

Capítulo 1

Bem-vindo ao MySQL

Olá, seja bem-vindo ao mundo do MySQL! A melhor maneira de conhecer uma ferramenta é por meio de uma leitura inicial sobre a origem, o propósito e os objetivos aos quais esta pretende atingir.

O primeiro capítulo abordará os conceitos iniciais do MySQL, comentando um pouco sobre sua história, onde a ferramenta está posicionada atualmente, suas principais características, recursos e compatibilidades, encerrando com citações dos principais e maiores casos de sucesso e sites relacionados sobre o assunto.

1.1 O que é o MySQL?

Em um mundo onde a globalização está cada vez mais presente, os processos cada vez mais automatizados e as barreiras de distância sendo quebradas pelo aumento da popularização da internet, a necessidade de armazenamento de dados e informações de cada mercado torna-se o primeiro passo para a migração de seu negócio para a internet. É nessa etapa que entra o MySQL, visando a suprir essa necessidade da melhor forma possível.

O MySQL é um servidor e gerenciador de banco de dados (SGBD) relacional, de licença dupla (sendo uma delas de software livre), projetado inicialmente para trabalhar com aplicações de pequeno e médio portes, mas hoje atendendo a aplicações de grande porte e com mais vantagens do que seus concorrentes. Possui todas as características que um banco de dados de grande porte precisa, sendo reconhecido por algumas entidades como o banco de dados open source com maior capacidade para concorrer com programas similares de código fechado, tais como SQL Server (da Microsoft) e Oracle.

1.1.1 História do MySQL

O MySQL teve origem quando os desenvolvedores David Axmark, Allan Larsson e Michael “Monty” Widenius, na década de 90, precisaram de uma interface SQL compatível com as rotinas ISAM que utilizavam em suas aplicações e tabelas. Em um primeiro momento, tentaram utilizar a API mSQL, contudo a API não era tão rápida quanto eles precisavam, pois utilizavam rotinas de baixo nível (mais rápidas que rotinas normais). Utilizando a API do mSQL, escreveram em C e C++ uma nova API que deu origem ao MySQL.

Com o ótimo resultado gerado por essa nova API, o MySQL começou a ser difundido e seus criadores fundaram a empresa responsável por sua manutenção, que é a MySQL AB.

A partir dessa fase, o MySQL tornou-se mais conhecido por suas características de rápido acesso e cada vez mais utilizado. Novas versões foram lançadas, contemplando novas necessidades e firmando, assim, sua posição no mercado. Sua mais recente versão é a 5.1, a qual conta com novos recursos, estabelecendo sua capacidade para competir com os bancos de dados privados de maior popularidade (SQL Server e Oracle).

1.1.2 Licença de uso

O MySQL é desenvolvido e distribuído por meio de duas licenças que irão depender do tipo de uso da ferramenta. Na maioria dos casos, seu uso é livre. Contudo, vale a pena conhecer as duas possíveis licenças do MySQL.

A primeira, software livre, é baseada nas cláusulas da GNU-GPL (General Public Licence), a qual estabelece o que se pode ou não fazer com a ferramenta e demais recursos. Além do programa, o seu código-fonte também é disponibilizado para que qualquer pessoa possa adaptá-lo às suas necessidades; contudo, todas essas situações serão tratadas e detalhadas na licença GNU-GPL.

Basicamente, a licença do tipo GNU-GPL baseia-se nos seguintes princípios:

- **Utilização:** Permite utilizar o software para qualquer propósito.
- **Distribuição:** Permite a livre distribuição do software entre todas as pessoas.
- **Didática:** Permite que seu funcionamento seja estudado a partir de seu código-fonte.

- **Colaboração:** Permite que seu código-fonte seja alterado para evoluir a ferramenta, desde que seu novo código-fonte continue sendo livre seguindo essa licença.

Para saber mais sobre essa licença, acesse o seu site oficial (em inglês):

<http://www.gnu.org>

A segunda licença é comercial. Para algumas situações sobre como embutir o MySQL dentro de suas aplicações comerciais (OEMs, ou seja, fora dos termos da GNU-GPL), obter suporte diferenciado, ou obter pacotes com mais ferramentas, essa licença pode ser mais apropriada. Maiores informações podem ser obtidas no site oficial da ferramenta:

<http://www.mysql.com/oem/licensing.html>

O melhor tipo de licença variará de caso para caso, dependendo da necessidade de treinamentos, suportes, entre outros fatores. Geralmente o MySQL pode ser utilizado gratuitamente. Para saber de qual tipo de licença a sua aplicação necessita, consulte ambos os endereços eletrônicos disponíveis nos parágrafos anteriores.

