

HSQLDB: um banco de dados livre escrito em Java

Carlos Emilio Padilla Severo

Este artigo descreve o banco de dados livre HSQLDB, demonstrando suas características potencialidades e principais ferramentas.

Introdução

O Hypersonic SQL Database (HSQLDB) é um projeto de banco de dados livre, escrito em Java, que permite a manipulação de banco de dados em uma arquitetura cliente-servidor, ou standalone. Uma grande vantagem de utilização do HSQLDB é a possibilidade de agregarmos o banco de dados ao pacote de nossas aplicações. O banco é multiplataforma e ocupa um pequeno espaço em disco. Outra característica do banco é a possibilidade de manipularmos bancos de dados em disco, memória ou em formato texto. Trata-se de uma tecnologia flexível e muito útil na construção de aplicações que manipulem banco de dados.

Nesse artigo, iremos ilustrar as principais características do banco de dados, destacando-se o conjunto de ferramentas constantes no pacote de distribuição. Trata-se de um banco de dados excelente para pequenas aplicações escritas em Java.

Página do projeto:

HSQLDB project: <http://hsqldb.sf.net>

Componentes do HSQLDB

No núcleo do pacote estão o RDBMS e o driver JDBC que disponibilizam as principais funcionalidades do banco, que são: o gerenciador de banco de dados relacional e o driver para conexão através de aplicações Java. Além disso, o pacote contém um conjunto de componentes e ferramentas para execução do SGBD. Através das ferramentas podemos criar estruturas de um banco de dados, acessar bancos de dados através de ferramentas para consulta, exportar e importar esquemas entre bancos de dados distintos. Além de outras facilidades disponibilizadas para o desenvolvedor. A seguir, veremos a descrição de cada um desses componentes:

HSQLDB JDBC Driver: o pacote de distribuição disponibiliza um driver padrão JDBC para conexão de aplicações Java com o SGBD. A conexão com o banco de dados segue um modelo de protocolo proprietário, mas também podemos realizar uma conexão via rede, através de protocolos Internet. Nesse artigo, veremos como realizar uma conexão de uma aplicação Java com o banco de dados de diversas formas.

Database Manager: duas versões de ferramentas para gerenciamento de banco de dados são disponibilizadas. Uma ferramenta escrita usando AWT e uma outra versão usando Swing. Trata-se de uma ferramenta gráfica para visualização do esquema do banco de dados, conjunto de tabelas e submissão de instruções SQL. A versão AWT pode ser executada como um Applet dentro de um navegador.

Transfer Tool: essa é uma ferramenta utilizada para transferências de esquemas SQL ou dados de uma fonte JDBC para outra. Trata-se de uma ferramenta bastante útil quando desejamos realizar uma migração de banco de dados, transferindo esquemas e o conjunto de dados entre duas tecnologias distintas.

Query Tool: a finalidade dessa ferramenta é prover ao desenvolvedor um software para interação com o SGBD através do envio de instruções SQL a partir de uma linha de comando, ou através de um arquivo texto contendo um conjunto de instruções. A ferramenta apresenta um shell interativo ao usuário.

SQL Tool: outra ferramenta do pacote para construção e submissão de instruções SQL ao banco de dados.

HSQLDB RDBMS

Este é o módulo gerenciador do banco de dados relacional disponibilizado pela ferramenta. O HSQLDB pode ser executado de diversas maneiras. Podemos executá-lo como um servidor de banco de dados, ou como um processo em aplicações standalone. Para cada modo de execução temos uma forma específica de conexão com o banco de dados. A forma de conexão é dada por um protocolo informado na string que forma a URL de conexão com o banco.

Executando no modo servidor

Para executarmos o gerenciador no modo servidor, devemos invocar o programa Server. O programa recebe alguns argumentos para que possamos iniciar ou criar um novo banco de dados. Podemos passar alguns argumentos por linha de comando, um deles é o argumento -? que apresenta uma ajuda na tela. Abaixo, veremos a chamada ao servidor iniciando o banco de dados denominado DBEmpresa.

```
java org.hsqldb.Server -database.0 DBEmpresa -dbname.0 empresa
```

O arquivo hsqldb.jar deverá estar gravado em um diretório mapeado através da variável de ambiente classpath, para que a linha de comando acima tenha efeito. O primeiro argumento -database.0 informa o nome do banco de dados. Enquanto que o segundo argumento, o -dbname.0, informa um alias para o banco de dados.

