

Armazenamento Nativo

Vanessa Braganholo

Como armazenar?

1. Sistema de arquivos
2. Banco de Dados Relacional/Objeto Relacional/etc. com suporte a XML (habilitado a XML ou híbrido)
3. Banco de Dados Nativo



Banco de Dados Nativo

- ▶ SGBDs que armazenam XML em sua forma nativa, geralmente como texto indexado ou como uma variante do DOM mapeado para uma estrutura proprietária





SGBDs XML Nativos





SGBDs XML Nativos



Um SGBD nativo:

- ▶ Define um modelo lógico para um documento XML e armazena e recupera documentos de acordo com este modelo
 - ▶ No mínimo, o modelo deve incluir elementos, atributos, PCDATA e a ordem dos elementos
 - ▶ Exemplos de modelos: modelo de dados do XPath, modelo DOM, etc.

Um SGBD nativo:

- ▶ Possui um documento como unidade fundamental de armazenamento lógico
 - ▶ Paralelo com tuplas dos bancos relacionais
- ▶ Não é exigido que exista nenhum modelo de armazenamento físico em particular
 - ▶ Exemplo: pode ser construído sobre um banco relacional, hierárquico, OO ou usar um formato proprietário qualquer (ex. arquivos compactados indexados)

Pontos principais

- ▶ Um SGBD nativo é um banco de dados especializado para armazenar dados XML
 - ▶ Armazena todos os componentes do modelo XML (elementos, atributos, etc.)
- ▶ *Documents go in and documents come out*
- ▶ Um SGBD nativo pode não ser um banco de dados isolado (pode ter um outro SGBD por “baixo dos panos”)

Alguns SGBDs nativos...

Berkley DB XML	Developer: Sleepycat Software URL: http://www.sleepycat.com/products/bdbxml.html Database type: Key-value
eXist	Developer: Wolfgang Meier URL: http://exist.sourceforge.net Database type: Proprietary
Tamino	Developer: Software AG (Germany), Consist (Brasil) URL: http://www.softwareag.com/tamino/ Database type: Proprietary. Relational data through ODBC
Timber	Developer: University of Michigan URL: http://www.eecs.umich.edu/db/timber Database type: Shore, Berkeley DB
XIndice	Developer: Apache Software Foundation URL: http://xml.apache.org/xindice Database type: Proprietary

Product	Developer	License	DB Type
4Suite, 4Suite Server	FourThought	Open Source	Object-oriented
BaseX	University of Konstanz	Open Source	Proprietary
Berkeley DB XML	Oracle	Open Source	Key-value
DBDOM	K. Ari Krupnikov	Open Source	Relational
dbXML	dbXML Group	Open Source	Proprietary
Dieselpoint	Dieselpoint, Inc.	Commercial	None (indexes only)
DOMSafeXML	Ellipsis	Commercial	File system(?)
EMC Documentum xDB	X-Hive Corporation	Commercial	Proprietary. Relational through JDBC
eXist	Wolfgang Meier	Open Source	Proprietary
eXtc	M/Gateway Developments Ltd.	Free	Post-relational
Extraway	3D Informatica	Commercial	Files plus indexes
Infonyte DB	Infonyte	Commercial	Proprietary
Ipedo XML Database	Ipedo	Commercial	Proprietary
Lore	Stanford University	Research	Semi-structured
MarkLogic Server	Mark Logic Corp.	Commercial	Proprietary
M/DB:X	M/Gateway Developments Ltd.	Free	Hierarchical
MonetDB/XQuery	CWI Database Group	Open Source	Proprietary
myXMLDB	Mladen Adamovic	Open Source	MySQL
Natix	University of Mannheim	Free / non-commercial	Proprietary
ozone	ozone-db.org	Open Source	Object-oriented
Qizx	XMLMind	Commercial	Proprietary
Sedna XML DBMS	ISP RAS MODIS	Free	Proprietary
Sekaiju / Yggdrasill	Media Fusion	Commercial	Proprietary
SQL/XML-IMDB	QuiLogic	Commercial	Proprietary (native XML and relational)
Sonic XML Server	Sonic Software	Commercial	Object-oriented (ObjectStore). Relational and other data through Data Junction
Tamino	Software AG	Commercial	Proprietary. Relational through ODBC.
TeraText DBS	TeraText Solutions	Commercial	Proprietary
TEXTML Server	IXIASOFT, Inc.	Commercial	Proprietary
TigerLogic XDMS	Raining Data	Commercial	Pick
Timber	University of Michigan	Open Source (non-commercial only)	Shore, Berkeley DB
TOTAL XML	Cincom	Commercial	Object-relational?
Virtuoso	OpenLink Software	Commercial	Proprietary. Relational through ODBC
XediX TeraSolution	AM2 Systems	Commercial	Proprietary
Xindice	Apache Software Foundation	Open Source	Proprietary
xml.gax.com	GAX Technologies	Commercial	Proprietary
Xpiori XMS	Xpiori	Commercial	Proprietary
XQuantum XML Database Server	Cognetic Systems	Commercial	Proprietary
XStreamDB Native XML Database	Bluestream Database Software Corp.	Commercial	Proprietary
Xyleme Zone Server	Xyleme SA	Commercial	Proprietary

Características de SGBDs nativos

- ▶ Armazena documentos
- ▶ Gerenciam “coleções” de documentos
- ▶ Suportam consultas XPath e XQuery
- ▶ Atualizações: linguagens próprias ou XQuery (XQuery Update Facility - <http://www.w3.org/TR/xquery-update-10/>)



Situação dos SGBDs Nativos

- ▶ Grande número de SGBDs nativos em oferta no mercado e na academia
- ▶ Armazenamento baseado em texto ou baseado em modelo proprietário
 - ▶ Ver discussão em <http://www.rpbouret.com/xml/XMLAndDatabases.htm#nativearchitecture>
- ▶ Lidam com grande volumes de documentos, alto *throughput*
- ▶ Grande eficiência das consultas
 - ▶ Consultas Full text



Alguns pontos que merecem consideração

- ▶ Nenhuma arquitetura padrão comum
- ▶ Nenhum padrão para benchmark de desempenho
 - ▶ Xmach – I (XML Data Management benchmark, September 2000, Timo Böhme, Erhard Rahm, University of Leipzig, Germany)
 - ▶ XBench (A Family of Benchmarks for XML DBMSs, Benjamin Bin Yao, M. Tamer Özsu, and John Keenleyside)
 - ▶ XMark (XMark: A Benchmark for XML Data Management, Albrecht Schmidt, Florian Waas, Martin L. Kersten, Michael J. Carey, Ioana Manolescu, Ralph Busse. VLDB 2002: 974-985)





Sedna

Sedna

- ▶ SGBD XML nativo, Open Source
 - ▶ <http://www.modis.ispras.ru/sedna/>
- ▶ Implementado em C
- ▶ Forma de armazenamento
 - ▶ Lista encadeada de blocos que armazenam nós XML
 - ▶ Índices são implementados usando Árvores B+
- ▶ Consultas: XQuery e XPath
- ▶ Atualizações: XQuery



