

SISTEMAS GERENCIAIS DE BANCO DE DADOS: UM ESTUDO DO MYSQL

Marcello Toledo da Silva

Especialista em Banco de Dados e Gestor de Tecnologia da Informação pela Claretiano, São Paulo, Brasil.

David Luiz de Souza Cavalcanti

Graduando em Sistema de Informação pela FALS, Praia Grande, São Paulo, Brasil.

RESUMO: O tema do presente trabalho é SISTEMAS GERENCIAIS DE BANCO DE DADOS: UM ESTUDO DO MYSQL, cujo objetivo é apresentar as vantagens e desvantagens dele, explicar qual banco de dados é mais recomendado para cada tipo necessidade. Afinal um dos grandes problemas que as empresas que utilizam banco de dados enfrentam, é a lentidão ao inserir e recuperar dados de seus sistemas gerenciadores. O mercado possui diversos sistemas de gerenciamento de banco de dados, onde a maioria são instalados e configurados com seus parâmetros. Encontramos no MySQL um padrão que atende melhor a todos.

Palavras-chave: Banco de Dados. MySQL. Vantagens e Desvantagens. Sistemas Gerenciais.

ABSTRACT:The theme of this work is DATABASE MANAGEMENT SYSTEMS: A STUDY OF MYSQL, whose purpose is to present the advantages and disadvantages of it, explain which database is most recommended for each type need. After all one of the big problems that companies that use database face, is the slowness when inserting and recovering data of their systems managers. The market has several database management systems, where most are installed and configured with their parameters. We find in MySQL a standard that best suits everyone.

Keywords: Database. MySQL. Advantages and Disadvantages. Management Systems.

INTRODUÇÃO

Segundo Almeida (2012), um BDD consiste em uma relação de “nós” (O termo é utilizado a fim de enfatizar a distribuição física dos sistemas), onde cada um pode participar de transações entre si, independentemente da quantidade de processos. Sendo o principal motivo de preocupações e dificuldades, a distribuição de dados.

Nos Sistemas de Bancos de Dados centralizados as informações estão localizadas em um único local, já nos distribuídos elas residem em diversos lugares.

Os tamanhos e funções dos processadores do sistema distribuído podem variar incluir microcomputadores, estações de trabalho e sistemas computacionais de uso

geral. Normalmente, dependendo do contexto no qual sejam mencionados, os processadores podem ser chamados de “nós”.

Em relação as vantagens, o gerenciamento de dados distribuídos possui níveis diferentes de transparência (distribuição ou rede, replicação e fragmentação). Proporciona também uma expansão mais fácil, melhor desempenho e melhoria na disponibilidade e confiabilidade.

Contudo, possui algumas falhas adicionais, como por exemplo: perda de comunicação e o particionamento de rede.

A tecnologia tem avançado constantemente, com isso é necessário que os processos tenham alta disponibilidade e segurança, sejam de uma empresa pequena, média ou de grande porte.

A utilização de um BDD (Banco de Dados Distribuídos), se tornar mais eficiente dependendo da sua implementação, pois não contará apenas com uma configuração centralizada. Será desenhado com distribuições focadas em locais estratégicos, dessa maneira o sistema fornecerá mais segurança e agilidade, pois mesmo quando sofrer qualquer tipo de falha, somente ele deverá ser tratado, enquanto os outros poderão continuar a trabalhar normalmente.

Nesse contexto, o artigo presente tem como objetivo analisar as vantagens e desvantagens de um sistema de banco de dados distribuídos com o foco no MySQL, apresentando seu desempenho, por meio da apresentação dos tipos de sistemas de gerenciamento de banco de dados interligados fisicamente (Banco de dados distribuídos) e sua proteção, junto a sua equidade e benefícios; da explanação da segurança do banco de dados e suas respectivas fases de aplicação; e da demonstração e detecção das falhas a fim de manter a rapidez em seu funcionamento.

Para confirmar a questão levantada na problemática, que diz que a complexidade do banco de dados pode gerar problemas técnicos o que por sua vez pode influenciar diretamente na estabilidade/velocidade dos processos, interferindo assim em um melhor desempenho; foi feito uma pesquisa de campo com 64 pessoas onde 98,4% acredita que a complexidade dos problemas, influenciam num melhor desempenho; e 54,7% dos entrevistados creem que o que causa maior da interferência no desempenho do sistema é a velocidade, 37,5% acreditam que seja a estabilidade, e o restante considera que possa ser a praticidade e a segurança

Em relação aos procedimentos metodológicos do presente artigo, trata-se de uma pesquisa bibliográfica, onde os dados foram pesquisados em materiais já elaborados, como livros, legislações e conteúdo disponibilizado na internet, juntamente ao estudo de caso para observar a prática. Em relação à natureza, trata-se de uma pesquisa aplicada, já que objetiva gerar conhecimentos para aplicação prática dirigida à solução de problema específico. Quanto à forma de abordagem, trata-se de uma pesquisa qualitativa, pois não requer o uso de métodos e técnicas estatísticas e o ambiente natural é a fonte direta para coleta de dados. Em relação aos objetivos, trata-se de uma pesquisa exploratória, pois visa o aprimoramento de ideias através de dados bibliográficos e de análise de exemplos com experiências práticas.

BANCO DE DADOS

De acordo com Monteiro (2004), atualmente, existe diversas maneiras de se desenvolver um software, programadores que pretendem criar um protótipo rápido, pulam a fase de planejamento e projeto e vão direto para as ferramentas de desenvolvimento. Os fatores importantes levados em consideração são o tempo para desenvolver o projeto e o custo, nesse caso o banco de dados (BD) é uma boa opção.

Segundo Alves (2004) antes de se apreender o que é BD, é necessário entender a diferença entre informação e dado. De acordo com o site Expert Digital (2017), dados são números grossos que, são de valor restrito, já a informação é um dado que foi transformado em um contexto útil. Por exemplo: Dado - Venda de computadores, Informação - Relatório de vendas por região e local.

Segundo Rezende (2006) um BD, é uma Coleção de dados que diz respeito a uma determinada organização, um conjunto de quatro componentes básicos: dados, hardware, software e usuários. Exemplo: lista telefônica, catálogo de CDs ou um sistema de controle de Recursos Humanos de uma empresa. Já um sistema de gerenciamento de banco de dados (SGBD) é um programa que possui recursos capazes de manusear as informações do BD e interagir com o usuário.

1.1 SISTEMAS GERENCIADORES DE BANCO DE DADOS

Segundo o site Dicas de Programador (2013), o que é feito em um BD é passado pelo SGBD, que fica responsável por salvar os dados no HD, manter em memória os

mais acessados, disponibilizar uma interface¹ para que programas e usuários externos acessem o BD, controlar o acesso a informações, manter cópias para recuperação de uma possível falha, garantir transações no banco de dados.

De acordo com Monteiro (2004), para o sistema operacional *Microsoft*, o *Access* pode ser considerado um sistema de gerenciamento de banco de dados para usuários finais e pequenas tarefas. Para usuários de *Unix* e *Linux*, existem diversas soluções pagas e públicas, como o *PostgreSQL*, *MySQL*, entre outros.

Funções

De acordo com Almeida (2012), as funções de um SGBD são: segurança, concorrência e propriedades AICD.

Segurança

O SGBD cria um sistema de proteção onde só é permitida entrada de usuários com autorização. Existem algumas regras que determinam quais os utilizadores que podem ter acesso, e a que ficheiros² podem assentir e quais operações podem efetuar (ler, adicionar, atualizar, apagar, etc). Alguns procedimentos permitem realizar cópias de segurança e recuperação de dados em caso de falhas, para assegurar a integridade deles. Na segurança física a informação deve estar bem guardada, na lógica ela é protegida através de métodos de segurança. E na integridade o SGBD garante a verificação das restrições, para manter os dados sempre válidos, diminuindo a difusão e elevando ao máximo sua consistência.

Concorrência

De acordo Schimiguel (2014), nas bases de dados multi-utilizadores, ainda pode haver uma tentativa de entrada. Por isso é importante ter conhecimento nas propriedades AICD, o SGBD providencia um mecanismo que assegura que a base seja atualizada corretamente, essa invenção é conhecida como mecanismo de controle de concorrência.

¹ A interface é o meio pelo qual o usuário se comunica com o sistema para realizar tarefas.

² Ficheiro é um recurso para armazenamento de informação, que está disponível a um programa de computador e é normalmente baseado em algum tipo de armazenamento durável.

Exemplos: Backup: Cópia de segurança, que permite a recuperação de dados importantes em caso de falha do disco rígido. *Inside Backup*, que é a informação guardada dentro da empresa e o *Outside Backup* onde ela é guardada fora.

Propriedades AICD

De acordo com Santos (2018) um SGBD deve manter 4 propriedades denominadas ACID (atomicidade, consistência, isolamento e durabilidade) para garantir a integridade dos dados. Elas asseguram que um BD irá continuar consistente após a execução de uma operação.

A Atomicidade é a característica que trata da transação de forma atômica³, ela programa o “tudo ou nada” em um banco de dados, onde todas as alterações têm que ser executadas com sucesso, se não nenhuma delas é realizada.

Já a Consistência é a propriedade que certifica que todas as mudanças serão realizadas no BD se forem consistentes, o ato só pode ser efetuado se os dados forem válidos, caso adverso será cancelado.

O Isolamento é a particularidade responsável por garantir que cada ligação no BD seja executada solitariamente. Ele garante que uma transação não irá interferir na outra, se acontecer de duas A e B serem realizadas no banco de dados por usuários distintos ao mesmo tempo, o isolamento garante que A será executada antes de B ou vice-versa.

A Durabilidade garante que toda operação bem ocorrida não seja perdida. Para o caso de uma ser executada com sucesso, até o fim de suas modificações serão armazenadas no BD e não serão perdidas mesmo que aconteça alguma falha no aparelhamento.

TIPOS DE GERENCIAMENTO DE BANCO DE DADOS

Segundo Rezende (2006) alguns tipos de SGBD são os seguintes:

³ Todas as instruções dentro de uma transação são tratadas de forma indivisível, como um átomo.

BM Informix

Que de acordo com o site IBM (2018) O *BM Informix* é um servidor de BD, veloz e escalável que comanda bancos de dados tradicionais, de objeto e dimensionais. Sua área de cobertura e seus recursos de autogerenciamento são incididos para soluções de gerenciamento de dados integradas. Seu servidor é executado nos sistemas operacionais *UNIX, Linux, Mac OS X e Windows*.

Suas publicações contêm as ferramentas de cliente. A *IBM Informix Client Software Development Kit (Client SDK)* inclui as interfaces de programação de aplicativos (APIs) para desenvolver aplicativos e fornecer conectividade de cliente.

O *IBM OpenAdmin Tool (OAT) for Informix* é um aplicativo da web para conduzir e considerar o desempenho de servidores do *Informix*.

O *IBM Informix DataBlade Developers Kit (DBDK)* contém instrumentos para desenvolver e enrolar módulos *DataBlade*, que são pacotes de software que abrem a funcionalidade do servidor de bancos de dados. Algumas edições do *Informix* incluem funcionalidade extra, como ferramentas de *warehousing*⁴ ou recursos de nuvem.

PostgreSQL

Segundo o Portal 4linux (2018), ele é um BD livre e o mais avançado mundo. É um banco de dados para especificação ANSI-SQL por ser adepto a esse padrão. Sua base de algoritmo veio do *Igres*, porém também derivam *Sybase e Microsoft SQL Server*. É totalmente comunitário, entretanto empresas como *EnterpriseDB, Fujitsu, Apple, Red Hat e VMWare*, participam de projetos com apoios, no formato de sociedade de desenvolvedores, empregando programadores que trabalham para a comunidade.

Firebird

De acordo com o site *Firebirdnews* (2010), o *Firebird* é um SGBD que tem o código aberto e não possui licença dupla, é totalmente gratuito. Seus principais recursos são: Suporte total a *Stored Procedures e Triggers*, Transações compatíveis com ACID, *Integridade Referencial, Multi Generational Architecture*, consome poucos

⁴ *Data Warehouse* é uma estrutura que armazena dados analíticos, destinados a gerencia de processos e tomada de decisões. Têm como característica não usar as regras de normalização necessárias às estruturas de banco de dados operacionais, aos sistemas de gestão (ERP) e relacionamento com o cliente (CRM)

recursos de processamento, Linguagem nativa para *Stored Procedures e Triggers* (PSQL), suporte para Funções Externas (UDFs). Praticamente não necessita de DBAs⁵ especializados, quase nenhuma configuração – assim que instala e já pode começar a usar, existe uma grande comunidade para se obter suporte gratuito, diversas formas de acesso ao banco de dados: nativo/API, *dbExpress*, *ODBC*, *OLEDB*, *.Net provider*, *JDBC nativo tipo 4*, *Python module*, *PHP*, *Perl*, etc. Suporte nativo para os maiores sistemas operacionais, incluindo o *Windows*, *Linux*, *Solaris*, *MacOS*.

HSQLDB

Segundo Lozano (2008) O HSQLDB é um BD criado totalmente em Java, e pode operar embutido em uma aplicação ou como um servidor de rede independente. Suporta dialeto SQL, incluindo *triggers*, integridade referencial, *outer joins*, visões, transações, campos BLOB, *schemas*, *roles* e consultas correlatas. Seus projetos vão desde servidores de aplicação J2EE⁶ como o *JBoss*, até ferramentas de desenvolvimento desktop como o *iReport*.⁷ A versão 2.0 do *OpenOffice*, é a popular suíte de escritórios que se firmou como opção ao Office da Microsoft, mesmo não sendo uma aplicação escrita em Java, inclui o HSQLDB como servidor de banco de dados embutido na aplicação *OOo Base* (que fornece funcionalidade similar ao *Access da Microsoft*).

DB2

De acordo com o Portal Gsti (2018), O IBM DB2 é um SGBD lançado pela IBM. Existem diversas versões do banco de dados DB2 que rodam desde em um simples computador de mão, até em potentes mainframes⁸ e trabalham com servidores baseados em sistemas *Unix*, *Windows* ou *Linux*.

⁵ Profissionais de T.I responsáveis pela operação em curso de banco de dados de uma organização e os aplicativos que acessam os bancos de dados.

⁶ A plataforma *Java J2EE (Java 2 Enterprise Edition)* surgiu com o objetivo de padronizar e simplificar a criação de aplicações empresariais. Para isso, propõe um modelo onde componentes J2EE (páginas *JSP*, *Servlets*, *EJB's*, etc) escritos pelos usuários da plataforma, podem fazer uso de serviços providos por esta, os quais simplificam sua implementação e possibilitam maior foco no negócio.

⁷ O *JasperReports* é uma ferramenta de relatório Java de software livre que pode gravar em vários destinos, como: *screen*, *printer*, *PDF*, *HTML*, *Microsoft Excel*, *RTF*, *ODT*, valores separados por vírgula ou arquivos XML

⁸ Um mainframe é um computador de grande porte, dedicado normalmente ao processamento de um volume grande de informações. Os mainframes são capazes de oferecer serviços de processamento a milhares de usuários através de milhares de terminais conectados diretamente ou através de uma rede.

MySQL

Segundo o portal Tectudo (2012), o MySQL é um sistema gerenciador de BD relacional que possui código aberto, onde a maioria das aplicações gratuitas servem para gerir suas bases de dados. Utiliza a linguagem SQL que é a linguagem mais popular para inserir, acessar e gerenciar o conteúdo armazenado num banco de dados. Nas aplicações web abertas e gratuitas, o conjunto mais usado é o LAMP, um acrônimo para *Linux, Apache, MySQL e Perl/PHP/Python*. São incluídas um sistema operacional, um servidor web, um sistema gerenciador de banco de dados e uma linguagem de programação. O *MySQL* é um dos componentes centrais da maioria das aplicações públicas da Internet.

Oracle

De acordo com o portal CriarWeb (2004), Oracle é uma ferramenta cliente/servidor vendida a nível mundial para a gestão de Bases de Dados. Em seu desenvolvimento, como é um sistema muito caro não está tão espalhado como outras bases de dados, por exemplo, *Access, MySQL, SQL Server*, etc.

SQL-Server

Segundo o site Msdn Microsoft (2018), o *Microsoft SQL Server* é um SGBD de análise de BD para soluções de comércio eletrônico, linha de negócios e data *warehouse*. A versão 2014 apresenta novos recursos na memória incorporados ao BD principal para OLTP (processamento de transações online) e data *warehousing*, completa com data na memória e BI existentes para apresentar a mais completa solução de banco de dados na memória do mercado. Ele também oferece novas soluções de recuperação de acidentes, backup e arquitetura híbrida com o *Windows Azure*, permite que os compradores usem suas habilidades existentes com as soluções locais que utilizem *datacenters* globais da *Microsoft*. Além disso, aproveita as vantagens de novos recursos do *Windows Server 2012 e Windows Server 2012 R2* para proporcionar uma escalabilidade sem paralelos para seu aplicativo em um ambiente físico ou virtual.

JADE

De acordo com Araújo (2008), o Jade é uma solução associada à linguagem e BD orientado a objetos. Seu ambiente é administrado pela ferramenta *JADE Object Manager*, cuida do armazenamento, gerência de transações, de cache, controle de

concorrência e ligação dinâmica deles. Ainda possui um banco de dados orientado chamado *JADE Object Database*, que grava definições de classes, código dos métodos e outros controles internos do ambiente. Suas aplicações construídas podem rodar nas plataformas *Windows* e *Linux*.

Microsoft Access

Segundo o site *Oficinadanet* (2010), o *Microsoft Access* é um Sistema de gerenciamento de BD da *Microsoft*, que está incluso no pacote do *Microsoft Office Professional*, que combina o *Microsoft Jet Database Engine* com uma interface gráfica do utilizador. Permite o desenvolvimento rápido de aplicações que abrangem a modelagem e estrutura de dados da interface a ser utilizada pelos usuários. Também os utiliza para armazenar em *Access/Jet*, *Microsoft SQL Server*, *Oracle*, ou qualquer recipiente compatível com ODBC.

COMPARATIVO

Com experiência no banco de dados citados acima, foi criado um comparativo entre eles, com seus pês e contras, como mostra na tabela a seguir:

Tabela 1 – Vantagens e Desvantagens

BANCO DE DADOS	PÓS	CONTRAS
<i>Bm informix</i>	Baixo custo de licenciamento; Baixo custo de mão de obra; Mantem a mesma performance que o concorrente, com menos <i>hardware</i> .	Falta de cursos para capacitação; Falta de material em português; Pouca facilidade de recursos no código SQL.
<i>PostgreSQL</i>	Boa organização; Suporte; Altamente extensível.	Complexo; Possui impossibilidade da criação de <i>outer joins</i> ; Não possui recuperação de falhas.
<i>Firebird</i>	Serviço de notificações; Armazenamento de dados em tempo real; Oferece uma ótima funcionalidade de testes	Migração difícil; Não suporta consultas refinadas nos dados por categorias; Não tem como classificar ou filtrar

<i>Db2</i>	Capacidade de organização Linguagens compatíveis Sistemas operacionais compatíveis	Pouco conhecido Pouco flexível Enfileiramento
<i>MySQL</i>	<i>Open source</i> Fácil Alto desempenho	Programa menos funções <i>Numberof</i> Vulnerabilidade Tamanho da base de dados
<i>Oracle</i>	Torna simples a criação e manutenção de data <i>warehouses</i> ; Protege o banco de ataques via web sem prejudicar a disponibilidade de informações e desempenho; Oferece uma separação segura dos dados dentro da base.	Instalação complexa; Exigência do x instalado; <i>Software</i> proprietário
<i>SQL Server</i>	Software de gestão empresarial -grade; Apoio de recuperação de dados; Fácil de usar	Custo; Compatibilidade limitada; Apenas servidor <i>Windows</i> .
<i>Microsoft Access</i>	Econômico Acesso integra-se bem com os outros membros do Microsoft Quando projetado corretamente, bancos de dados de acesso podem ser portados (geralmente com alguma dificuldade) do <i>SQL Server ou Oracle</i> .	Má concepção relacional; Tentativa de controlar excessivamente os formulários de banco de dados com controles personalizados para tudo; Problema com informações na tela de leitura.

Fonte: autor, 2018.

MYSQL

De acordo com Neves e Ruas (2005), o MySQL é um SGBD que suporta SQL, é *open source*⁹ e um dos mais utilizados, contém mais de 5 milhões de instalações ativas. Foi desenvolvido e disponibilizado pela empresa *MySQL AB Limited Company*, que vende um conjunto de serviços e produtos relacionados com a tecnologia MySQL. Seus principais clientes são : *Alcatel, AOL, The Associated Press, Caterpillar, Cox Communications, DaimlerChrysler, Dow Jones, EarthLink, Enercon, Ericsson, Google,*

⁹ *Open source* é um termo em inglês que significa código aberto. Isso diz respeito ao código-fonte de um software, que pode ser adaptado para diferentes fins.

Hoover's Online, Hewlett-Packard, Lucent, Lufthansa, NASA, Nortel, NYSE, Omaha Steaks, Sabre Holdings, Siemens, Suzuki, Texas Instruments, Time Inc., UPS e US Census Bureau.

Segundo Axmark e Widenius (2017) o MySQL é um SGBD relacional¹⁰ onde registra dados em listas separadas em vez de colocar todos apenas em um local. Logo proporciona velocidade e flexibilidade. A parte SQL é uma linguagem comum, usada para acessar banco de dados, ele é *Open Source*, portanto é possível qualquer pessoa fazer o *download*, sem pagar nada, usar e modificar o programa.

Ainda de acordo com Axmark e Widenius (2017) ele é um sistema cliente/servidor, que representa um servidor SQL multitarefa, feito para suportar diferentes acessos, diversos programas, clientes, bibliotecas, ferramentas administrativas e interfaces de programação além de possuir um conjunto de recursos práticos, desenvolvidos com a cooperação dos usuários, ser bem rápido, confiável, e fácil de usar. Foi desenvolvido para suportar bancos de dados grandes de modo veloz. Embora esteja em constante desenvolvimento, atualmente esse servidor oferece um rico e vantajoso conjunto de funções. O MySQL é altamente adaptável para acessar banco de dados na Internet graças a sua conectividade, velocidade e segurança.

HISTÓRIA

Segundo o site Prezi (2015) o MySQL teve origem na década de 90 quando David Axmark, Allan Larsson e Michael “Monty” Widenius, necessitaram de uma interface SQL compatível com as rotinas ISAM que utilizavam em suas aplicações e tabelas. De acordo com os autores Axmark e Widenius (2017), a intenção inicial era usar o MySQL, para conectar as listas utilizando rápidas rotinas de baixo nível, porém depois de algum tempo perceberam que o MySQL não era rápido e nem flexível para atender as demandas de necessidade. Isto derivou uma nova interface SQL para o BD, mas com praticamente a mesma Interface API¹¹ do MySQL. Ela foi escolhida para

¹⁰ *Software* que controla o armazenamento, recuperação, exclusão, segurança e integridade dos dados em um banco de dados

¹¹ Application Programming Interface. No português “Interface de Programação de Aplicações”. Elas são uma forma de integrar sistemas, possibilitando benefícios como a segurança dos dados, facilidade no intercâmbio entre informações com diferentes linguagens de programação e a monetização de acessos.

facilitar a portabilidade para códigos de terceiros que era escrito para ser levado facilmente para uso com o MySQL.

PRINCIPAIS CARACTERÍSTICAS

Ainda segundo o artigo Axmark e Widenius (2017) as principais características do SGBD MySQL são:

- Portabilidade: Ele é testado com uma ampla faixa de compiladores diferentes, funciona em diversas plataformas, utiliza o *GNU*¹², *Automake*, *Autoconf*, e *Libtool* para portabilidade. *APIs para C, C++, Eiffel, Java, Perl, PHP, Python, Ruby e Tcl*. Suporte total a *multi-threads*, além de fornecer estruturas de armazenamento transacional e não transferencial. É fácil de se adicionar outro mecanismo de armazenamento, e tem um sistema de alocação de memória rápido. Possui funções SQL que são implementadas por meio de uma biblioteca de classes otimizada. O código do MySQL foi testado com *Purify* (um detector comercial de falhas de memória) e também com o *Valgrind*, uma ferramenta GPL¹³.
- Segurança: Um sistema seguro que admite a verificação baseada em estações/máquinas. As Senhas são protegidas, pois toda a circulação de senhas é criptografada quando você se conecta ao servidor.
- Escalabilidade e limites: O Servidor MySQL contém 50.000.000 registros. São admitidos até 32 índices por tabela, cada um pode ser mesclado de 1 a 16 colunas, seu tamanho máximo é de 500 bytes (que pode ser alterado na compilação do MySQL). Um índice pode usar o prefixo de campo com um tipo CHAR¹⁴ ou VARCHAR.
- Conectividade: Os usuários podem se conectar ao servidor usando sockets TCP/IP, em qualquer plataforma. No *Windows*, os clientes podem se conectam usando *named pipes*. No *Unix*, eles podem se conectar usando arquivos *sockets*.
- Localização: O servidor apresenta avisos de erros ao usuário em várias línguas.

¹² GNU é um sistema operacional tipo Unix cujo objetivo desde sua concepção é oferecer um sistema operacional completo e totalmente composto por software livre - isto é, que respeita a liberdade dos usuários.

¹³ É a designação da licença de software para software idealizada por Richard Matthew Stallman em 1989, no âmbito do projeto GNU da Free Software Foundation

¹⁴ CHAR é texto de comprimento fixo. VARCHAR é texto de comprimento variável.

De acordo Teixeira (2013) uma outra característica desse sistema é a compatibilidade, pois o MySQL é compatível com a maioria dos sistemas operacionais existentes atualmente no mercado. Por ser desenvolvido em C e C++, faz com que sua acessibilidade entre os diversos sistemas operacionais seja mais fácil. Exemplo de alguns desses sistemas:

- *Windows*: Compatível com todas as versões;
- *Linux*: Sendo compatível com as principais versões, como *Fedora, Core, Debian, SuSE e RedHat*;
- *Unix*: Sendo compatível com as versões *Solaris, HP-UX, AIX, SCO*;
- *FreeBSD*;
- *Mac OS X Server*.

VISÃO GERAL DO MYSQL

De acordo com Axmark e Widenius (2017) A MySQL AB é a empresa dos fundadores e principais desenvolvedores do MySQL. Ela detém os direitos autorais do código fonte do MySQL, do logo e da marca. Seu intuito é se dedicar ao máximo ao MySQL e ao *Open Source*. Portanto conseguiu fazer com que o MySQL seja: o melhor e o mais utilizado banco de dados no mundo, acessível para todos, fácil de usar. Sempre melhorado, permanecendo rápido e seguro e Livre de erros.

MYSQL E SEU DESEMPENHO

De acordo com Neto (2011), um dos grandes problemas que as empresas que utilizam banco de dados enfrentam, é a lentidão ao inserir e recuperar dados de seus sistemas gerenciadores. O mercado possui diversos SGBD, onde a maioria são instalados e configurados com seus parâmetros. Porém os SGBDs relacionais¹⁵, através de suas estruturas, consolidaram-se sobre os tradicionais no desenvolvimento de aplicativos, pois muitos dos controles de manipulação de dados, manutenção e consultas foram delegados ao banco de dados, assim facilita e reduz o trabalho para os desenvolvedores.

¹⁵ **Um banco de dados relacional** é um mecanismo de armazenamento que permite a persistência de dados e opcionalmente implementar funcionalidades. Neste contexto, o objetivo deste artigo é apresentar uma visão geral de tecnologias de sistemas de gerenciamento de banco de dados relacionais (SGBDR) e explorar questões práticas aplicáveis ao seu uso em organizações modernas.

Ainda de acordo com Neto (2011), para o sistema possuir um bom desempenho é necessário tomar boas decisões durante seu projeto. Algumas considerações devem ser feitas ainda no início, como: o volume esperado de dados em cada relação do sistema, quais consultas serão realizadas com mais frequência, entre outros.

Fases do Sistema e sua Aplicabilidade

De acordo com Ribeiro (2010) uma fase que é subsequente de ajuste do sistema e se torna muito importante é a *Database Tuning*, pois ela possui o objetivo de maximizar o desempenho e a estabilidade. Nela, as informações coletadas no projeto, podem ser revisadas por meio da coleta de estatísticas sobre os padrões de uso da corporação. A utilização dos recursos pode ser monitorada para revelar gargalos, como a disputa pelos mesmos dados por vários usuários ou processos simultaneamente.

Assim, surge constantemente a necessidade de monitoramento e revisão do projeto físico do banco de dados, visto que os objetivos dessa fase são os seguintes:

- Preparar as aplicações para que sejam executadas mais rapidamente;
- Diminuir o tempo de resposta de consultas/transações;
- Melhorar o desempenho;
- A revisão das decisões de projeto na fase de sintonia.

Segundo o site IMasters (2017), um usuário iniciante de MySQL, provavelmente vai encontrar a necessidade de fazer algum *tuning* no servidor, para melhorar a performance. Alguns ajustes podem ser antecipados no momento da instalação do servidor, o principal é o: *innodb_buffer_pool_size*, que é à área da memória usada pelo InnoDB.

Segundo Ribeiro (2010) o *Tuning* envolve tratar diretamente as excessivas disputas por bloqueios, entre as transações, sobrecarga de registrar logs, a alocação de recursos, tais como discos, memória e processos para uma utilização mais eficiente, entre outros.

Configurações para Ajuste de Desempenho

Segundo Ribeiro (2010), para designar o tamanho padrão do buffer utilizado pelo servidor MySQL usa-se o comando:

```
"shell> MySQLd --verbose --help"
```

Ele produz uma lista com as opções do MySQL e as variáveis configuráveis do sistema. Para seu servidor, pode-se ver os valores atuais das variáveis do sistema através do comando:

```
"MySQL> SHOW VARIABLES"
```

Utilizando o comando acima, pode-se verificar algumas estatísticas e indicadores de status do servidor.

Caso aconteça de ter muitas conexões simultâneas, os problemas de troca de dados podem ocorrer, a menos que o MySQL estiver configurado para utilizar um mínimo de memória para cada conexão. Para isso, podem ser aproveitados os seguintes recursos:

```
shell> MySQLd_safe  
--key_buffer_size=512K  
--sort_buffer_size=100K  
--read_buffer_size=100K &
```

ou:

```
shell> MySQLd_safe --key_buffer_size=512K  
--sort_buffer_size=16K  
--table_cache=32 --read_buffer_size=8K  
--net_buffer_length=1K &
```

Caso seja preciso executar um *GROUP BY* ou *ORDER BY* em tabelas que são muito maiores do que a memória disponível, é preciso aumentar o valor de *RECORD RND BUFFER SIZE* para acelerar a leitura de registros após a operação de ordenação.

Para ver os efeitos de uma mudança de parâmetro, basta usar o comando:

```
shell> MySQLd --key_buffer_size=32M --verbose --help
```

Performance

De acordo com Ribeiro (2010), para ambientes com baixa concorrência de dados é recomendado o MyISAM¹⁶, pois nele não ocorre processo de escrita (*Updates/Inserts*) apenas de leitura (*Selects*), não é indicado para aplicações que precisam de uma maior granularidade de bloqueio de dados (comandos de escrita).

Algumas de suas características são:

- Tabela padrão no MySQL;
- Faz com que uma escrita bloqueie completamente o acesso a todos os dados armazenados na tabela que está sendo alterada;
- Manipulação flexível da coluna;
- Suporta índices e comprimidos;
- Pode ser convertido em tabelas;
- Possui um grande desempenho para leitura e escrita quando é baixa a concorrência;
- Os índices são armazenados em árvores binárias balanceadas;
- Não provê controle de transações e nem integridade referencial.

O Storage Engine Maria

Segundo Ribeiro (2010), uma versão com as características do MyISAM é o *storage engine* Maria a diferença, é que ele possui um suporte a recuperação de falhas e logs mais detalhados. Nele também, são suportadas tabelas não-transacionais, com gravação imediata em disco, que podem ser recuperadas em caso de problemas e com alterações rigidamente armazenadas no log.

A recuperação ocorre automaticamente, porem as operações que devem ser protegidas contra falhas deverão ser inseridas dentro de cláusulas *Lock Tables* e *Unlock Tables*. Ao ser executado, as transações serão com AICD, elas são atômicas por serem indivisíveis, consistentes pois devem obedecer às regras do banco de dados, isoladas e permanentes porque o resultado da transação deve ser sólido, mesmo em caso de falhas.

¹⁶ MyISAM era o mecanismo de armazenamento para as versões do sistema de gerenciamento de banco de dados MySQL anteriores à 5.5. Ele é baseado no antigo código ISAM mas possui muitas extensões (Wikipedia, 2018).

Para uma boa capacidade transacional, recomenda-se usar o InnoDB, um tipo de tabela transacional que passou a ser parte integrada das distribuições do MySQL. Possui suporte para transações ACID.

Algumas de suas características são:

- Suporta Foreign Keys e integridade referencial com implementação dos *Constraints SET NULL, SET DEFAULT, RESTRICT e CASCADE*;
- Possui um alto desempenho com grandes volumes de dados e um número elevado de concorrência entre leitura e escrita;
- Programa o Lock de registro;
- Possui uma facilidade de implementação dos níveis de isolamento;
- Ele não exige nenhum recurso específico, mas, em determinadas situações tem se mostrado extremamente rápido comparando com outros SGBDs transacionais.

O storage engine Falcon

De acordo com Ribeiro (2010), uma das novidades que a MySQL AB já apresentou é o *Falcon*, que possui uma arquitetura muito atraente, por ter recursos avançados com muita simplicidade, e pela necessidade de pouca manutenção por manifestar poucos problemas. Sua aplicação é recomendada também em arquiteturas habilitadas para suportar grande quantidade de memória, que se baseiam em sete componentes.

São eles:

- *USER TABLESPACES* onde contém os dados e índices atuais armazenados em um banco de dados do *storage engine*;
- *FALCON SERIAL LOG* que contém as modificações mais atuais
- *PAGE CACHE*: região de memória para o armazenamento de páginas de dados que estão sendo escritas ou lidas;
- *SYSTEM METADATA* que armazena informações internas e relevantes ao seu funcionamento;
- *RECORD CACHE* local de memória que armazena dados ainda não receberam um *commit* e continuam a ser manipulados;

- *SYSTEM CACHE* é a região de memória que contém informações do contexto das transações correntes, e sistemas de metadados;
- *WORKER THREADS* são *Threads* que movem os dados já escritos no log do *storage engine* para dentro do *page cache*, e dele para o disco. Além de gerenciar dados BLOB.

É muito importante, trabalhar em um banco de dados que possua compatibilidade com o padrão MySQL. Com o *Falcon*, todos os dados pertencentes a um BD são armazenados em um registro de *TABLESPACE* no diretório localizado no sistema de arquivos. Será utilizado para armazenar tabelas, independente de qual banco de dados pertence. Além de reutilizar completamente as páginas em branco que resultaram de dados apagados, por isso não terá necessidade de realizar reestruturações constantes de tabelas.

COMPATIBILIDADES AOS PADRÕES

De acordo com o Axmark e Widenius (2017), o servidor MySQL tem muitas expansões aos padrões SQL. Um de seus principais objetivos é trabalhar em acordo com o padrão SQL-99, mas sem sacrificar velocidade e confiança, sem receio de adicionar extensões ao SQL se é aumentada a usabilidade do MySQL Server continuasse a suportar bancos de dados transacionais e não transacionais para atender tanto o modo pesado na web quanto o uso de missão crítica, ele foi projetado inicialmente para trabalhar com bancos de dados de tamanho médio em sistemas computacionais pequenos. O design compacto desse servidor torna as direções plausíveis sem qualquer conflito na árvore fonte.

VELOCIDADE

De acordo com o portal Turbosite (2018), existem alguns comandos que podem ajudar na questão velocidade, deste banco de dados, como:

- *MYSQLREPORT*: Este comando, é utilizado para gerar relatórios sobre bancos de dados que estão próximos ao limite de sua capacidade de uso e estabelecer um patamar para operações e usos adicionais. O usuário também pode se ver obrigado a modificar arquivo de configuração (*my.cnf*) para registrar o log de consultas lentas, identificando as que estão processando de maneira lenta e as possíveis causas.

- **EXPLAIN:** Já esse comando deve ser aplicado, antes do script *SELECT* para que o MySQL retorne um registro contendo a análise do script, para a tabela consultada. Esse processo retorna algumas informações, como: *ID, SELECT_TYPE, TABLE, TYPE, KEY_LEN, REF*, significando que as informações que você vai querer para avaliar o desempenho do código são: *POSSIBLE_KEYS*: que mostra os índices disponíveis para utilização naquela consulta; *KEY*: exibe o índice escolhido pelo MySQL para a realização daquela consulta; *ROW*: É o número considerado de linhas percorridas para encontrar o resultado do *SELECT*; *EXTRA*: São sugestões de melhoria em índices das consultas, como: *DISTINCT, NOT EXIST, USING INDEX*, entre outras.
- **ANALYZE:** É o comando utilizado para gerar uma distribuição de chaves, para a tabela a ser empregada pelo otimizador de consultas do MySQL, para definir quais índices serão melhores se empregados em uma consulta.
- **OPTIMIZE:** É o comando usado quando existe grandes quantidades de inserções/exclusões em uma tabela, principalmente nos casos em que a tabela contém muitos campos do tipo *varchar, text* e *blob*, o disco será comprometido à várias fragmentações. Nesse tipo de situação, esse comando deve ser usado com frequência pois ele irá otimizar a movimentação de cabeça de leitura e gravação do disco durante a recuperação dos dados, concretizando a desfragmentação do disco, causada por campos de comprimento variável.
- **LOAD DATA INFILE:** Este comando é empregue quando é necessário a importação de dados para uma tabela, ele, no lugar do comando *Select*, irá proporcionar um desempenho 20 vezes melhor.
- **TRUNCATE:** Diferente do comando *DELETE* que apaga a primeira linha, o *Truncate* deleta todas de uma vez. Portanto ele é utilizado quando o usuário não está interessado no número de linhas suprimidas de uma tabela.

PERMISSÕES NO MYSQL

Ainda de acordo com o site Turbosite (2018) uma forma de melhorar o desempenho do sistema é criar usuários com permissões específicas, pois diminuem o *overhead* das verificações das autorizações em todos os comandos realizados no MySQL.

BENCHMARK é comando utilizado para compreender quanto tempo uma determinada função ou expressão MySQL está demorando, é utilizada a função MySQL embutida:

```
SELECT BENCHMARK (10000, (SELECT col1 FROM funcionarios LIMIT 1))
```

Recordando que, para passar consultas na função **BENCHMARK**, obrigatoriamente a *query* deve retornar 1 linha com 1 coluna.

PESQUISA DE CAMPO

Para confirmar a questão levantada na problemática, foi realizada uma pesquisa de campo para usuário de banco de dados, 64 pessoas responderam essa pesquisa que continha as seguintes perguntas:

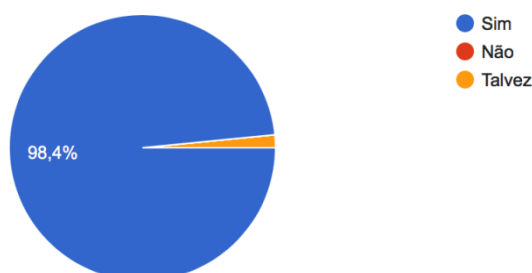
- 1) Você acredita que as complexidades dos problemas influenciam em um melhor desempenho?
- 2) O que você considera que causa maior interferência no desempenho do sistema?
 - A) velocidade
 - B) segurança
 - C) praticidade
 - D) estabilidade
 - E) não há interferência

Respondendo a primeira pergunta, 98,4 % acreditam que a complexidade dos problemas, influenciam um melhor desempenho, como é mostrado no gráfico abaixo:

Gráfico 1 – Pesquisa de campo

Você acredita que a complexidade dos problemas, influenciam em um melhor desempenho?

64 respostas



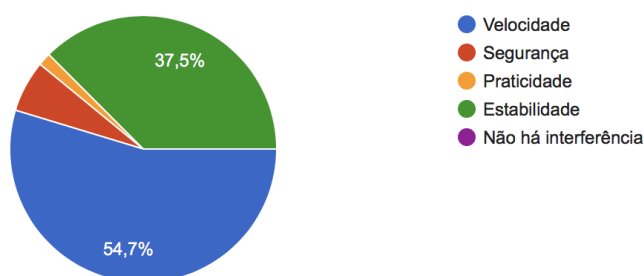
Fonte: autor, 2018.

Já na segunda, 54,7% creem que o que causa maior interferência no desempenho do sistema é a velocidade, 37,5% acreditam que seja a estabilidade, e o restante considera que possa ser a praticidade e a segurança, como mostra o gráfico a seguir.

Gráfico 2 – Pesquisa de campo

O que você considera que causa maior interferência no desempenho do sistema?

64 respostas



Fonte: autor, 2018.

CONCLUSÃO

Com este trabalho, concluo que o SGBD MySQL é recomendado para um grande público, desde os iniciantes aos profissionais com extrema expertise. Um banco de dados com um bom desempenho, facilidade de aprendizagem e programação que pode ser utilizado em qualquer tipo de aplicação.

Além de garantir a disponibilidade da rede, confirmando a hipótese 01. Este BD não fica para trás no quesito segurança e eficácia, mesmo com toda a acessibilidade e o fato de ser gratuito, o que confirma a hipótese 02. O MySQL disponibiliza diversos códigos para manter estável a velocidade do processo e garantir a rapidez, confirmando a hipótese 03.

Para quem tem dúvida sobre qual banco de dados utilizarem, no meu ponto de vista, o MySQL é a melhor recomendação, pois além de oferecer uma gama completa de produtos certificados, testados e homologados pela própria MySQL AB, possui treinamento, consultoria e apoio para fazer um bom projeto.

Por este motivo ele virou o mais popular banco de dados *open source* do mundo, pois possui uma ótima consistência, alta performance, confiabilidade e é fácil de usar. Atualmente ele é utilizado em mais de 6 milhões de instalações em todos os continentes. Funciona em plataformas como *Linux, Windows, HP-UX, AIX, Netware*, traz ao usuário uma maior flexibilidade e controle.

REFERENCIAL

ALMEIDA, Roniere. **O que é um sistema de banco de dados bem distribuído?** 2012. Disponível em: <<https://www.devmedia.com.br/o-que-e-um-banco-de-dados-distribuido/24762>>. Acesso em 22 de abr. de 2018.

ALVES, Willian Perereia. **Fundamentos de Banco de Dados**. 1. ed. São Paulo: Érica, 2004.

ARAÚJO, Marcos Antônio. **Bancos de dados orientados a objetos**. 2008. Disponível em: <<https://www.devmedia.com.br/artigo-da-sql-magazine-43-bancos-de-dados-orientados-a-objetos/7055>>. Acesso em: 28 mai. 2018.

AXMARK, David; WIDENIUS, Michael. **Manual de referência do MySQL 4.1**. 2017. Disponível em: <http://www.fpw.com.br/versao2017/web/my_sql.pdf>. Acesso em: 02 jun. 2018.

CRIARWEB. **O que é Oracle**. 2004. Disponível em: <<http://www.criarweb.com/artigos/230.php>>. Acesso em: 29 mai. 2018.

DICASDEPROGRAMADOR. **O que é um SGBD**. 2013. Disponível em: <<https://dicasdeprogramacao.com.br/o-que-e-um-sgbd/>>. Acesso em: 15 mai. 2018.

EXPERTDIGITAL. **Diferenças entre Dados e Informação**. 2017. Disponível em: <<https://expertdigital.net/diferenca-entre-dados-e-informacao/>>. Acesso em: 12 mai. 2018.

FIREBIRDNEWS. **Conheça o que é Firebird em 2 minutos**. 2010. Disponível em: <https://www.firebirdnews.org/docs/fb2min_ptbr.html>. Acesso em: 29 mai. 2018.

IBM. **Visão do Produto**. 2018. Disponível em: <https://www.ibm.com/support/knowledgecenter/pt-br/SSGU8G_12.1.0/com.ibm.po.doc/po.htm>. Acesso em: 25 mai. 2018

IMASTER. **MySQL Performance Tuning: Sempre configure o InnoDB Buffer Pool**. 2017. Disponível em: <<https://imasters.com.br/data/mysql-performance-tuning-sempre-configure-o-innodb-buffer-pool>>. Acesso 30 ago. 2018.

LOZANO, Fernando. **Artigo Java Magazine 30: O novo HSQLDB**. 2010. Disponível em: <<https://www.devmedia.com.br/artigo-java-magazine-30-o-novo-hsqldb/8757>>. Acesso: 29 mai. 2018.

MONTEIRO, Emiliano Soares. **Projeto de Sistemas e Banco De Dados**. 1. ed. Rio de Janeiro: Brasport, 2004.

MSDN.MICROSOFT. **Microsoft SQL Server**. 2018. Disponível em: <<https://msdn.microsoft.com/pt-br/library/bb545450.aspx>>. Acesso em: 28 mai. 2018.

NETO, Arilo Claudio Dias. **Bancos de Dados Relacionais**: Artigo Revista SQL Magazine 86. 2011. Disponível em: <<https://www.devmedia.com.br/bancos-de-dados-relacionais-artigo-revista-sql-magazine-86/20401>>. Acesso em: 01 jun. de 2018.

NEVES, Pedro M.C; RUAS, Ruy P.F. **O guia prático do MySQL**. 1. ed. Lisboa: Centro Atlântico, 2005.

OFICINADANET. **Access**. 2010. Disponível em: <<https://www.oficinadanet.com.br/access>>. Acesso em: 05 mai. 2018.

PREZI. **História do MySQL**. 2015. Disponível em: <https://prezi.com/_-kde_tbx1n5/historia-do-mysql/>. Acesso em: 30 mai. 2018.

PORTALGSTI. **O que é IBM DB2?** 2018. Disponível em: <<https://www.portalgsti.com.br/ibm-db2/sobre/>>. Acesso em: 27 mai. 2018.

REZENDE, Ricardo. **Conceitos Fundamentais de Banco de Dados**. 2006. Disponível em: <<https://www.devmedia.com.br/conceitos-fundamentais-de-banco-de-dados/1649>>. Acesso em: 19 mai. 2018.

RIBEIRO, Robson Afranio. **Performance do banco de dados MySQL**. 2010 Disponível em: <<https://www.devmedia.com.br/performance-do-banco-de-dados-mysql/18508>>. Acesso em: 30 ago. 2018.

SANTOS, Rodrigo de carvalho. **Estudo Comparativo dos Sistemas Gerenciadores de Bancos de Dados: Oracle, SQL Server e PostgreSQL**. 2018. Disponível em: <<http://www.unipac.br/site/bb/tcc/tcc-6cda526f17394bae9bbbed711b2188920.pdf>>. Acesso em: 25 mai. 2018.

SCHIMIGUEL, Juliano. **Gerenciamento de banco de dados: análise comparativa de SGBD'S**. 2014. Disponível em: <<https://www.devmedia.com.br/gerenciamento-de-banco-de-dados-analise-comparativa-de-sgbd/30788>>. Acesso em: 28 mai. 2018.

TEIXEIRA, José Ricardo. **Introdução ao MySQL**. 2013. Disponível em: <<https://www.devmedia.com.br/introducao-ao-mysql/27799>>. Acesso em: 07 ago. 2018.

TECHTUDO. **O que é e como usar MySQL**. 2012. Disponível em: <<http://www.techtudo.com.br/artigos/noticia/2012/04/o-que-e-e-como-usar-o-mysql.html>>. Acesso em: 25 mai. 2018.

TURBOSITE. **Dicas para otimizar seu MySQL**. 2018. Disponível em: <<https://www.turbosite.com.br/blog/dicas-para-otimizar-seu-mysql> >. Acesso em 20 ago. 2018.

4LINUX.**O que é PostgreSQL**. 2018. Disponível em: <<https://www.4linux.com.br/o-que-e-postgresql>>. Acesso em: 24 mai. 2018.