Caso o servidor seja carregado sem informarmos algum banco de dados, o mesmo gera um banco de dados padrão denominado test.

Executando no modo servidor Web

Uma modalidade de execução do SGBD como um servidor Web, o qual permite a conexão de uma aplicação Java via protocolo HTTP. Essa característica é útil quando estamos tentando disponibilizar o banco de dados em máquinas protegidas por firewalls. A linha abaixo, apresenta a forma de execução no modo servidor Web.

```
java org.hsqldb.WebServer
```

Conectando uma aplicação a um servidor

Aplicações clientes realizam uma conexão a um servidor em execução via JDBC. Para isso, a aplicação deverá utilizar o driver disponível no próprio hsqldb.jar. A seguir, veremos um exemplo de conexão.

```
try{
    Class.forName("org.hsqldb.jdbcDriver");
}
catch(SQLException e){
    System.out.println("Erro ao carregar o driver JDBC. ");
}
Connection con = DriverManager.getConnection("jdbc:hsqldb:hsqldb://localhost/DBEmpresa", "sa", "");
```

A conexão do exemplo acima utiliza a porta "9001", que é a porta padrão para conectar a aplicação Java, via JDBC, com o banco de dados DBEmpresa. O usuário padrão do HSQLDB é o "sa", cuja senha é uma string vazia.

Conectando uma aplicação no modo standalone

Neste caso, o banco de dados fará parte da própria aplicação. Ou seja, ambos serão empacotados juntos e ambos rodarão em uma mesma JVM. Para pequenas aplicações que serão executadas em um desktop, sem o uso da rede, é um conceito muito interessante. Veja, logo abaixo, um exemplo de como realizar uma conexão com um banco de dados standalone:

```
Connection con = DriverManager.getConnection("jdbc:hsqldb:file:/banco/DBEmpresa", "sa", "");
```

Conectando uma aplicação no modo memória

O HSQLDB permite que possamos executar o SGBD de forma que os dados sejam mantidos em memória. Essa é uma técnica que mantém todo o conjunto de dados de uma tabela na memória do



computador, permitindo uma melhor performance do SGBD. Esse modo de execução deverá ser utilizado somente em algumas situações especiais. A seguir, veremos o exemplo de conexão utilizando o protocolo de memória.

```
Connection con = DriverManager.getConnection("jdbc:hsqldb:mem:empresa", "sa", "");
```

Arquivos do banco de dados

Um banco de dados consiste de dois a cinco arquivos. Todos compartilhando o mesmo nome, mas com extensões que os diferenciam. Por exemplo, um banco de dados chamado DBEmpresa é constituído pelos seguintes arquivos:

DBEmpresa.properties: este é o arquivo de propriedades que contém alguns ajustes gerais das propriedades do banco de dados. As propriedades poderão ser editadas pelo desenvolvedor, o arquivo mantém propriedades como versão do SGBD, características do cache, se o banco vai ser inicializado em modo somente leitura, entre outras.

DBEmpresa.script: nesse arquivo estão contidos os scripts em SQL para criação do esquema do banco de dados. O script cria a estrutura de tabelas, índices, restrições, usuários, etc. Além disso, mantém o conjunto de dados inseridos nas tabelas persistentes do banco.

BEmpresa.log: arquivo no formato texto que mantém as alterações recentes sobre o banco de dados. O arquivo identifica o usuário que acessou o banco e ação realizada pelo mesmo.

DBEmpresa.data: como o próprio nome sugere, mantém os dados do banco que estão armazenados em tabelas do tipo cached. Esse tipo de arquivo será criado quando existirem tabelas do tipo cached no esquema do banco de dados. Para tabelas em memória, os dados são persistidos no arquivo de script.

DBEmpresa.backup: trata-se de um arquivo compactado, formato ZIP, que mantém o último estado consistente do banco de dados.

Todos os arquivos citados são necessários, caso existam no diretório onde foi criado o banco. Nunca deverão ser deletados. Se o desenvolvedor não criou tabelas do tipo cached, os arquivos .data e .backup não serão criados.

Tipos de tabelas geradas pelo HSQLDB

O HSQLDB suporta tabelas temporárias e três tipos de tabelas persistentes. As tabelas temporárias não são armazenadas em disco e sobrevivem somente enquanto a conexão com o banco de dados estiver ativa. Tabelas temporárias são visíveis através da conexão que as criou, conexões concorrentes não terão acesso as mesmas.

Os modelos de tabelas persistentes são: tabelas em memória, tabelas cached e tabelas no formato de arquivo texto. Tabelas em memória é o padrão de criação quando invocamos o comando CREATE TABLE. Os dados desse tipo de tabela são mantidos em memória, mas qualquer alteração em sua estrutura ou conjunto de dados, serão reportadas para o arquivo em disco com a extensão .script. O arquivo de script é carregado quando o banco de dados é inicializado. A diferença entre tabelas em memória para as tabelas temporárias é que as primeiras são persistidas, enquanto que as demais são destruídas quando uma conexão é perdida.

Já as tabelas do tipo cached são criadas através do comando CREATE CACHED TABLE. Esse modelo de tabela é diferenciado, pois parte de seus índices ou dados são mantidos em memória, para obtenção de uma melhor performance. Trata-se de uma estrutura de dados mais indicada quando uma tabela armazena um grande volume de dados.

O planejamento dos tipos de tabelas deverá levar em consideração o volume de dados que irão armazenar. Para tabelas pequenas o ideal é se utilizar o modelo de memória. Enquanto que para grandes tabelas o modelo cached é mais indicado.

As tabelas do tipo texto utilizam um delimitador denominado CSV (Comma Separated Value) para delimitar o conteúdo do arquivo. Os arquivos neste formato são úteis para a importação de estruturas

definidas em outros banco de dados. Trata-se e um modelo de intercâmbio de informações entre tecnologias de banco de dados.

As ferramentas contidas no pacote HSQLDB.JAR

Nessa seção, trataremos de uma ferramenta essencial para o gerenciamento de banco de dados. A ferramenta que iremos analisar é a Database Manager, cuja finalidade é prover ao desenvolvedor um front-end para definição do esquema do banco. A ferramenta também disponibiliza uma interface para interação e envio de instruções SQL via conexão JDBC. Para invocarmos o Database Manager, submetemos a seguinte linha de comando:

```
java org.hsqldb.util.DatabaseManager
      OU
java org.hsqldb.util.DatabaseManagerSwing
```

Note que o aplicativo está dentro do pacote útil e duas versões são disponibilizadas: uma versão utilizando uma interface baseada no conjunto de componentes gráficos do pacote AWT; e uma versão que utiliza componentes Swing. A figura 1 apresenta a interface de conexão da ferramenta com um banco de dados. A versão Swing foi utilizada para exemplificação.



Figura 1. Interface de conexão do Database Manager.

Na interface de conexão o usuário deverá informar o tipo de mecanismo de conexão: servidor, memória, standalone, ou outro; o driver de conexão, a URL, o usuário e a senha do mesmo.

A próxima figura, apresenta a interface principal da ferramenta. O Database Manager apresenta uma interface dividida em três áreas de informações. A primeira área, disponibilizada à esquerda da interface, apresenta uma estrutura hierárquica onde a estrutura do banco de dados poderá ser visualizada. Na estrutura o desenvolvedor poderá verificar as características de tabelas, campos, índices e restrições do banco. Além de outras propriedades do SGBD.

Uma área de edição de instruções SQL é apresentada para o usuário, ao lado da área de propriedades do banco. Nessa área o desenvolvedor irá escrever e submeter comandos SQL para manutenção de dados e definição do esquema do banco. Os botões apresentados abaixo do menu principal da ferramenta permitem que o usuário limpe a área de edição para escrita de novos comandos, ou submeta um determinado comando ao SGBD. Os resultados dos comandos de interação do usuário são apresentados em uma área abaixo da caixa de submissão de instruções SQL.

Outras funcionalidades podem ser exploradas através do menu principal da ferramenta. Vejamos algumas funções presentes no menu principal.

File: possui opções para abertura de scripts SQL gravados em arquivos texto, gravações de um conjunto de instruções em um arquivo em disco, salvar o resultado de uma consulta em disco e realização de conexões.

View: disponibiliza algumas opções para configuração da forma de apresentação de resultados, assim como, atualização da estrutura hierárquica que apresenta o esquema e propriedades de um banco de dados.

Command: essa opção de menu apresenta um conjunto de instruções SQL para auxílio ao usuário. Ao selecionar uma determinada instrução disponível no menu, o template já será apresentado ao usuário na caixa de comandos.

Recent: guarda uma lista das últimas instruções SQL submetidas pelo usuário através da caixa de comandos.

Options: mantém um conjunto de opções como controle de transações, ajuste de autocommit, login e logoff do banco, etc.

Tools: permite que o desenvolvedor possa acessar outras ferramentas a partir do Database Manager, tais como: ferramenta para exportação do esquema do banco, restauração de um esquema exportado e ferramenta para transferência de informações.

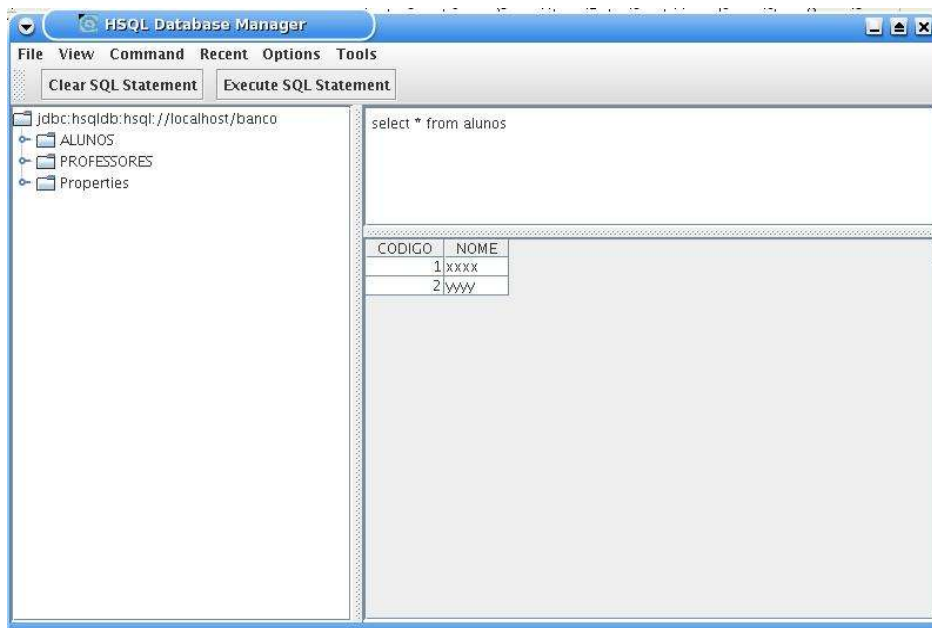


Figura 2. Interface principal do Database Manager.

Considerações finais

O RDBMS HSQLDB é uma ferramenta muito flexível para manipulação de banco de dados. Trata-se de uma excelente ferramenta para quem trabalha com Java e para construção de pequenas aplicações standalone. Em disciplinas voltadas a área de desenvolvimento de software é um grande achado, pois o software cabe em um disquete, tornando-se muito prático. Para quem trabalha com educação, trata-se de uma excelente opção para o ensino de conectividade de banco de dados com JDBC.

Carlos Emilio Padilla Severo (emilio.severo@gmail.com) é professor universitário, trabalha na FACSUL-CESUR lecionando disciplinas de desenvolvimento de software e coordenando o núcleo de tecnologia da informação da instituição.