Arquivos para Instalação

- ▶ **SGBD:**

- ▶ <http://modis.ispras.ru/sedna/download.html>
- ▶ Funciona via linha de comando

- ▶ **Interface de Administração:**

- ▶ Existem duas: uma oficial, outra da UFC
- ▶ Vamos usar a UFC, que funciona melhor
- ▶ <http://www.lia.ufc.br/~flavio/sednaadmin/>



Instalação

▶ SGBD

- ▶ Basta descompactar o arquivo
- ▶ Editar a variável de ambiente PATH, adicionando o diretório bin do Sedna

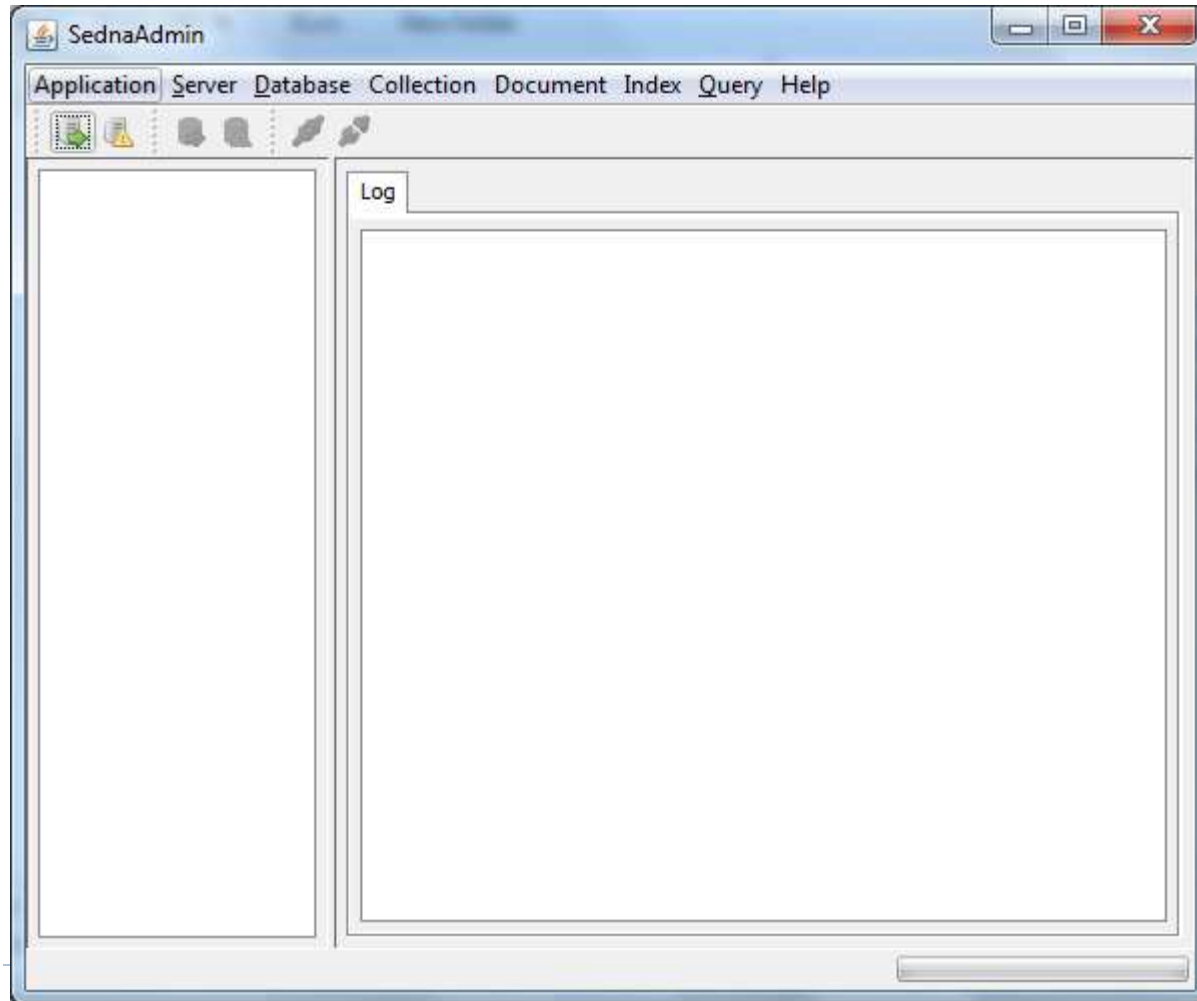
▶ Interface de Administração

- ▶ Basta descompactar o arquivo
- ▶ Informar qual o diretório raiz do Sedna



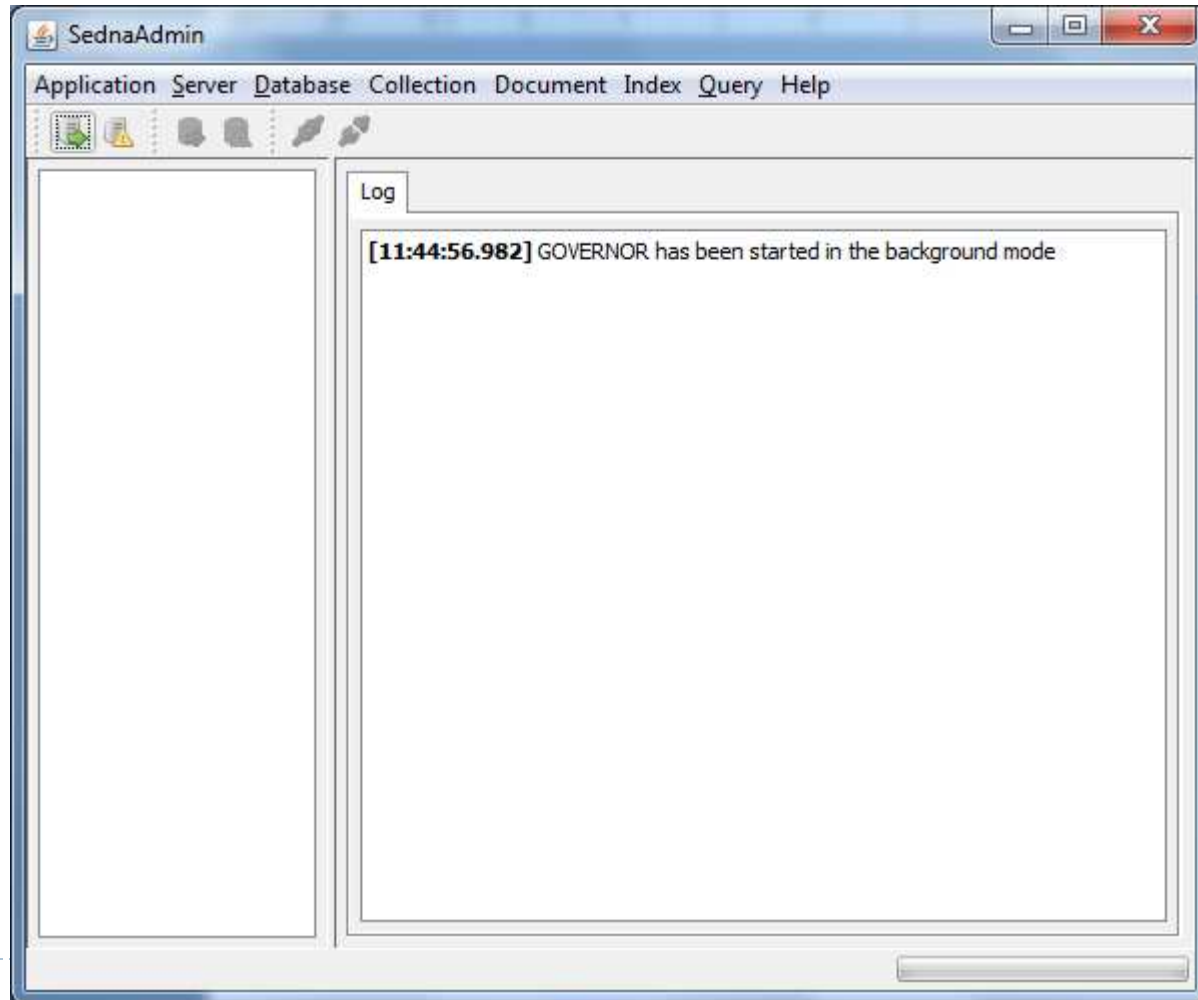
Para executar

- ▶ Iniciar a interface de Administração



