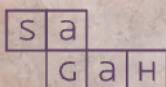


MODELAGEM E DESENVOLVIMENTO DE BANCO DE DADOS

Fabrício Felipe Meleto Barboza



SOLUÇÕES
EDUCACIONAIS
INTEGRADAS



Tipos de bancos de dados

Objetivos de aprendizagem

Ao final deste texto, você deve apresentar os seguintes aprendizados:

- Identificar os vários tipos de bancos de dados.
- Reconhecer os tipos de bancos de dados pela sua sintaxe.
- Relacionar os bancos de dados com os vários tipos existentes.

Introdução

Neste capítulo, você vai estudar sobre como identificar e reconhecer os diversos tipos de bancos de dados pela sua sintaxe. Assim, será capaz de reconhecer os padrões de cada sintaxe e identificar o banco de dados associado. Você verá exemplos de uso e os diferentes padrões em cada um dos bancos de dados: PostgreSQL, Microsoft SQL Server, MySQL/MariaDB e Oracle. Adicionalmente, verá como relacionar os bancos de dados em função dos vários tipos existentes atualmente.

Os vários tipos de bancos de dados

Vamos relembrar o conceito de bancos de dados? Segundo Korth, Silberschatz e Sudarshan (2012), banco de dados “[...] é uma coleção de dados inter-relacionados, representando informações sobre um domínio específico”.

Dessa forma, estudaremos, neste capítulo, os diversos tipos de bancos de dados mais comumente utilizados e conhecidos atualmente, limitando-nos às quatro primeiras posições desse mesmo ranking de utilização.

Usualmente, as empresas utilizam um dos quatro bancos de dados a seguir em suas aplicações, sistemas ou projetos:

- MySQL/MariaDB;
- Oracle;
- Microsoft SQL Server;
- PostgreSQL.

Analisaremos cada um desses bancos de dados, abordando sua origem, plataforma compatível, características únicas e escopo de trabalho.

MySQL/MariaDB

O MySQL foi inserido junto ao MariaDB em virtude deste segundo ser um *fork* do primeiro, tendo estrutura e comandos bem semelhantes e de funcionamento idêntico. O MySQL era um banco de dados *opensource* até a sua compra pela Oracle em 2009. Esse paralelismo é tão grande e forte que os próprios administradores de bancos de dados do MySQL conseguem, de forma bem fácil, gerenciar um banco de dados do tipo MariaDB e vice-versa.



Saiba mais

Fork significa uma bifurcação. Portanto, o MariaDB ser um *fork* do MySQL quer dizer que o MySQL teve uma bifurcação em sua história que originou o MariaDB.

A instalação de um banco de dados do tipo MySQL ou MariaDB pode ser realizada tanto em sistema operacional Linux nas distribuições CentOS, RedHat, Ubuntu, Debian ou Fedora quanto no Windows.



Fique atento

Lembre-se que o banco de dados MySQL é quase que totalmente compatível com o MariaDB exatamente porque o MariaDB é um *fork* do MySQL. Um dos criadores originais do MySQL desenvolveu o MariaDB.

Até mesmo a forma de abrir e realizar uma conexão ao banco de dados é semelhante entre o MySQL e o MariaDB. Outra característica curiosa é a semelhança entre os dois logotipos que representam cada um destes bancos de dados (Figura 1).



Figura 1. Logos do MySQL (a) e do MariaDB.

Fonte: MySQL (2018, documento on-line) e MariaDB

Repare que ambos os animais presentes nos logotipos fazem parte do ambiente marinho: um golfinho para o MySQL e uma foca para o MariaDB. O licenciamento do MySQL é pago e o MariaDB veio para suprir a comunidade *opensource* e, assim, não ter custo pelo licenciamento do banco de dados.

Sua grande disseminação se dá em virtude de diversos sistemas e linguagens de programação terem uma fácil e rápida conexão com ele. Como linguagens, destacam-se o PHP e Python. Já para os sistemas, temos Wordpress e Magento.

