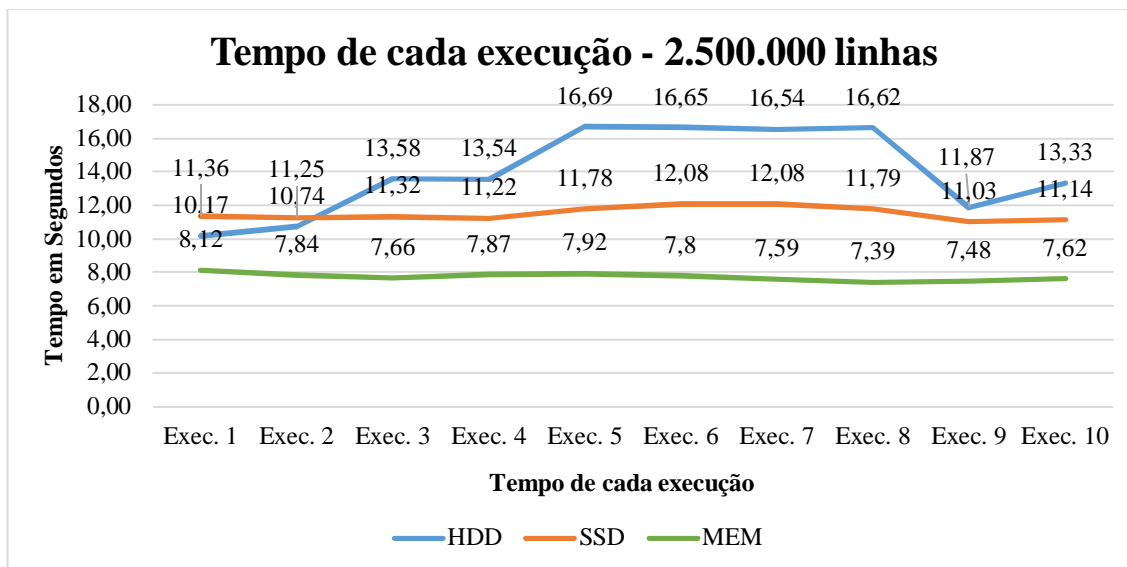


Gráfico 3 - Resultado de cada consulta com 2.500.000 linhas.

No gráfico 4 são exibidos os tempos médios de execução da consulta com 2.500.000 de linhas obtidos por meio de consulta à tabela v\$sqlarea.

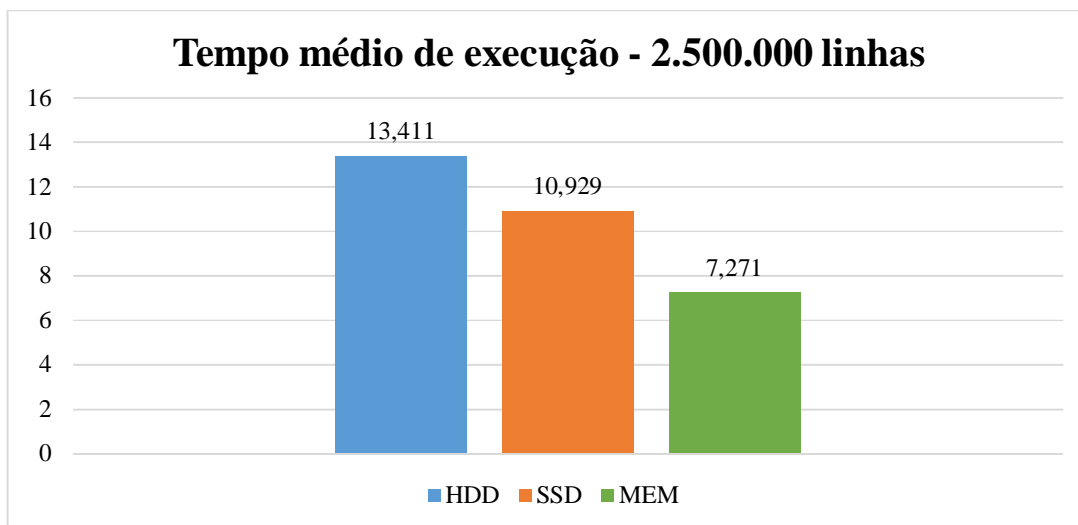


Gráfico 4 - Tempo médio da consulta com 2.500.000 de linhas

Nos testes com 2.500.000 de linhas a consulta em disco rígido (*HDD*) novamente foi a menos ágil, a consulta *SSD* teve poucas variações de tempo de execução, e na média das execuções apresentou resultados 19% melhores que o *HDD*, já a consulta em banco de dados em memória novamente não teve variações e seus resultados foram 56% melhores que o modelo tradicional (*HDD*).

No gráfico 5 são exibidos os tempos individuais de execução de cada consulta com 5.000.000 de linhas obtidos através da ferramenta SQLPLUS.

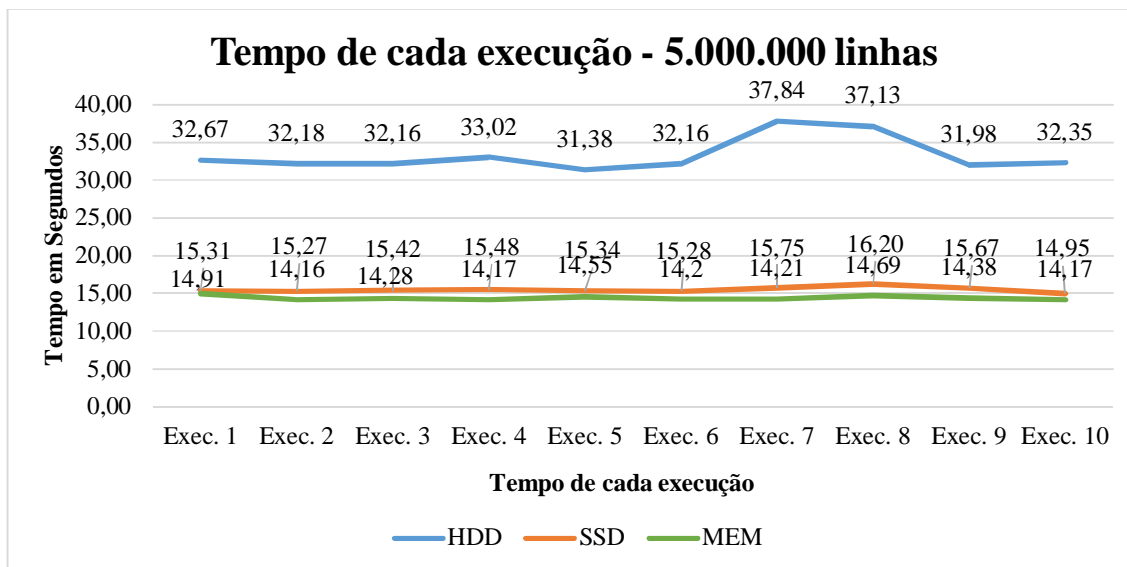


Gráfico 5 - Resultado de cada consulta com 5.000.000 linhas

No Gráfico 6 são exibidos os tempos médios de execução da consulta com 5.000.000 de linhas obtidos por meio de consulta à tabela v\$sqlarea.

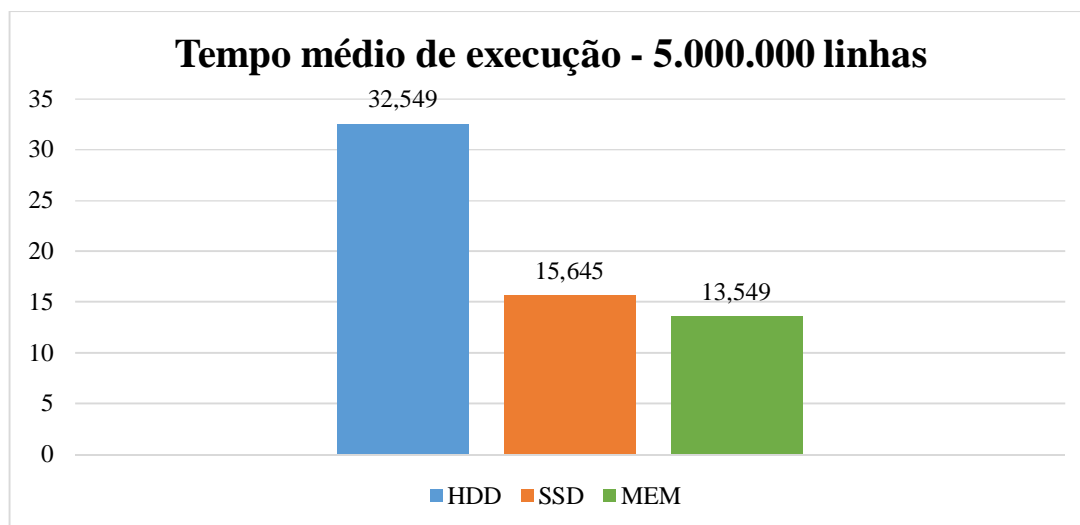


Gráfico 6 - Tempo médio da consulta com 5.000.000 de linhas

Nos testes com 5.000.000 de linhas a consulta em disco rígido (*HDD*) manteve seu desempenho relativamente baixo em relação às demais, a consulta *SSD* não teve variações de tempo de execução significativas, e na média das execuções apresentou resultados 52% melhores que o *HDD*, já a consulta em banco de dados em memória praticamente não teve variações e seus resultados foram 59% melhores que o modelo tradicional (*HDD*).

No Gráfico 7 são exibidos os tempos individuais de execução de cada consulta com 10.000.000 de linhas obtidos através da ferramenta SQLPLUS.

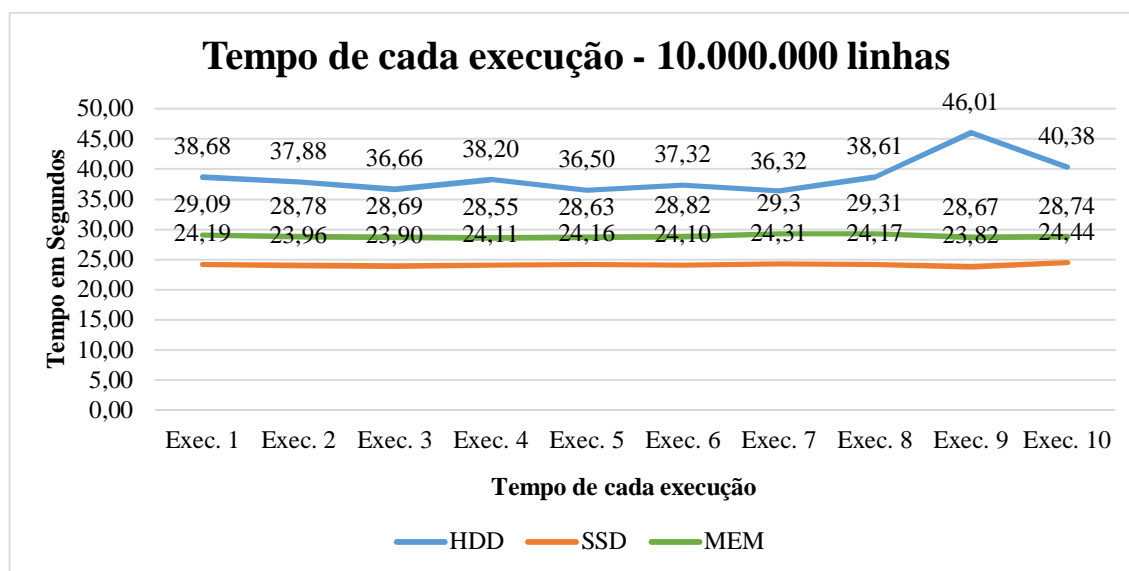


Gráfico 7 - Resultado de cada consulta com 10.000.000 linhas

No gráfico 8 são exibidos os tempos médios de execução da consulta com 10.000.000 de linhas obtidos por meio de consulta à tabela v\$sqlarea.

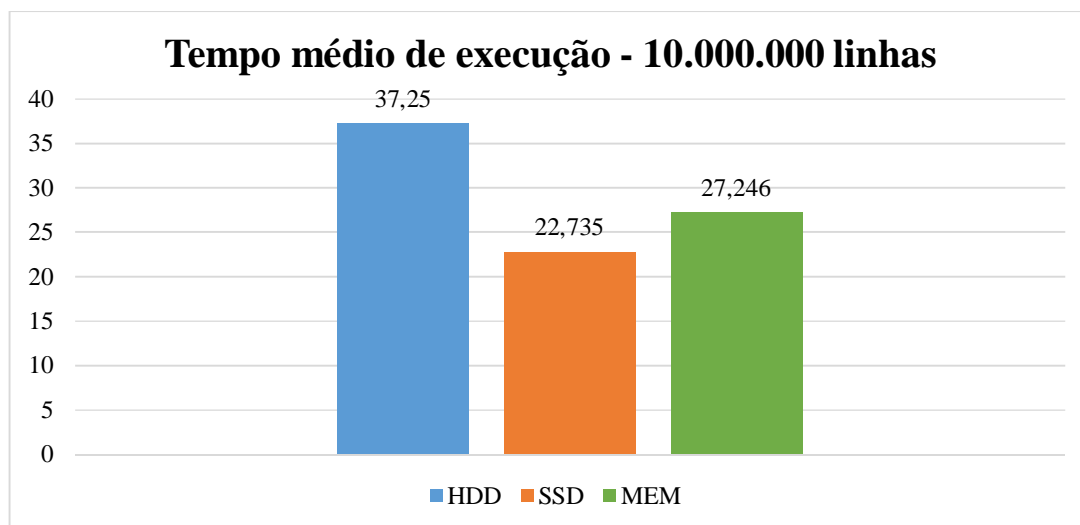


Gráfico 8 - Tempo médio da consulta com 10.000.000 de linhas.

Nos testes com 10.000.000 de linhas a consulta em disco rígido (*HDD*) não apresentou melhoria em relação as demais testes e novamente foi a mais lenta, a consulta *SSD* não teve variações de tempo de execução significativas, e na média das execuções apresentou resultados 49% melhores que o *HDD* o que lhe garantiu a primeira colocação neste teste, a consulta em banco de dados em memória praticamente não teve variações, porém seu desempenho surpreendeu negativamente e seus resultados foram apenas 27% mais rápidos do que o modelo tradicional (*HDD*) e ficou em segundo lugar neste teste.

4.2 DISCUSSÃO

O teste da consulta com 10.000.000 de linhas contrariou os demais resultados que apontaram o banco de dados em memória como o mais rápido, para não deixar dúvidas o teste foi repetido várias vezes, foi reiniciado o banco de dados, reiniciando o computador, até foi aumentado a memória do banco para o 10 gigabytes, mas os resultados foram os mesmos, não tendo nenhuma alteração. Um possível explicação para esse resultado é a análise dos dados coletados na tabela v\$sqlarea depois da execução dos testes, esses dados são apresentados no quadro 4, mostram o tempo de *CPU* e as leituras de disco usadas para a execução de cada teste, neles é possível notar que o tempo de *CPU* e leituras em disco são praticamente iguais no *HDD* e no *SSD*, já as consultas em memória praticamente não fazem leituras em disco, mas o uso de

CPU foi 20% maior que as outras tecnologias na consulta com 5.000.000 de linhas e 35% maior na consulta com 10.000.000 de linhas, que pode ter sido determinante para o baixo desempenho é baixa frequência do *CPU* utilizado um intel core i5 4200U com 2300 MHz.

Tecnologia	Linhas	Tempo de <i>CPU</i>	Leituras em Disco
<i>HDD</i>	5.000.000	102812500	2531921
<i>SSD</i>	5.000.000	110718750	2530821
MEM	5.000.000	135343750	0
<i>HDD</i>	10.000.000	182437500	2531921
<i>SSD</i>	10.000.000	181390625	2530821
MEM	10.000.000	271890625	1

Quadro 4: Recursos consumidos na execução das consultas

Ao final dos experimentos nota-se que na maioria dos testes o banco de dados em memória foi mais rápido, porém quando consultado um maior número de dados a tecnologia *SSD* se sobressai perante as demais opções analisadas, tendo um retorno de informações 16,55% mais rápido que o da segunda colocada. Os resultados encontrados são semelhantes aos de DIEGO MACÊDO (2013), PICHILIANI (2015), OLHAR DIGITAL (2015), DEVMEDIA (2016) e COSTA e PADOVANI (2016), que apontaram o *SSD* como uma tecnologia com melhor desempenho que o *HDD*, porém não realizaram comparações com o banco de dados *in-memory*.

5 CONCLUSÃO

Há várias tecnologias de armazenamento de informações disponíveis no mercado, optar pela que melhor atenda às necessidades das empresas é uma das dificuldades dos profissionais da área de banco de dados. A partir disso, percebeu-se a importância de estudos comparativos a respeito deste assunto, perante isso este trabalho teve como objetivo realizar um comparativo de tempo e desempenho entre algumas tecnologias de armazenamento de informações, *HDD*, *SSD* e *in-memory*.

Para tanto, foi necessário desenvolver e criar o cenário para a aplicação do experimento. Foram aplicados quatro diferentes montantes de informações, mais de dez vezes, nas três tecnologias, onde foi possível verificar que nas três primeiras quantidades de informações a tecnologia *in-memory* teve melhor desempenho, contudo, com uma quantidade maior de dados, a *SSD* se sobressaiu. No último teste, da consulta que seleciona 10.000.000 de linhas o desempenho do banco de dados em memória surpreendeu e ficou abaixo do desempenho do *SSD*, mesmo repetindo o teste várias vezes e até aumentando a memória disponível para o banco de dados em memória o resultado se manteve. Acredito que este resultado abaixo do esperado não se repita em um equipamento com melhores configurações.

Os resultados encontrados, em parte, conferem com os já existentes na literatura. Contudo, não há como indicar a melhor alternativa de armazenamento, ou seja, há a necessidade de sempre buscar avaliar a realidade e quantitativo dos dados que serão armazenados, para assim indicar o melhor custo-benefício. Vale ressaltar que a tecnologia *in-memory* no banco de dados Oracle é um recurso disponível apenas na versão *enterprise*, cujo valor é bem elevado.

Por fim, sugere-se para pesquisas futuras aplicar este experimento em outros quantitativos de informações, e realizar comparações com novas tecnologias, para confrontar os resultados e enriquecer essa importante área de estudos.

REFERÊNCIAS

CEAVI/UEDESC, 2013. **O que é um SSD?**. Disponível em: http://ceavi.udesc.br/arquivos/id_submenu/487/ssd.pdf

COSTA, Jober H. B.; PADOVANI Carlos R. P. **Comparação entre um HD Sata em um Solid State Drive (SSD) na instalação, manuseio e acesso de uma base de dados MySQL**. FATEC - Faculdade de Tecnologia Botucatu. São Paulo, 2016. Disponível em: <http://www.fatecbt.edu.br/ocs/index.php/VJTC/VJTC/paper/viewFile/706/911>

DATE, Christopher J. **Introdução a sistemas de banco de dados**. Tradução de Daniel Vieira. Rio de Janeiro: Elsevier Editora Ltda, 2003 – 8ª ed. 15º Impressão.

DEVMEDIA, 2016. **Testando o desempenho do HD com banco de dados**. Disponível em: <http://www.devmedia.com.br/testando-o-desempenho-do-hd-com-banco-de-dados/37429>

ELMASRI, Ramez; NAVATHE, Shamkant B. **Sistemas de banco de dados**. 6ª ed. São Paulo: Editora Pearson, 2011.

FERREIRA, João E.; ITALIANO, Isabel C.; TAKAI, Osvaldo K. **Introdução a banco de dados**. DCC-IME-USP, 2005. São Paulo.

GEHRKE, Johannes; RAMAKRISHNAN, Raghu. **Sistemas de Gerenciamento de Banco de Dados**. São Paulo: McGraw-Hill, 2008.

GEIER, Florian. **The differences between SSD and HDD technology regarding forensic investigations**. 67 p. Linnaeus University, 2015. Disponível em: <http://lnu.diva-portal.org/smash/get/diva2:824922/FULLTEXT01.pdf>

MACÊDO, D; **HD, SSD ou disco híbrido, qual o melhor para sua empresa?**. Publicado em 2013. Disponível em: <http://www.diegomacedo.com.br/hd-ssd-ou-disco-hibrido-qual-o-melhor-para-sua-empresa/>

MATSUMOTO, Cristina Yoshie. **Maringá Management: Revista de Ciências Empresariais**. v. 3, n.1 - p.45-55, jan./jun. 2006.

OLIVEIRA, Felipe M. L de. **Arquitetura de Solid State Driver (SSD)**. SEMAC, 2014. São Paulo. Disponível em: https://www.semac.sjrp.unesp.br/~aleardo/cursos/arqcomp/Semin_SSD.pdf

ORACLE, 2015. **Oracle Database 12c: In-Memory (Parte I)**. Disponível em: <http://www.oracle.com/technetwork/pt/articles/database-performance/oracle-database-12c-in-memory-2595527-ptb.html>

PICHILIANI, Mauro. **Comparando o desempenho do disco no SGBD em HD, SSD, Ramdrive e Cloud.** Publicado em 2015. Disponível em: <<http://pt.slideshare.net/MauroPichiliani/comparando-o-desempenho-do-disco-no-sgbd-hd-sdd-ramdrive-e-cloud>>

PROGRESS, 2016. **O que é a computação In-Memory?**. Disponível em: <http://progresslatamblog.com/2016/04/11/o-que-e-a-computacao-in-memory/>

VINHAS, Lúvia. Fundamentos de bancos de dados. **Earth System Science Centre.** 2016. Disponível em: <<http://ess.inpe.br/courses/lib/exe/>>

TECHTUDO, 2013. **HD, SSD ou disco híbrido: veja as diferenças e o que levar em conta ao comprar.** Disponível em: <<http://www.techtudo.com.br/noticias/noticia/2013/10/o-que-levar-em-conta-na-hora-de-comprar-um-ssd-ou-disco-hibrido.html>>