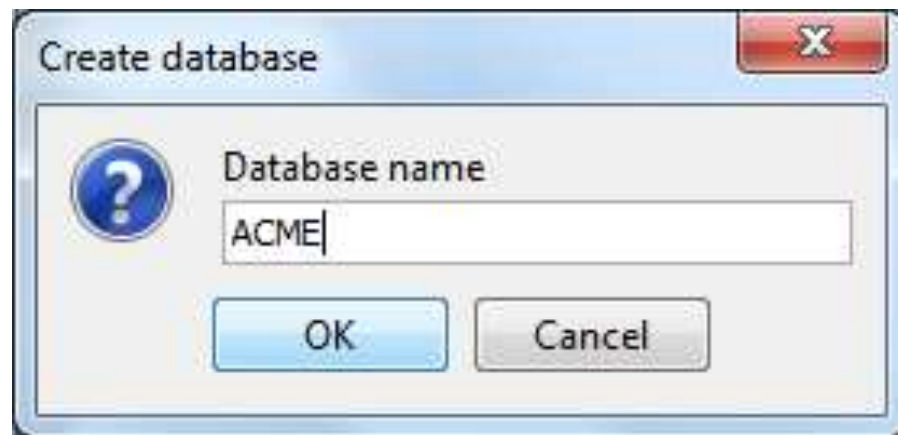
Iniciar o Servidor Sedna

▶ Menu Server/Start

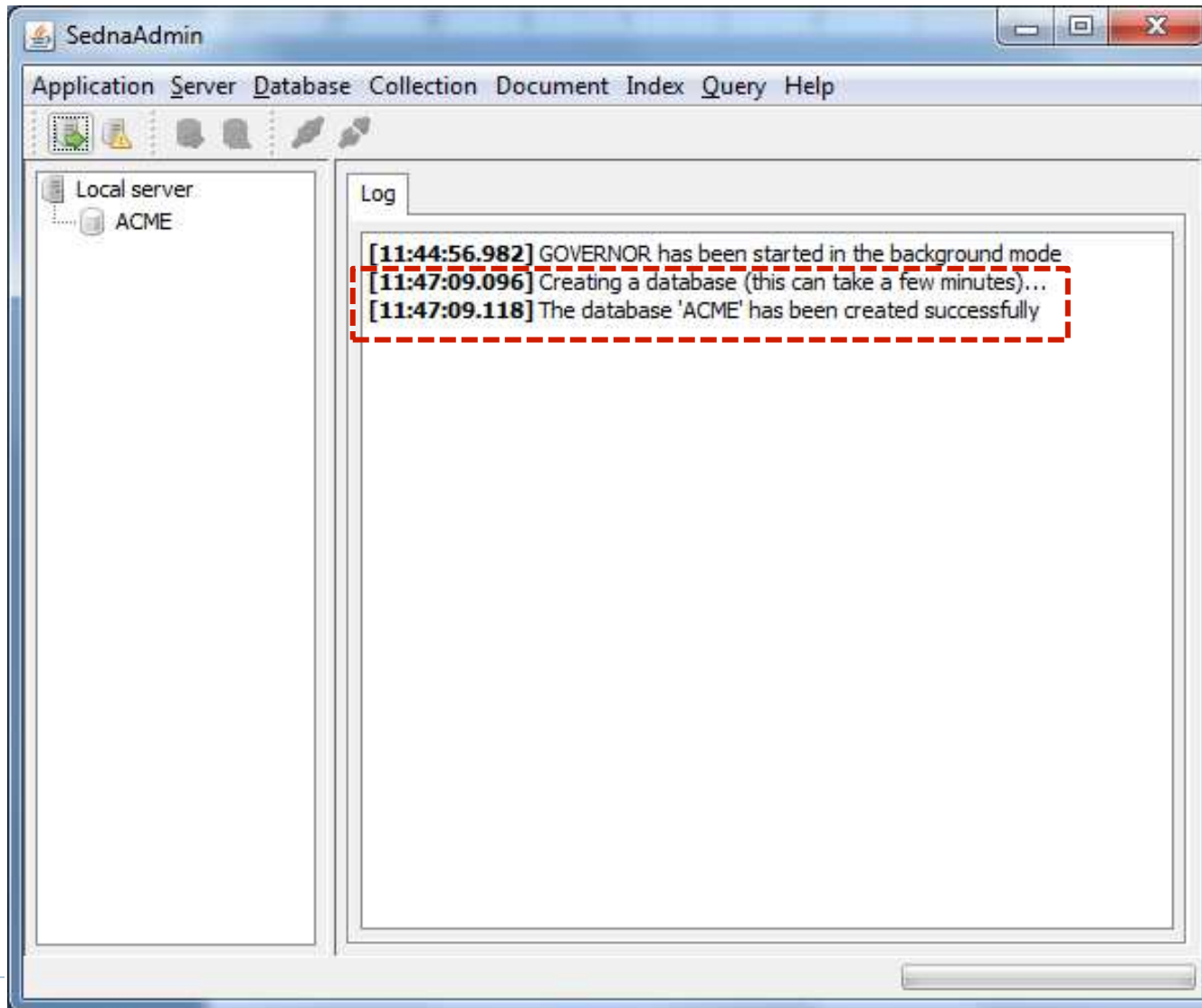


Criar um database

- ▶ Menu Database/Create
- ▶ Informar o nome do database:ACME

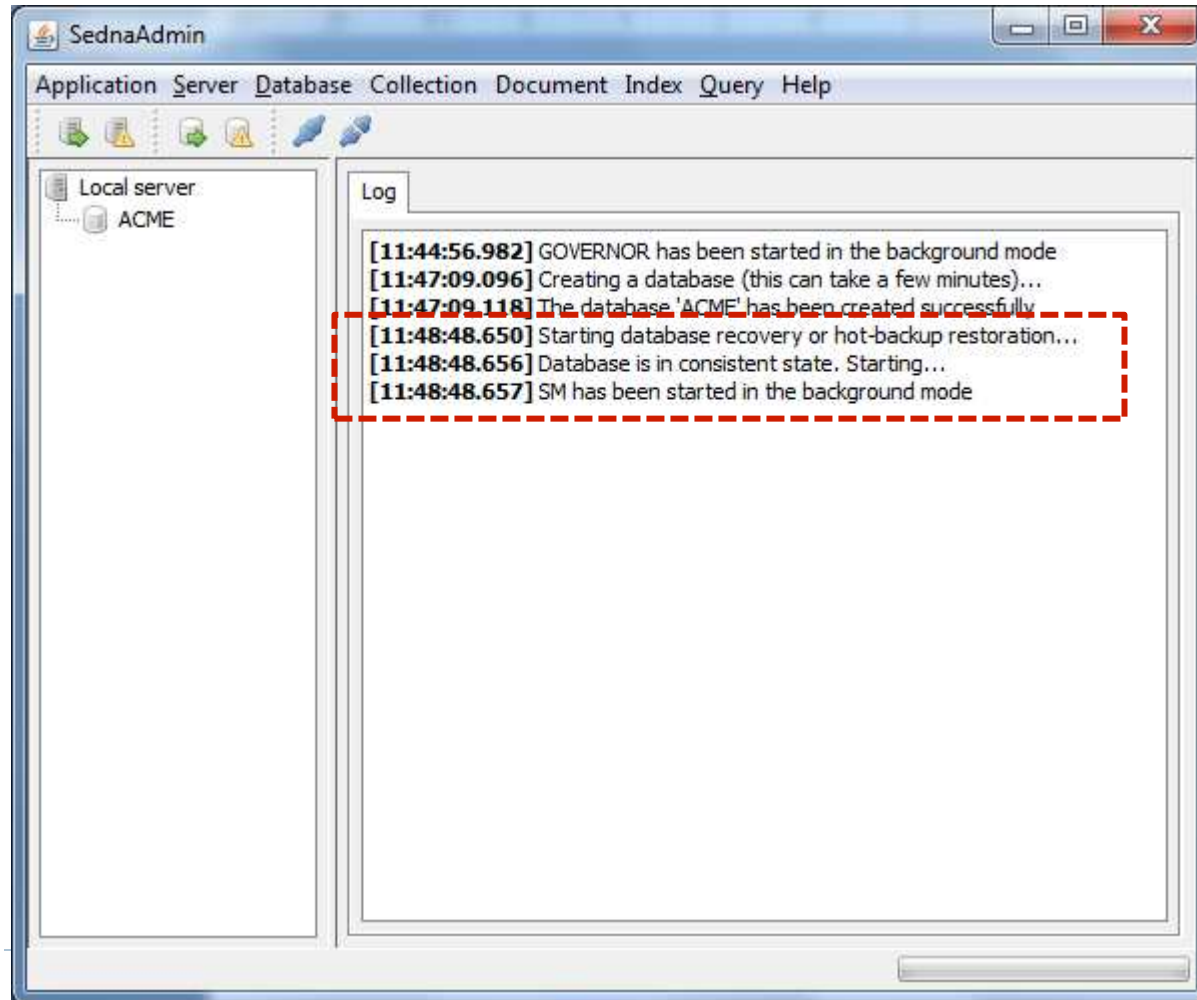


Criar um database



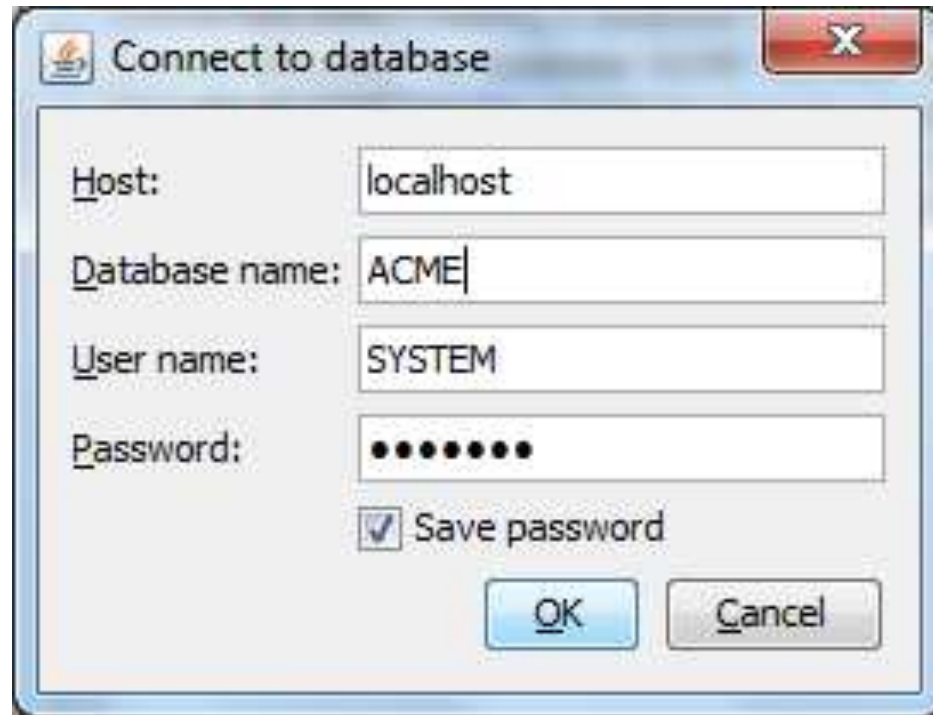
Iniciar o database

▶ Menu Database/Start



Conectar ao Database

- ▶ Menu Database/Connect



Connect to database

Host: localhost

Database name: ACME

User name: SYSTEM

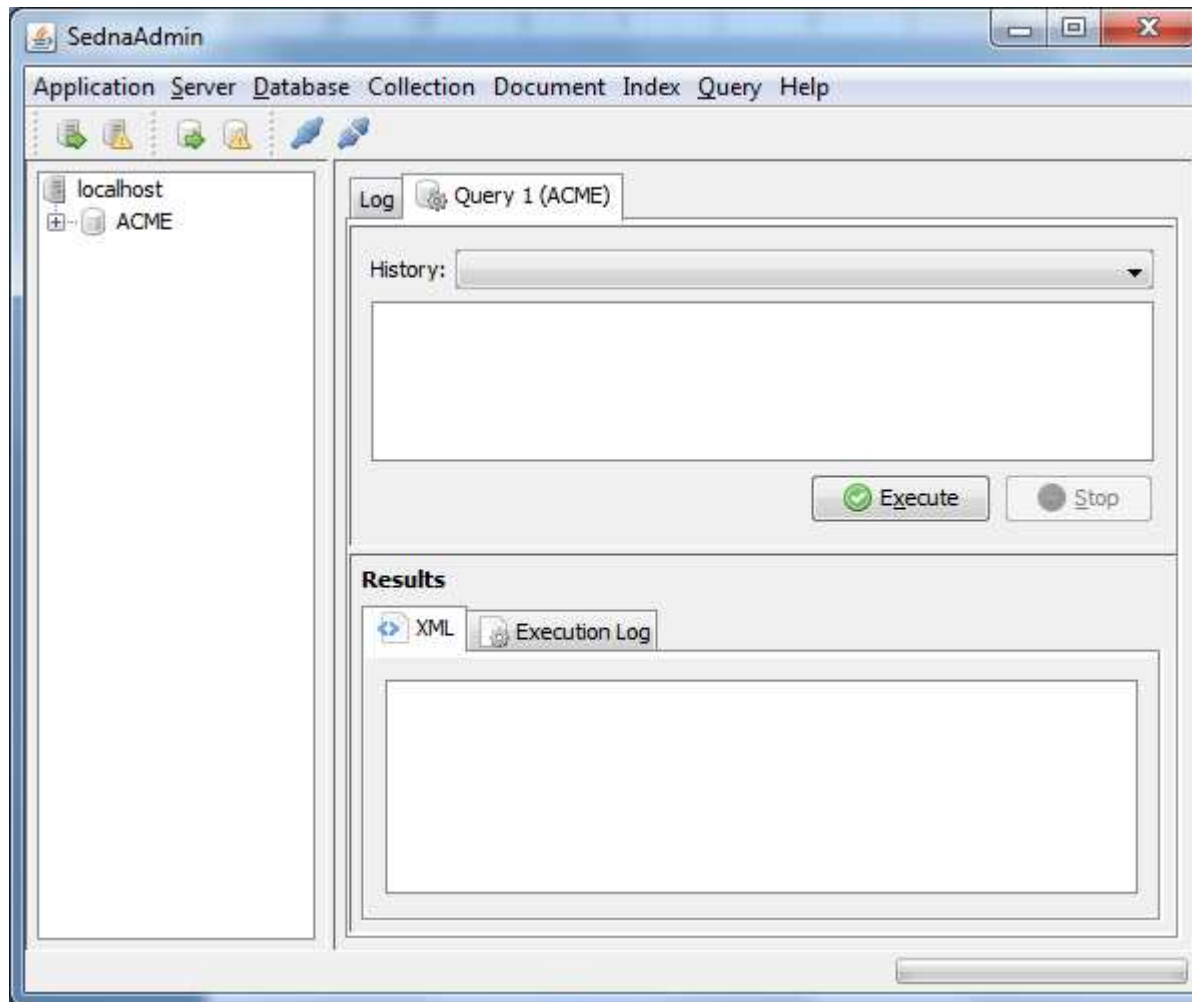
Password: ●●●●●●

Save password

OK Cancel

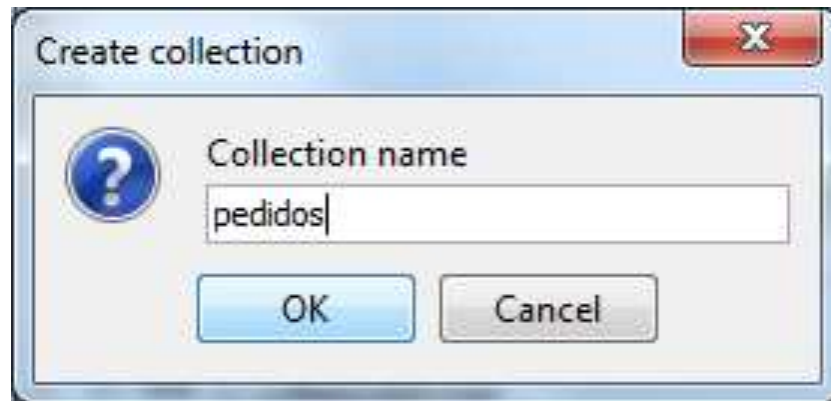


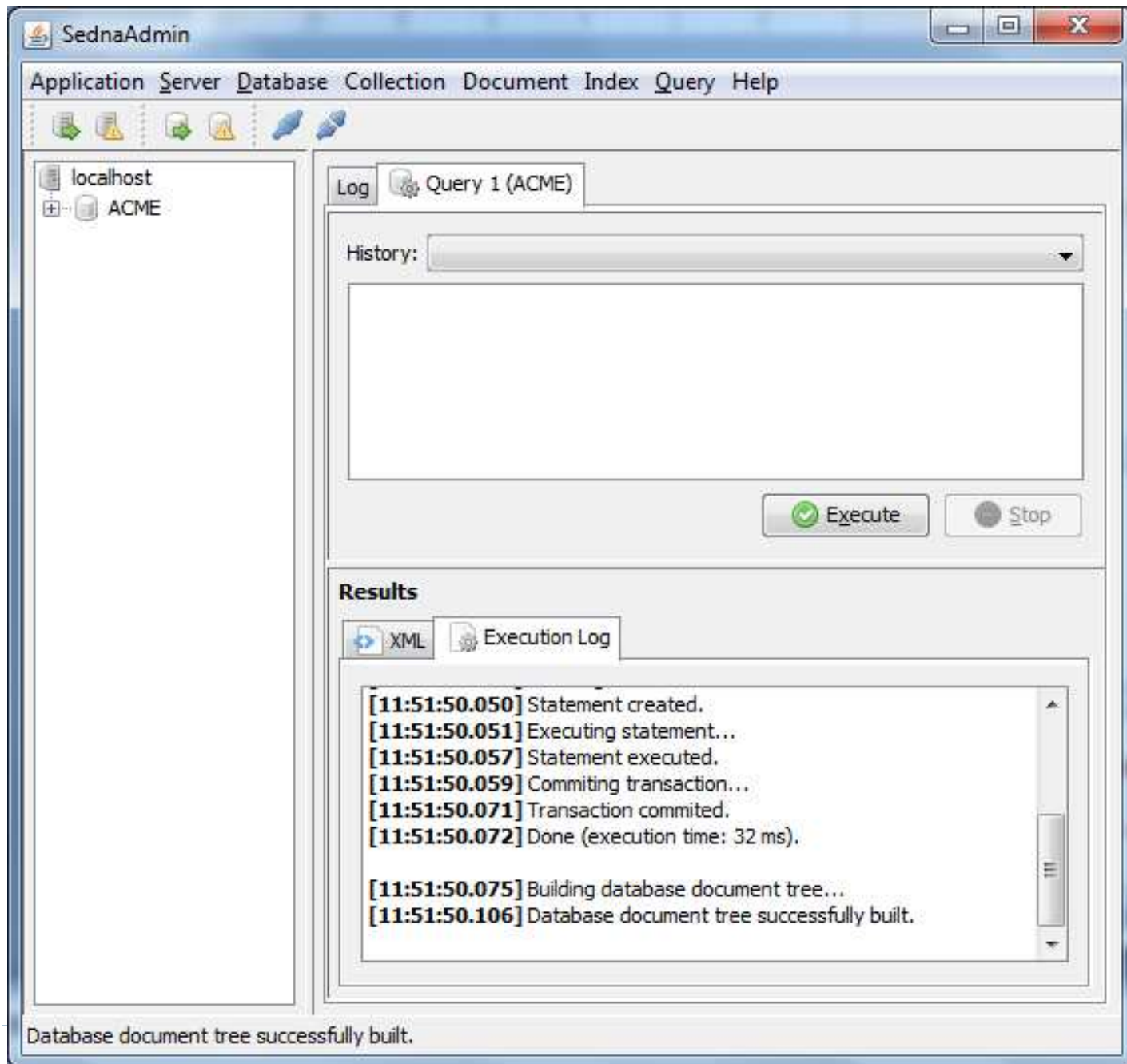
Conectar ao Database



Criar uma coleção

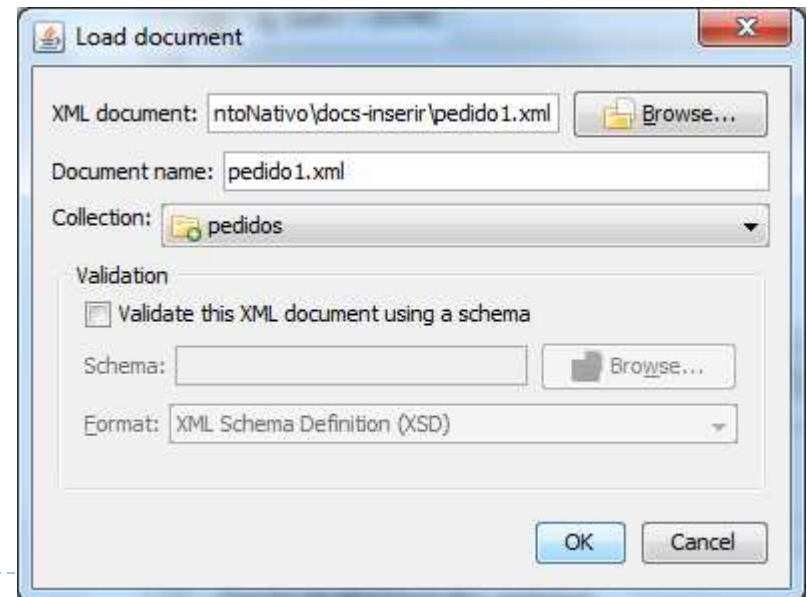
- ▶ Menu Collection/Create collection
- ▶ Digitar o nome da coleção
 - ▶ Vamos criar uma coleção chamada “pedidos” para armazenar os pedidos que são recebidos via Web pela empresa ACME





Agora vamos inserir documentos

- ▶ Baixar arquivo zip da página da disciplina que contém os documentos
- ▶ Menu Document/Load Document
- ▶ Selecione o documento pedido1.xml
- ▶ Verifique o nome que o documento terá (chave primária)
- ▶ Selecione o nome da coleção onde o documento deve ser inserido (pedidos)





- localhost
 - ACME
 - pedidos
 - pedido1.xml
 - Schema
 - \$modules
 - \$db_security_data

Log Query 1 (ACME)

History:

Results

```
[11:57:00.129] Executing statement...  
[11:57:00.129] Statement executed.  
[11:57:00.131] Committing transaction...  
[11:57:00.148] Transaction committed.  
[11:57:00.159] Done (execution time: 60 ms).  
  
[11:57:00.162] Building database document tree...  
[11:57:00.196] Database document tree successfully built.  
[11:57:00.198] Retrieving schemas...  
[11:57:00.278] Schemas successfully retrieved.
```

Schemas successfully retrieved.

Outra forma de inserir

- ▶ O Sedna possui uma funcionalidade que permite carregar vários documentos de uma única vez para uma coleção
- ▶ A funcionalidade exige que os documentos sejam especificados através de um comando **LOAD**
- ▶ O comando **LOAD** tem 3 parâmetros
 1. Nome do arquivo a ser carregado
 2. Chave que vai ser usada para identificar o documento
 3. Nome da coleção (opcional)
 4. Se houver o uso de mais de um comando **LOAD**, separá-los com **&**



Exemplo: arquivo load-data.xquery

```
LOAD "pedido2.xml" "pedido2.xml" "pedidos" &
```

```
LOAD "pedido3.xml" "pedido3.xml" "pedidos" &
```

```
LOAD "pedido4.xml" "pedido4.xml" "pedidos"
```



Carregamento de arquivos

- ▶ O carregamento deve ser feito via linha de comando
- ▶ A interface de administração ainda não permite isso
- ▶ **Digitar o comando**
 - ▶ `se_term -file load-data.xquery ACME`



Carregamento de arquivos

```
C:\Windows\system32\cmd.exe

15/11/2010 11:27 <DIR> .
15/11/2010 11:27 <DIR> i:
15/11/2010 11:58          135 load-data.xquery
15/11/2010 11:18    1.559 pedido.xsd
24/05/2006 12:42    481 pedido1.xml
15/11/2010 11:26    484 pedido2.xml
30/03/2004 14:07    613 pedido3.xml
30/03/2004 14:07    505 pedido4.dtd
15/11/2010 11:26    615 pedido4.xml
15/11/2010 11:15    596 pedidoComDTD.xml
15/11/2010 11:20    643 pedidoComERROSSchema.xml
15/11/2010 11:18    670 pedidoComSchema.xml
24/05/2006 13:50    1.082 xupdate.xml
          11 File(s)          7.383 bytes
          2 Dir(s) 77.907.456.000 bytes free

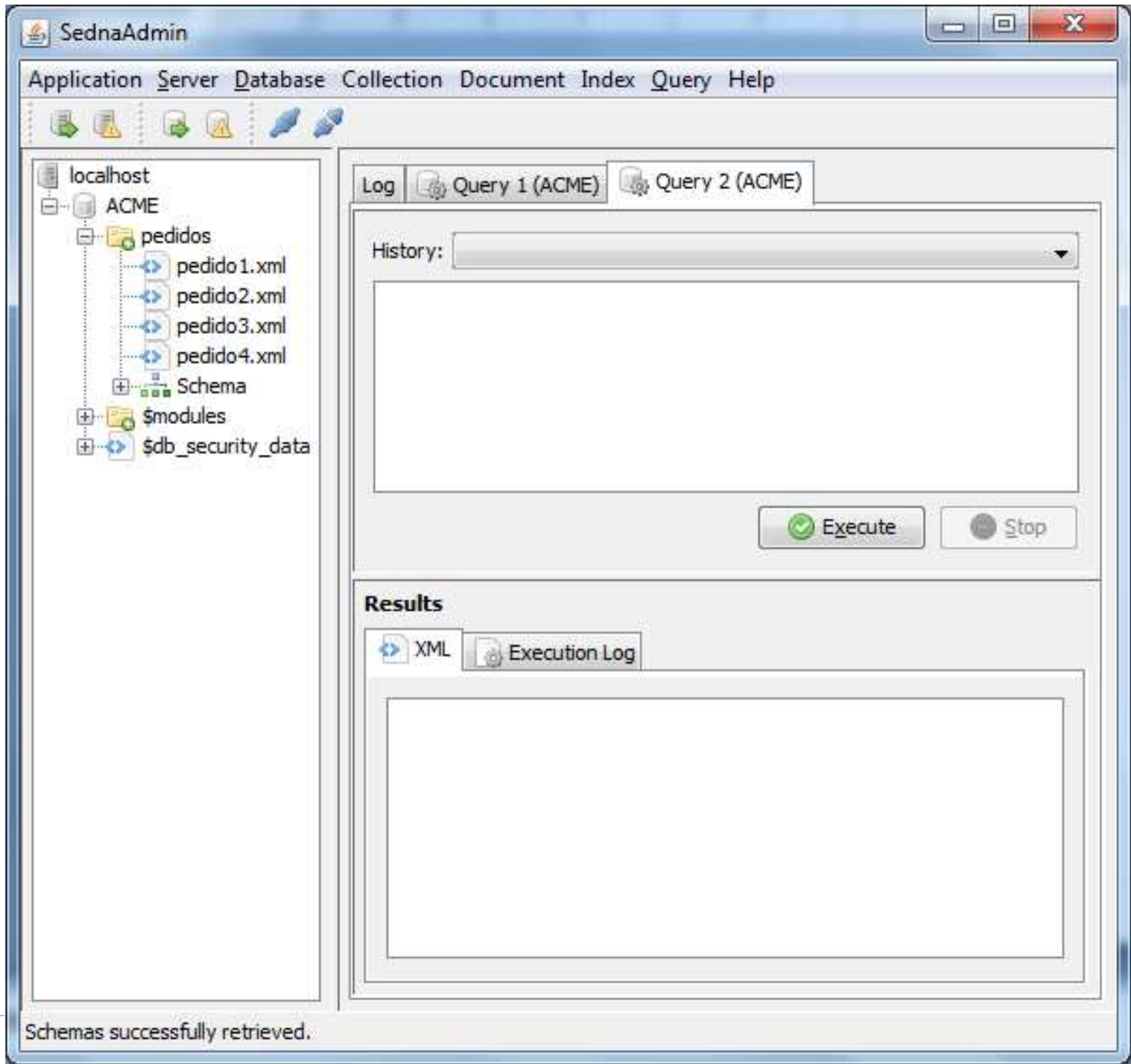
C:\Users\vanessa\Documents\Disciplinas\2010-2-XML\Aulas\06-ArmazenamentoNativo\d
ocs-inserir>se_term -file load-data.xquery ACME
Bulk load succeeded
Bulk load succeeded
Bulk load succeeded

C:\Users\vanessa\Documents\Disciplinas\2010-2-XML\Aulas\06-ArmazenamentoNativo\d
ocs-inserir>
```


Na interface de Administração

- ▶ Existe um problema de refresh
- ▶ É necessário conectar novamente ao database para ver os documentos que acabamos de inserir
- ▶ Menu Database/Connect

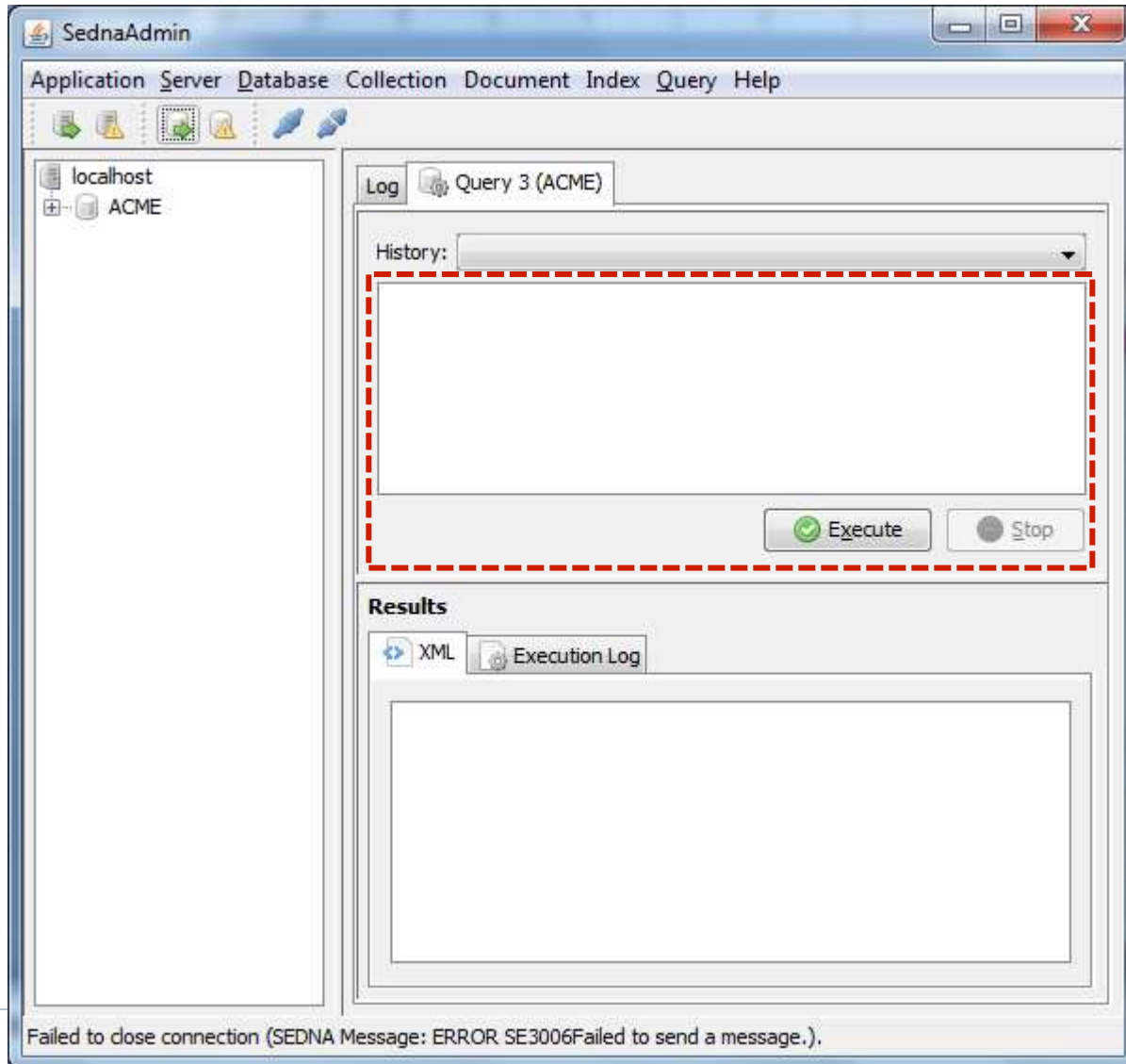




Schemas successfully retrieved.

Fazendo consultas XQuery...

Consultas



Para ver o conteúdo de um document

The screenshot shows the SednaAdmin application interface. On the left, a tree view shows the database structure under 'localhost' and 'ACME', with 'pedidos' expanded to show 'pedido1.xml' through 'pedido4.xml'. The main area displays a query in the 'History' field: `for $p in doc("pedido1.xml", "pedidos") return $p`. Below the query are 'Execute' and 'Stop' buttons. The 'Results' section shows the XML output for 'pedido1.xml':

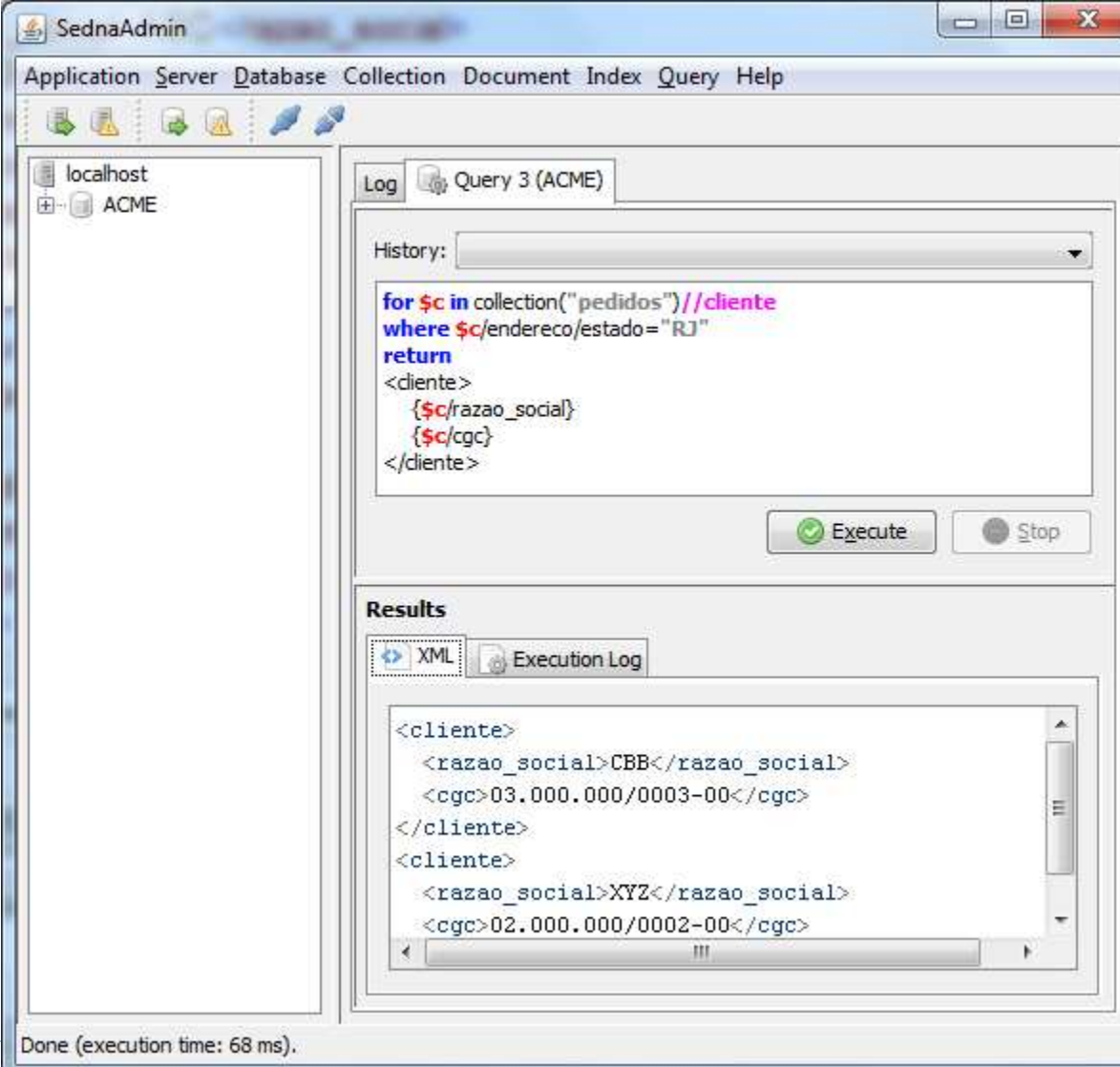
```
<?xml version="1.0" standalone="yes"?>
<pedido numero="888">
  <cliente>
    <razao_social>ABC</razao_social>
    <cgc>00.000.000/0001-00</cgc>
    <endereco>
      <logradouro>Rua das Flores,75 </logradouro>
```

At the bottom left, it says 'Done (execution time: 58 ms)'.

Segundo parâmetro da função doc é o nome da coleção

Para consultar uma coleção uso de **collection**

- ▶ Consultar os pedidos que foram feitos por clientes do estado RJ
- ▶ Retornar `razao_social` e `cgc`



The screenshot shows the SednaAdmin web interface. The left sidebar displays a tree view with 'localhost' and 'ACME'. The main area is titled 'Query 3 (ACME)' and contains a query editor with the following code:

```
for $c in collection("pedidos")//cliente
where $c/endereco/estado="RJ"
return
<cliente>
  {$c/razao_social}
  {$c/cgc}
</cliente>
```

Below the query editor are 'Execute' and 'Stop' buttons. The 'Results' section shows the XML output:

```
<cliente>
  <razao_social>CBB</razao_social>
  <cgc>03.000.000/0003-00</cgc>
</cliente>
<cliente>
  <razao_social>XYZ</razao_social>
  <cgc>02.000.000/0002-00</cgc>
```

At the bottom of the window, it says 'Done (execution time: 68 ms)'.

Exercícios

1. Faça uma consulta que retorna o elemento `itens_pedido` de todos os pedidos em que a razão social do cliente é “ABC”.
2. Faça uma consulta que retorna todos os produtos de todos os pedidos.
3. Crie mais 2 documentos XML e os insira na coleção pedidos. Repita as consultas anteriores.
4. Faça uma consulta que retorne todos os itens cujo preço é igual à quantidade comprada.
5. Faça uma consulta que retorna todos os itens que possuem “caneta” como parte do nome do produto.



Usar o Sedna em aplicações...

- ▶ **Developer Guide:** [ProgGuide.pdf](#) dentro do diretório docs do Sedna
- ▶ **Exercício:** fazer uma aplicação Java que:
 - ▶ Se conecta no Sedna
 - ▶ Cria uma coleção
 - ▶ Insere documentos na coleção
- ▶ **IMPORTANTE:** a senha do usuário SYSTEM é **MANAGER**



Ao terminar

- ▶ Database/Stop
- ▶ Server/Stop