Oracle

A gigante de tecnologia Oracle possui um banco de dados de mesmo nome e que é um dos bancos de dados mais utilizados do mundo.



Saiba mais

Um dos maiores salários dos profissionais de tecnologia da informação é o dos administradores do banco de dados Oracle.

Seu lançamento ocorreu em 1980 e ele domina o mercado desde então. É um banco de dados do tipo relacional, tem uma base confiável e robusta, e é muito utilizado para projetos muito grandes, com várias transações ocorrendo em um mesmo segundo, tanto de leitura quanto de escrita, além de uma incrível performance.

Como seu licenciamento é pago e a cifra gira em um valor muito alto, na casa de cinco ou seis dígitos, as empresas que o utilizam são de porte médio para grande em função do alto custo — além do já mencionado custo diferenciado pelo profissional que o administra.

A Oracle, exatamente por saber de seu diferencial competitivo no mercado, lançou diversas certificações para capacitar o profissional que irá trabalhar com o seu sistema de banco de dados. Essas certificações também atestam, para a empresa contratante, o conhecimento e a experiência do profissional em questão. Conforme lembra Almeida (2018), os demais sistemas de bancos de dados também possuem certificação, mas a trilha de certificação da Oracle é a mais estruturada e reconhecida no mercado de trabalho exatamente pela importância do banco de dados em questão. O banco de dados Oracle é instalado em sistemas operacionais Linux e também em sistemas operacionais Microsoft Windows.

Microsoft SQL Server

A gigante de Redmond lançou o seu banco de dados em 1989 e vem evoluindo esse banco desde então. Um limitador do sucesso do Microsoft SQL Server é que, até pouco tempo atrás, só era possível instalá-lo em sistema operacional Windows, deixando toda a comunidade que trabalha com *opensource* Linux sem a possibilidade de usá-lo. Eis que a Microsoft, em 2017, lançou uma versão do banco de dados para Linux de forma a reverter a situação e alavancar seu uso.

Como o Oracle, o Microsoft SQL Server tem o seu licenciamento na modalidade paga. O grande trunfo do Microsoft SQL Server é a sua aproximação da linguagem .NET e, também, seus desenvolvedores.

PostgreSQL

Outro banco de dados de licenciamento livre é o PostgreSQL. Ele é do tipo relacional e foi lançado em 1989 pela PostgreSQL Global Development Group.

O PostgreSQL, assim como o MySQL e o MariaDB, tem forte apelo para aplicações WEB de pequeno e médio porte. Grande parte dos sites e sistemas do tipo WEB utilizam um banco de dados PostgreSQL para armazenamento dos dados.

Diversos sistemas operacionais são suportados pelo PostgreSQL (Figura 2), sendo tanto Linux quanto o Microsoft Windows. Dessa forma, seu concorrente no mundo dos bancos de dados acaba sendo, por eliminação e proximidade, o MariaDB.

Seu logo é composto pela representação da cabeça de um elefante, remetendo à lenda da ideia de que um elefante nunca se esquece do que ou quem ele visualizou.



Figura 2. Logo do PostgreSQL.

Fonte: PostgreSQL (2018, documento on-line).

Os tipos de bancos de dados e sua sintaxe

Para facilitar o seu entendimento, serão disponibilizados exemplos de inserção, edição e deleção de um dado. Esses comandos podem ser utilizados em qualquer um dos bancos de dados MySQL, MariaDB, Oracle, Microsoft SQL Server ou PostgreSQL.