1.2 Utilizações recomendadas

Inicialmente, o MySQL foi projetado para trabalhar com aplicações de pequeno a médio porte, algo em torno de 100 milhões de registros por tabela, tendo como tamanho médio aproximadamente 100MB por tabela. Contudo, esses números eram os recomendados para as primeiras versões da ferramenta. Atualmente os limites e capacidades do MySQL ultrapassaram essas fronteiras inúmeras vezes. Uma revisão completa dos números e capacidades de armazenamento do MySQL será abordada no tópico 1.4.

Além de ser extremamente rápido, pelo fato de armazenar os dados em tabelas no modo ISAM (código de baixo nível), o MySQL é altamente confiável. Por possuir essas características, a ferramenta é indicada para uso em aplicações em todas as áreas de negócio, independentemente do tamanho de sua aplicação. Vale a pena ressaltar que a partir da versão 5, os códigos ISAM foram substituídos por uma nova versão, chamada de MyISAM, que provê melhor desempenho, entre outras melhorias.

Atualmente, tem sido o banco de dados open-source mais utilizado em aplicações intensas para a internet (24 horas por dia, 7 dias na semana), tais como soluções Web e lojas virtuais, pelo fato de essas aplicações demandarem rápido acesso para

a geração de páginas HTML, qualquer que seja a linguagem de programação que faça a conexão com o banco.

Uma lista de clientes e usuários do MySQL pode ser obtida no tópico 1.7, onde a ferramenta foi aplicada nos mais variados tipos de projetos, em sua grande maioria de missão crítica, e atingindo o sucesso desejado.

1.3 Compatibilidade

A seguir, algumas informações gerais sobre a compatibilidade do MySQL no que diz respeito a sistemas operacionais que o suportam e versão de SQL utilizada.

1.3.1 Sistemas operacionais

A maioria dos sistemas operacionais existentes no mercado suporta a execução do MySQL. Por ser um programa escrito em C e C++, isto torna extremamente fácil a sua portabilidade entre diferentes plataformas. Dentre as principais, pode-se destacar:

- Linux (Fedora Core, Debian, SuSE, RedHat);
- Unix (Solaris, HP-UX, AIX, SCO);
- FreeBSD;
- Mac OS X Server;
- Windows (2000, 2003, XP).

As plataformas mais indicadas para o uso com o MySQL são aquelas que possuem a biblioteca thread e o gerenciador de arquivos mais estáveis, entre outros fatores.

1.3.2 Versões de SQL

O MySQL tem como base o padrão de SQL-92 ODBC levels 0-3.51.

Um dos principais objetivos da ferramenta é evoluir para utilizar como base o padrão SQL-99. Contudo, nem todos os códigos para essa conversão poderiam ser implementados na versão atual, sem que se abrisse mão da velocidade e confiança do banco de dados. Isto se deve ao fato de alguns comandos do padrão SQL-99 demandarem maior processamento de dados, o que poderia comprometer algumas das principais características da ferramenta.

Entre os próximos recursos do programa, faz parte atingir o objetivo de ter o padrão SQL-99 como base (o qual já é utilizado em grande parte), mantendo sua capacidade de processamento e segurança.

1.4 Características

A seguir, algumas das principais características existentes no MySQL.

1.4.1 SGBD

O MySQL, além de banco de dados, contém todas as características de um SGBD (Sistema Gerenciador de Banco de Dados), que é o MySQL Server. Além de armazenar os dados, a ferramenta provê todas as características de multiacesso a estes, entre outras funcionalidades de um SGBD, como, por exemplo, gerenciamento de acesso, integridade dos dados e relacional, concorrência, transações, entre outros. O conceito de SGBD será abordado mais detalhadamente no tópico A.4 do Apêndice A.

1.4.2 Portabilidade

Desenvolvido utilizando as linguagens de programação C e C++, unido com o uso de GNU Automake, Autoconf e Libtool, torna o MySQL uma aplicação altamente portátil entre diferentes sistemas, plataformas e compiladores. Além disso, fornece sua API para várias outras linguagens, como Java, Python, PHP, Perl, C, C++, entre outras.

1.4.3 Multithreads

Usa programação de threads utilizando-as diretamente no kernel da plataforma. Além de aumentar significativamente a velocidade de processamento, ainda facilita a integração da ferramenta em hardwares com mais de uma CPU.

1.4.4 Formas de armazenamento

O MySQL disponibiliza vários tipos de tabelas para armazenamento de dados, tendo cada tipo suas próprias características. A vantagem dessa variedade de tabelas é a possibilidade de escolher o tipo em cada situação diferente. Enquanto um tipo prioriza velocidade, outro prioriza volume de dados, entre outras características. Para conhecer com mais detalhes os principais tipos de tabelas e métodos de armazenamento do MySQL, leia o tópico 3.1.

1.4.5 Velocidade

Maior velocidade no acesso aos dados em razão de vários fatores em seu desenvolvimento como tabelas ISAM (substituídas pelo novo mecanismo MyISAM na versão 5), utilização de cachês em consultas, utilização de indexação BTREE para tabelas do tipo HEAP, algoritmos de buscas, entre outros recursos.

1.4.6 Segurança

Sistema gerenciador de conexões que trabalha com criptografia no tráfego de senhas. Ainda, em complemento ao uso de autenticação baseado em senhas, é possível habilitar uma espécie de firewall de autenticação, responsável por habilitar as devidas conexões somente para estações e domínios especificados em sua lista de acesso.

1.4.7 SQL

O SQL utilizado pelo MySQL é altamente veloz. Isto se deve ao fato de a linguagem ter sido implementada por meio de códigos e funções altamente otimizadas pelos seus desenvolvedores.

Essa característica de otimização tem como consequência uma grande vantagem e uma desvantagem: o lado bom é a velocidade na execução dos códigos SQL. O lado ruim é que para otimizar tais códigos, nem todos os padrões da versão mais atual do SQL foram migrados para o MySQL, pois poderiam comprometer a velocidade do banco de dados. O MySQL baseia-se nas versões mais atuais do SQL, em parte incluindo sua versão 2003. Mas na maioria dos casos, essa desvantagem não deve influenciar na aplicação.

1.4.8 Fulltext search

A funcionalidade de busca no estilo fulltext search já estava presente na versão 4 do MySQL, tendo aparecido pela primeira vez entre as versões 3 e 4. Este tipo de busca é um dos mais utilizados em grandes soluções integradas com bancos de dados.

Existem dois modos com que o MySQL gerencia os índices, sendo um a criação de índices, tendo como base campos do tipo texto (recomendado para bancos de dados de pequeno e médio portes), e outro em uma tabela própria de índices, fornecendo um melhor desempenho para bancos de dados de grande porte.

1.4.9 Capacidades

O MySQL tem um alto poder de execução e de armazenamento. Dependendo da plataforma onde a ferramenta será utilizada, suas tabelas poderão armazenar espaços extraordinários, ficando limitadas somente ao tamanho máximo de arquivos com que a plataforma em questão pode manipular. No caso de tabelas do tipo InnoDB, cujo armazenamento pode ser realizado por um ou mais arquivos separados, é possível armazenar até 65.536 TB (terabytes).

No caso de expressões SQL, o SGBD suporta execuções de scripts SQL com até 61 uniões de tabelas (joins), e em se tratando de velocidade de execução, o MySQL pode ser enquadrado entre os mais velozes, se não o mais veloz, justamente por este ter sido um dos motivos que levou seus programadores a desenvolvê-lo, baseado em tecnologias que permitiram tal fato.

O MySQL é um banco de dados extremamente poderoso, pronto para executar mais de um bilhão de consultas por dia de um site, ou até mesmo processar milhares de transações por minuto, conforme os próprios papers e artigos do MySQL AB.

1.5 Novidades na versão 5

A cada nova versão do MySQL, novas funcionalidades são implementadas, mas foi com a versão 5 que o programa alcançou definitivamente sua posição como concorrente de grandes bancos de dados existentes no mercado.

Os recursos que ainda faltavam no MySQL foram implementados e disponibilizados nesta versão. Dentre as principais demandas existentes e atendidas, podemos citar as stored procedures e triggers, que são blocos de código armazenados no banco de dados, geralmente de execuções de alta frequência, ou acionadas por eventos. A seguir, um breve resumo das principais funcionalidades incorporadas no MySQL 5.

1.5.1 Stored procedures

Dentre as funcionalidades mais aguardadas para o MySQL, sem dúvida podemos citar as stored procedures, que são blocos de código SQL armazenados no servidor, os quais são chamados (ou se preferir, invocados) a partir das aplicações integradas ao banco de dados.

Seu conceito foi padronizado pela primeira vez na versão SQL-99. De lá para cá, os grandes bancos de dados têm disponibilizado este recurso, que além de centralizar

códigos para não haver redundância deles em diversas partes do sistema, também aumenta a velocidade de processamento das consultas e torna os códigos das aplicações mais claros e simples, pois várias consistências sobre dados e relacionamentos podem ser feitas no lado do servidor de banco de dados. A versão implementada no MySQL baseia-se no modelo ANSI 2003.

Maiores detalhes sobre stored procedures serão abordados no tópico 6.2.

1.5.2 Triggers

Assim como as stored procedures, os triggers (também conhecidos como gatilhos) são blocos de código SQL armazenados no servidor, mas não para serem invocados pela aplicação integrada ao banco de dados, e sim ser iniciados a partir de algum evento pré-cadastrado que ocorre no sistema (determinado horário do dia no caso dos event schedulers, antes de uma inserção ou alteração, entre outras possibilidades). Maiores detalhes sobre triggers serão abordados no tópico 6.5.

1.5.3 Visões

Outra novidade na versão 5 do MySQL é a implementação de visões, que são consultas pré-programadas ao banco de dados que permitem unir duas ou mais tabelas e retornar uma única tabela como resultado, quando invocadas. Além disso, podem ser utilizadas para filtrar informações, exibindo somente os dados específicos de uma determinada categoria de uma ou mais colunas da tabela.

O uso de visões torna o banco de dados mais prático, rápido e otimizado, pois operações frequentes com uniões de tabelas podem agora ser centralizadas. É possível também utilizá-las para controle de acesso, permitindo que determinados usuários acessem dados de uma visão, mas não das tabelas utilizadas por esta, restringindo assim algumas informações que não deveriam ser disponibilizadas a todos. Maiores detalhes sobre views serão abordados no tópico 3.5.

1.5.4 Cursores

A partir desta nova versão do MySQL também é possível a utilização de cursores para navegação em conjuntos de resultados. De forma simples, é possível navegar pelos registros de uma tabela a partir de laços de repetição, permitindo realizar operações necessárias e transações à parte para cada linha da tabela. Maiores detalhes sobre cursores serão abordados no tópico 6.6.

1.5.5 Information Schema

Assim como nos grandes bancos de dados e sistemas gerenciadores deles, existem algumas tabelas responsáveis apenas pela organização dos recursos do banco de dados, conhecidos como dicionário de dados, ou metadados. Desta forma, é possível realizar consultas sobre a estrutura do banco de dados por meio dessas tabelas.

1.5.6 Transações distribuídas XA

O que garante a qualidade e confiabilidade de uma transação em um banco de dados é o conjunto de propriedades conhecidas como ACID (Atomicidade, Consistência, Isolamento e Durabilidade). O conceito transações distribuídas XA – uma espécie de extensão da ACID – é que fornece a possibilidade de gerenciamento dessas transações realizadas com a união de múltiplos bancos de dados (transações distribuídas) para a execução de uma mesma transação.

Por exemplo, em determinadas situações pode surgir a necessidade de integração de duas bases de dados distintas, mas que em algum momento necessitem uma da outra para realizar uma operação.

Maiores detalhes sobre transações distribuídas XA serão abordados no tópico 6.3.

1.5.7 Integridade referencial

Os relacionamentos entre tabelas distintas agora são gerenciados pelo banco de dados nos momentos de inclusão, alteração e exclusão. Este recurso visa a manter confiável as relações existentes no banco de dados. Maiores detalhes sobre integridade referencial serão abordados no tópico A.4.3 do Apêndice A.

1.5.8 Replicação

Visando a aumentar ainda mais a disponibilidade do servidor, tornou-se possível a partir da versão 5 configurar servidores réplicas (clones) com o MySQL, unidirecionais e bidirecionais. Em outras palavras, réplicas são outros servidores que estão com suas informações sincronizadas em um servidor principal, geralmente visando a aumentar o poder de processamento e disponibilidade na parte de hardware. Maiores detalhes sobre replicação serão abordados no Capítulo 9.

1.5.9 Clusterização

Assim como o objetivo da utilização da replicação, a clusterização é baseada na integração e sincronismo de dois ou mais servidores para dividirem a demanda de execuções entre si. Novamente, o principal objetivo deste recurso é possibilitar que mais hardware seja utilizado, podendo alcançar poder de processamento de quantos hardwares estiverem integrados ao cluster.

Além da sincronização de um cluster, é possível também especificar um balanceador de cargas para este. Desta forma, este recurso não gerenciará apenas o redirecionamento de servidores secundários no caso de o primário sair do ar, mas sim balanceará as consultas recebidas pelo cluster e irá distribuí-las pelos servidores de acordo com suas ociosidades.

Maiores detalhes sobre clusterização serão abordados no Capítulo 10.

1.6 Comparações com outros bancos de dados

O MySQL conta com uma série de recursos para análise do tempo de execução de operações. Isto possibilitou que seus desenvolvedores realizassem alguns testes comparativos com outros bancos de dados.

Esses testes foram realizados baseados em um computador com Windows NT4 e acesso via ODBC. Dentre os bancos de dados mais conhecidos, os resultados mostrados nas tabelas 1.1 e 1.2 foram obtidos:

Tabela 1.1 – Leitura de 2.000.000 por índice

Banco de dados	Segundos
mysql_odbc	464
db2_odbc	1.206
ms-sql_odbc	1.634
oracle_odbc	20.800
sybase_odbc	17.614

Tabela 1.2 – Inserção de 350.768 linhas

Banco de dados	Segundos
mysql_odbc	619
db2_odbc	3.460
ms-sql_odbc	4.012
oracle_odbc	11.291
sybase_odbc	4.802

Outras vantagens do MySQL é a possibilidade de realizar exclusões de tabelas diferentes em um único comando **DELETE**, permitir utilizar tabelas de bancos de dados distintos em uma mesma consulta de união (**JOIN**), maior capacidade de colunas em uma única tabela, entre outros. Uma tabela comparativa completa entre o MySQL e diversos outros bancos de dados pode ser obtida em <http://dev.mysql.com/tech-resources/crash-me.php>.

- ✦ **IMPORTANTE:** Ao se comparar dois ou mais bancos de dados é fundamental levar em conta os aspectos técnicos de que cada aplicação ou servidor necessita. Não são todos os casos em que as aplicações utilizarão tabelas de tamanho superior a 65.000TB (terabytes), ou que utilizarão 2.500 colunas em uma única tabela. Portanto, um dos fatores mais importantes na comparação entre os bancos de dados é a sua velocidade de execução, entre outros fatores técnicos.

1.7 Casos de uso

O portfólio de clientes e usuários do MySQL conta com centenas de grandes projetos bem-sucedidos e de grande porte ao redor do mundo. Entre os casos mais famosos, vale a pena citar o da NASA.

O departamento NAIS (dentro da NASA) é responsável pela gerência dos contratos da empresa, movimentando transações que chegam a milhões de dólares por ano. Em razão da grande demanda de sua utilização e altos custos de manutenção do banco de dados Oracle utilizados anteriormente no sistema, a equipe do NAIS optou por migrar os seus bancos de dados para uma solução open source. Com base nisso, estudaram qual seria a melhor opção de código aberto e optaram por utilizar o MySQL.

De início, muitas pessoas ficaram apreensivas com a alteração do banco de dados, contudo o sucesso da migração e as vantagens oferecidas por ele garantiram a satisfação de todos. Perguntaram ao diretor do NAIS os motivos que o levaram a migrar para o MySQL, e ele respondeu que o custo de suporte técnico do MySQL era praticamente 1% do que o NAIS gastava com suporte técnico da Oracle.

Além deste, diversos outros grandes casos podem ser destacados em projetos de missão crítica, entre eles: Bradesco, Telemar, HP-Compaq, Sony, Cisco Systems, Dataprev, Nokia, Motorola, Alcatel, Ibest, Yahoo Finance, US Navy, US Army, US Federal Reserve Bank, Suzuki, Hyperion, Lycos, além de muitos outros que podem ser obtidos no site oficial do MySQL.

Dentre os principais cases do MySQL, podem-se destacar bancos de dados com mais de 60 mil tabelas, tabelas com aproximadamente 5 trilhões de registros (linhas), ressaltando que esses valores são apenas cases existentes do MySQL, sendo o seu limite de capacidade muito acima dos apresentados.

1.8 Sites relacionados

- Site oficial:

<http://www.mysql.com>

- Site de ajuda (fóruns, lista de discussão, canais IRC):

<http://www.mysql.com/support/>

- Site do MySQL Brasil:

<http://www.mysqlbrasil.com.br>

- Site da licença de uso GNU:

<http://www.gnu.org>