Como complemento e para diferenciar os bancos, também serão demonstrados os comandos particulares de cada banco de dados para exibir todos os *databases* e o comando para mostrar todas as tabelas de um *database*. Esses comandos serão exemplificados nos quatro bancos de dados descritos no parágrafo anterior. Para a inserção de dados, é utilizada a sintaxe a seguir.

```
INSERT INTO
    table_name ( field1, field2,...fieldN )
VALUES
    ( value1, value2,...valueN );
```

Exemplificando em um *insert* na tabela “cliente” o cadastro de José da Silva, podemos usar algo semelhante a:

```
INSERT INTO
    cliente ( CPF, nome, telefone )
VALUES
    ( '123.456.789-00', 'José da Silva', '11-911111111' );
```

Já para a edição de dados, é utilizada a sintaxe a seguir:

```
UPDATE
    table_name
SET column1 = value1, column2 = value2, columnN = valueN
WHERE
    condition;
```

Exemplificando em uma edição da tabela “cliente” e no cadastro de José da Silva, modificando seu número de telefone, podemos usar algo semelhante a:

```
UPDATE
    cliente
SET telefone = '11-922222222'
WHERE
    nome = 'José da Silva';
```

Para a exclusão de dados, é utilizada a sintaxe a seguir:

```
DELETE FROM
    table_name
WHERE
    column1 = value1, column2 = value2, columnN = valueN;
```

Exemplificando a deleção com o cadastro de José da Silva da tabela “cliente”, podemos usar algo semelhante a:

```
DELETE FROM
    cliente
WHERE
    nome = 'José da Silva';
```


Portanto, os comandos apresentados anteriormente podem ser utilizados em qualquer um dos quatro bancos de dados estudados aqui: MySQL ou MariaDB, Oracle, Microsoft SQL Server ou, por último, mas não menos importante, o PostgreSQL.

Os comandos de exibir os *databases* ou exibir as tabelas são particulares para cada banco de dados e, dessa forma, foram separados. A seguir, são apresentados cada um deles.

MySQL/MariaDB

A sintaxe do MySQL é praticamente idêntica ao MariaDB exatamente pelo fato deste segundo ser um *fork* do primeiro.

Para exibir todas os *databases* no MySQL ou no MariaDB, utiliza-se:

```
SHOW DATABASES;
```

O retorno do comando será o nome de todos os *databases* contidas no banco de dados independentemente de ter, ou não, tabelas, um *database* por linha.

Já para exibir todas as tabelas de um *database*, primeiro é necessário fazer a seleção do *database* desejada:

```
USE nomeDataBase;
```

Após a seleção do *database*, pode-se realizar a consulta das tabelas com o comando:

```
SHOW TABLES;
```

O retorno desse comando será uma listagem de todas as tabelas daquele *database* selecionada previamente, uma tabela por linha.

Oracle

Para o Oracle, os comandos de exibição são diferentes do MySQL ou do MariaDB. Caso precise exibir todos os *databases*, você pode usar o comando:


```
SELECT NAME  
FROM v$database;
```

Novamente, o retorno do comando será o nome de todos os *databases* contidos no banco de dados independentemente de ter ou não tabelas, um *database* por linha. Se a necessidade é exibir todas as tabelas de um *database*, antes de executar o comando necessário para conseguir o resultado esperado, é preciso selecionar o *schema* desejado:

```
CONNECT nomeDataBase;
```

Após isso, rode o comando a seguir para exibir as tabelas:

```
SELECT table_name  
FROM USER_TABLES;
```

Microsoft SQL Server

Para que o Microsoft SQL Server liste os *databases*, é necessário fazer a consulta de forma um pouco diferente dos demais bancos:

```
SELECT [name]  
FROM master.dbo.sysdatabases;
```

Caso não queria que o retorno inclua os *databases* do sistema, ou seja, que exiba somente os *databases* que foram criados, pode-se usar o comando:

```
SELECT [name]  
FROM master.dbo.sysdatabases  
WHERE dbid > 4;
```

Já para listar as tabelas de um *database*, é necessário escolher um *database* com o comando:

```
USE nomeDataBase;
```

Após o comando de escolha do *database*, execute o comando para realizar a consulta de todas as tabelas daquele *database*:

```
SELECT *  
FROM sys.Tables;
```

Perceba que o Microsoft SQL Server é menos preparado para trazer esse tipo de informação de *database* ou tabela se comparado aos demais bancos de dados.

PostgreSQL

Exibir todos os *databases* no PostgreSQL é bem simples. Basta digitar:

```
\list
```

O retorno do comando será o nome de todos os *databases* contidos no banco de dados independentemente de ter, ou não, tabelas, um *database* por linha. Já para exibir todas as tabelas de um *database*, primeiro é necessário fazer a seleção do *database* desejada com o comando:

```
\connect nomeDataBase
```

Após a seleção do *database*, pode-se realizar a consulta das tabelas com o comando: `\dt`

O retorno desse comando será uma listagem de todas as tabelas daquele *database* selecionado previamente, uma tabela por linha.

Relação entre os bancos de dados

A relação entre os bancos de dados que utilizam a linguagem SQL para consultas é idêntica para o trabalho de manipulação dos dados envolvidos, seja o comando de INSERT, UPDATE ou até mesmo o DELETE.

Partindo dessa premissa e também pelo que explica Gomes (2018), a diferenciação entre os bancos de dados, mesmo os que utilizam a linguagem SQL, ocorre no âmbito de gerenciamento desses bancos, seja na consulta aos nomes de *databases* que o mesmo possui ou ainda na listagem do nome das tabelas (Quadro 1).

Quadro 1. Relação de banco de dados que utilizam a linguagem SQL

| Banco de dados | Tipo | Exibir os <i>databases</i> |
|----------------------|-----------|---|
| MySQL / MariaDB | SQL | Show databases; |
| Microsoft SQL Server | SQL | SELECT [name] FROM master. dbo.sysdatabases; |
| Oracle | SQL/NoSQL | SELECT NAME FROM v\$database; |
| PostgreSQL | SQL | \list |
| MongoDB | NoSQL | show dbs; |

**Fique atento**

Lembre-se de que os bancos de dados que utilizam a linguagem SQL para realizar a manipulação dos dados têm a mesma sintaxe para esse fim.

**Saiba mais**

Edgar Frank "Ted" Codd, pesquisador da IBM na época e que lançou o artigo intitulado *Relational Model of Data for Large Shared Data Banks*, foi considerado o pai dos bancos de dados relacionais.



Referências

ALMEIDA, R. *Certificação Oracle*. 2018. Disponível em: <www.linhadecodigo.com.br/artigo/451/certificacao-oracle.aspx>. Acesso em: 04 jun. 2018.

GOMES, E. H. *Linguagem SQL: linguagem de manipulação, consulta e controle de dados*. 2018. Disponível em: <ehgomes.com.br/disciplinas/bdd/sql.php>. Acesso em: 04 jun. 2018.

KORTH, H. F.; SILBERSCHATZ, A.; SUDARSHAN, S. *Sistema de banco de dados*. 6. ed. Rio de Janeiro: Campus, 2012.

MARIADB. 2018. Disponível em: <<http://mariadb.org/>>. Acesso em: 04 jun. 2018.

MYSQL. 2018. Disponível em: <<https://www.mysql.com/>>. Acesso em: 04 jun. 2018.

POSTGRESQL. 2018. Disponível em: <<https://www.postgresql.org/>>. Acesso em: 04 jun. 2018.

Leituras recomendadas

MIRANDA, W. *Certificações Oracle: diferença entre o desenvolvedor e o DBA*. 2018. Disponível em: <aprendaplsql.com/oracle/certificacao/certificacoes-oracle-diferenca-entre-o-desenvolvedor-e-o-dba/>. Acesso em: 04 jun. 2018.

REZENDE, R. *Conceitos fundamentais de banco de dados*. 2006. Disponível em: <<https://www.devmedia.com.br/conceitos-fundamentais-de-banco-de-dados/1649>>. Acesso em: 04 jun. 2018.

Encerra aqui o trecho do livro disponibilizado para esta Unidade de Aprendizagem. Na Biblioteca Virtual da Instituição, você encontra a obra na íntegra.

Conteúdo:

