

5 Implementação de Um Protótipo

Neste capítulo é apresentada a implementação de um protótipo para armazenamento de objetos de aprendizado multimídia e para acesso, por meio de um navegador (*browser*), a uma visão global dos objetos de aprendizado distribuídos em dois locais de processamento de dados interligados.

5.1. Descrição do Problema

A crescente disseminação no uso do aprendizado eletrônico tem colocado o foco das atenções sobre os problemas envolvidos com a produtividade da elaboração de seu conteúdo.

Composto de elementos multimídia, além dos tradicionais textos, seu compartilhamento está sendo visualizado como uma alavanca para a aumentar a produtividade da elaboração de material de aprendizado eletrônico.

Considerando empresas e universidades produzindo textos, imagens, áudios e vídeos voltados para aprendizado eletrônico, as dificuldades em compartilhar este material despontam quando consideramos que o mesmo é armazenado usualmente em mídias avulsas (CD, disquetes, fitas) ou diretórios de discos rígidos dos computadores dos projetistas de aprendizado. São meios de armazenamento cujo objetivo está longe de ser o mais adequado quando consideradas as necessidades de compartilhamento. Estas dificuldades no compartilhamento de elementos de material de aprendizado são questões que esta dissertação tem por objetivo resolver.

Para o protótipo que será descrito neste capítulo foi considerada a situação exemplo.

Dois professores, que trabalham em dois sítios separados geograficamente, ao conversarem sobre seus trabalhos identificaram muita semelhança no material utilizado em seus cursos e vislumbraram a possibilidade de incrementá-lo com o compartilhamento de elementos. Como conclusão, os dois professores querem

integrar os dados de seus cursos, cada um em um site, e disponibilizar este material pela Web.

Para realizar esta idéia de compartilhar objetos de aprendizado vem a descrição do protótipo, cuja arquitetura está apresentada na **Figura 26**.

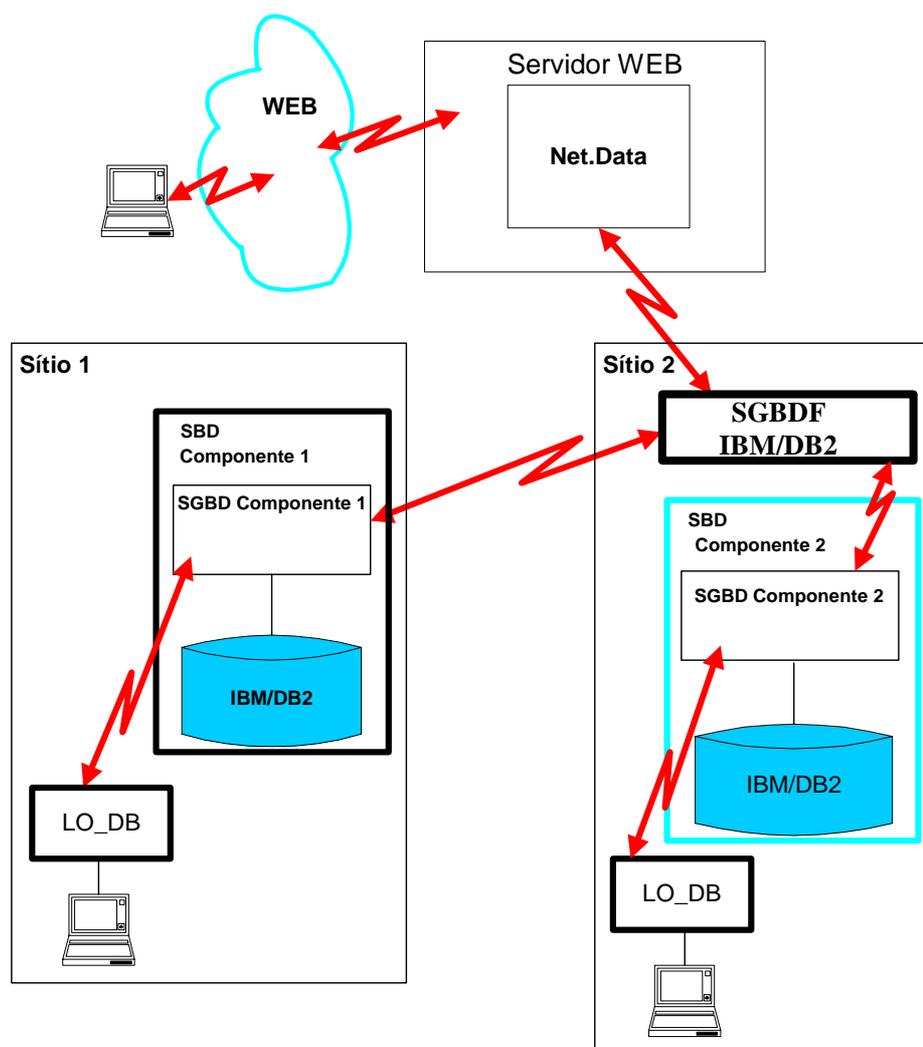


Figura 26. Arquitetura do Protótipo

Este protótipo teve por objetivo implementar uma aplicação para armazenamento de objetos de aprendizado em bancos de dados que estão distribuídos por dois sítios interligados. Cada sítio dispõe de um gerenciador de banco de dados autônomo e foram considerados requisitos para a aplicação:

- Armazenar os objetos de aprendizado contendo uma descrição em formato XML e dispor de arquivos de apoio de áudio, vídeo ou imagem;

- Possibilitar o acesso aos dados inseridos dentro da codificação XML;
- Permitir o acesso aos dados do tipo áudio, vídeo ou imagem;
- Permitir o acesso uniforme aos dados tornando transparente a sua distribuição física.

Não fizeram parte do protótipo:

- Elaborar material de aprendizado;
- Elaborar fórmulas de composição de objetos de aprendizado;
- Explorar objetos de aprendizado através das fórmulas de composição.

5.2. Sistema de Gerenciamento de Banco de Dados

O protótipo foi montado utilizando o gerenciador IBM DB2 instalado em dois servidores interligados por TCP/IP. A **Figura 27** apresenta o esboço do sistema de banco de dados utilizado.

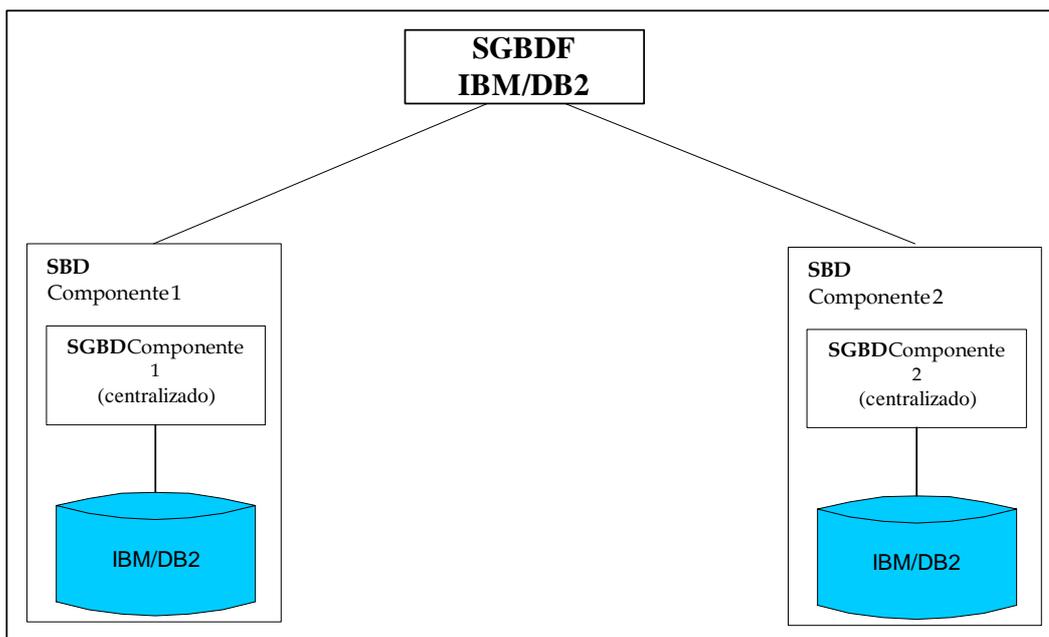


Figura 27. Configuração de Banco de Dados do Protótipo

5.3. Base de Dados Local

O esquema físico das bases de dados locais para armazenamentos de objetos de aprendizado está apresentado na **Figura 28**. O mesmo esquema foi utilizado no dois sites.

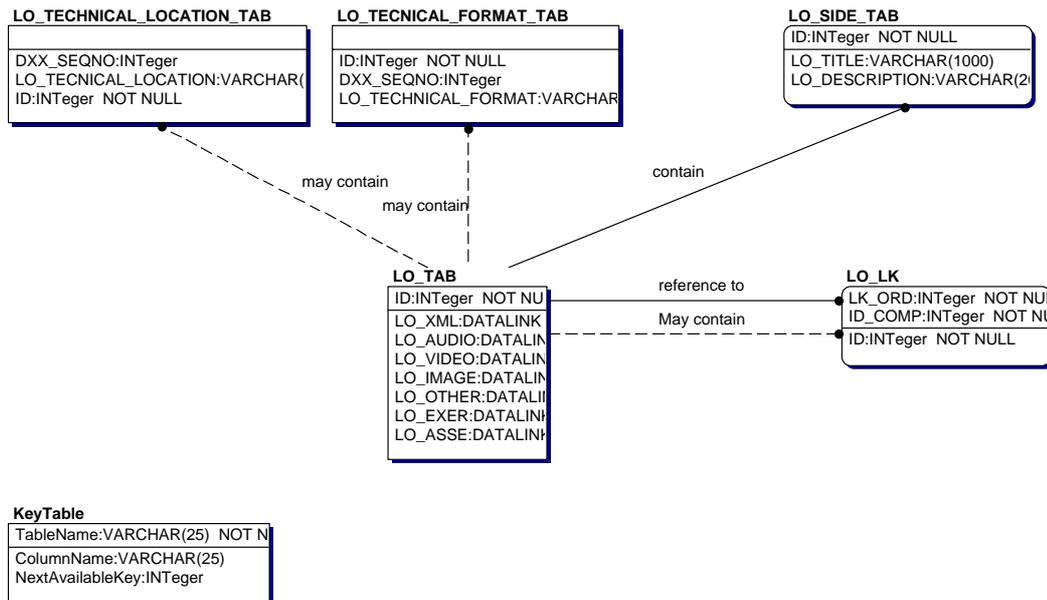


Figura 28. Esquema físico da base de dados

5.3.1. Tabela KeyTable

Utilizada como recurso para gerar um identificador interno para objetos de aprendizado.

| Atributo | Formato | Descrição |
|------------------|-------------|---|
| ColumnName | VARCHAR(25) | Coluna utilizada como índice nas tabela de referencia |
| NextAvailableKey | INTEGER | Valor do índice a ser utilizado |
| TableName | VARCHAR(25) | Tabela de referencia a ser indexada |

5.3.2. Tabela LO_TAB

Utilizada para manter as informações básicas sobre objetos de aprendizado. Observe-se que, por imposição do gerenciador de banco de dados IBM DB2, nesta entidade o atributo associado com o conteúdo do objeto de aprendizado foi decomposto em diversos atributos especializados por tipo de dado:

- áudio
- vídeo
- imagem
- outros

| Atributo | Formato | Descrição |
|----------|----------------|--|
| ID | VARCHAR(20) | Identificador único |
| LO_ASSE | BLOB | Arquivo com avaliação |
| LO_AUDIO | DB2AUDIO | Arquivo com conteúdo sonoro |
| LO_EXER | BLOB | Arquivo com exercícios |
| LO_IMAGE | DB2IMAGE | Arquivo com imagem |
| LO_OTHER | BLOB | Arquivo com conteúdo não áudio, vídeo ou imagem. |
| LO_VIDEO | DB2VIDEO | Arquivo com vídeo |
| LO_XML | DB2XML.XMLCLOB | Arquivo com descrição XML |

5.3.3. Tabela LO_LK

Utilizada para conter os relacionamentos entre objetos de aprendizado.

| Atributo | Formato | Descrição |
|----------|---------|---|
| ID | INTEGER | Identificador do objeto de aprendizado principal |
| ID_COMP | INTEGER | Identificador do objeto de aprendizado componente |
| LK_ORD | INTEGER | Ordem de execução do objeto componente |

5.3.4. Redundância de Dados

As seguintes informações foram implementadas ao nível de projeto físico do banco de dados, visando otimizar o resultado de consultas. Esta implementação foi realizada por meio de tabelas paralelas, contendo as informações existentes na codificação dos objetos de aprendizado:

- TITULO: Identifica o título atribuído ao objeto de aprendizado;
- DESCRIÇÃO: conteúdo descrevendo o objeto de aprendizado;
- Arquivos técnicos associados e seus formatos, referindo-se exclusivamente aos arquivos de apoio contidos na descrição XML do objeto de aprendizado.

Esta implementação é obtida através do arquivo DAD descrito a seguir e esta materializada nas tabelas:

- LO_SIDE_TAB
- LO_TECHNICAL_FORMAT_TAB
- LO_TECHNICAL_LOCATION_TAB

5.3.4.1. Arquivo DAD

Os dados referentes à descrição do objeto de aprendizado em XML serão armazenados em uma única coluna da tabela.

Para fins de desempenho, evitando a constante extração baseada em rótulos são criadas tabelas paralelas, administradas pelo gerenciador de banco de dados e atualizadas concomitantemente com os dados.

Um exemplo de definição de tabela vinculada a um rótulo é apresentado a seguir, e uma listagem completa do arquivo utilizado no protótipo consta do item **Apêndices – 10.2. Listagem do Arquivo DAD.**

```

<table name="LO_side_tab">
  <column name="LO_title"
    type="varchar(1000)"
    path="/lom/general/title/langstring"
    multi_occurrence="NO"/>
  <column name="LO_description"
    type="varchar(2000)"
    path="/lom/general/description/langstring"
    multi_occurrence="NO"/>
</table>

```

5.3.4.2. Tabela LO_SIDE_TAB

Utilizada como recurso para melhorar o desempenho de pesquisas por título dos objetos de aprendizado.

Após a implementação caberá ao gerenciador a sua atualização.

| Atributo | Formato | Descrição |
|----------------|---------------|---|
| ID | INTEGER | Identificador do objeto de aprendizado |
| LO_DESCRIPTION | VARCHAR(2000) | Extrato referente ao rótulo <DESCRIPTION> |
| LO_TITLE | VARCHAR(1000) | Extrato referente ao rótulo <TITLE> |

5.3.4.3. Tabela LO_TECHNICAL_FORMAT_TAB

Utilizada como recurso para melhorar o desempenho de pesquisas por formato dos objetos de aprendizado. Como o campo em foco pode ter múltiplas ocorrências torna-se necessária à alocação de uma nova entidade.

Após a implementação caberá ao gerenciador a sua atualização.

| Atributo | Formato | Descrição |
|---------------------|--------------|--|
| ID | INTEGER | Identificador do objeto de aprendizado |
| DXX_SEQNO | INTEGER | Identificador de ocorrência |
| LO_TECHNICAL_FORMAT | VARCHAR(500) | Extrato referente ao rótulo <FORMAT> |

5.3.4.4. Tabela LO_TECHNICAL_LOCATION_TAB

Utilizada como recurso para melhorar o desempenho de pesquisas sobre a localização dos objetos de aprendizado. Como o campo em foco pode ter múltiplas ocorrências torna-se necessária à alocação de uma nova entidade.

Após a implementação caberá ao gerenciador a sua atualização.

| Atributo | Formato | Descrição |
|-----------------------|--------------|--|
| ID | INTEGER | Identificador do objeto de aprendizado |
| DXX_SEQNO | INTEGER | Identificador de ocorrência |
| LO_TECHNICAL_LOCATION | VARCHAR(500) | Extrato referente ao rótulo <LOCATION> |

5.4. Aplicação LO_DB

Para carga dos dados no banco de dados foi utilizada uma versão da aplicação LO_DB desenvolvida especificamente para atualizar no gerenciador de banco de dados IBM DB2 dados não estruturados como XML, áudio, vídeo e imagem.

Foi utilizada a mesma aplicação para carregar os dados de cada site. Um resumo do material final carregado é apresentado no item **5.7. Resumo do Processo de Carga.**

As telas exemplo apresentadas ao longo do documento não necessariamente correspondem ao apresentado no item resumo, tendo em vista diferentes utilizações e experimentos realizados ao longo dos trabalhos.

A descrição detalhada desta aplicação está apresentada no item **Apêndices - 10.3. Documentação da Aplicação LO_DB.**

A versão desenvolvida possui as seguintes funções:

- Carga de Informações Básicas
- Carga de Componentes Associados
- Lista Títulos Cadastrados

Antes de iniciar o processo de carregar os dados sobre objetos de aprendizado no banco de dados alguns passos devem ser executados, tendo em vista limitações dos produtos empregados.

Algumas ferramentas de autoria geram a descrição XML do objeto de aprendizado produzido.

Tendo em vista limitações da extensão XML, conforme a versão do IBM DB2, as descrições XML dos objetos de aprendizado devem ser ajustadas.

O primeiro passo visa retirar recursos não suportados por versões do IBM DB2. Por exemplo, usando a versão 7.2 do extensor, deverão ser retirados do início da descrição:

- a) *xmlns*: declaração de namespace
- b) *xmlns:xsi*: declaração de instância de nomes de atributos padronizados e da localização física do esquema de nomes de atributos

Abaixo está um exemplo como produzido pela ferramenta de autoria:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v3.5 (http://www.xmlspy.com) (NETg) -->
<lom xmlns="http://www.imsglobal.org/xsd/imsmd_v1p2"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://www.imsglobal.org/xsd/imsmd_v1p2 imsmv1p2p2.xsd">
  <general>
```

Ao eliminar a referência externa para validar os dados descritivos deve ser indicado o endereço do arquivo DTD onde será feita a validação:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v3.5 (http://www.xmlspy.com) by Boyd W Nielsen (NETg) -->
<!DOCTYPE lom SYSTEM "h:\dxx\dtd\ims.dtd">
<lom>
  <general>
```

Permanece em vigor assim, o recurso para validar as descrições de objetos de aprendizado segundo o padrão IMS. Será feito automaticamente através do gerenciador consultando o arquivo padrão de definição (DTD).

Como comentado, a versão utilizada neste protótipo não permitiu o uso de *namespace*, que possibilitaria apontar diretamente para o endereço eletrônico http://www.imsglobal.org/xsd/imsmd_v1p2.

Foi criado no servidor um arquivo com a descrição IMS e as adaptações necessárias.

Um exemplo deste arquivo é apresentado no item **Apêndices – 10.1. Listagem do Arquivo DTD.**

Para carregar no banco de dados as informações de um objeto de aprendizado, foi implementado um processo que é realizado em duas etapas:

1. Carga de informações básicas;
2. Carga da expressão de composição.

As informações básicas compreendem:

- Sua definição em XML
- Arquivo de conteúdo genérico
- Arquivo de apoio em áudio (opcional)
- Arquivo de apoio em vídeo (opcional)
- Arquivo de apoio em imagem (opcional)
- Arquivo de exercícios (opcional)
- Arquivo de avaliação (opcional)

A carga dos dados básicos é executada através da tela apresentada na **Figura 29**. Ao ser executada com sucesso a operação atribui um identificador único para o objeto de aprendizado incluído. Este identificador será utilizado nas futuras referências.

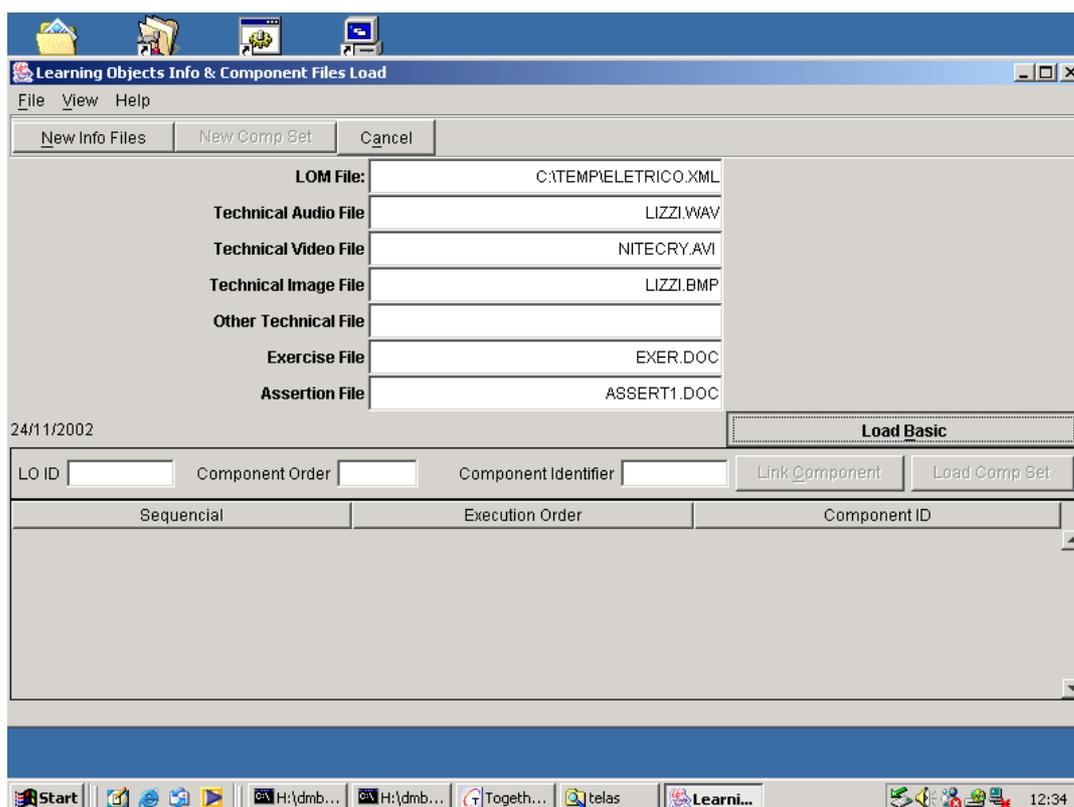


Figura 29. Tela de dados básicos

O preenchimento de ligações entre objetos de aprendizado é feito em dois passos utilizando-se a mesma tela, na sua parte inferior, conforme **Figura 30**.

Inicialmente é incluído o conjunto de componentes associados a cada objeto de aprendizado, bem como a informação quanto a sua precedência de execução dentro do conjunto.

São consideradas as seguintes suposições para esta tarefa:

- a) A identificação utilizada para os componentes será aquela gerada internamente pelo sistema;
- b) A definição de precedência será efetuada via entrada de dados.

A entrada de dados de composição pode ser iniciada após a ativação da aplicação ou em seqüência a entrada de dados básicos.

Outra forma de iniciar o cadastramento de dados de composição é em continuidade à entrada de dados básicos.

Com a montagem do conjunto bem sucedida executa-se o comando de armazenar este conjunto na base de dados.

No exemplo apresentado na **Figura 30** está sendo armazenada a seguinte expressão:

$$LO_{77} = \{LO_{12}, LO_{15}\}$$

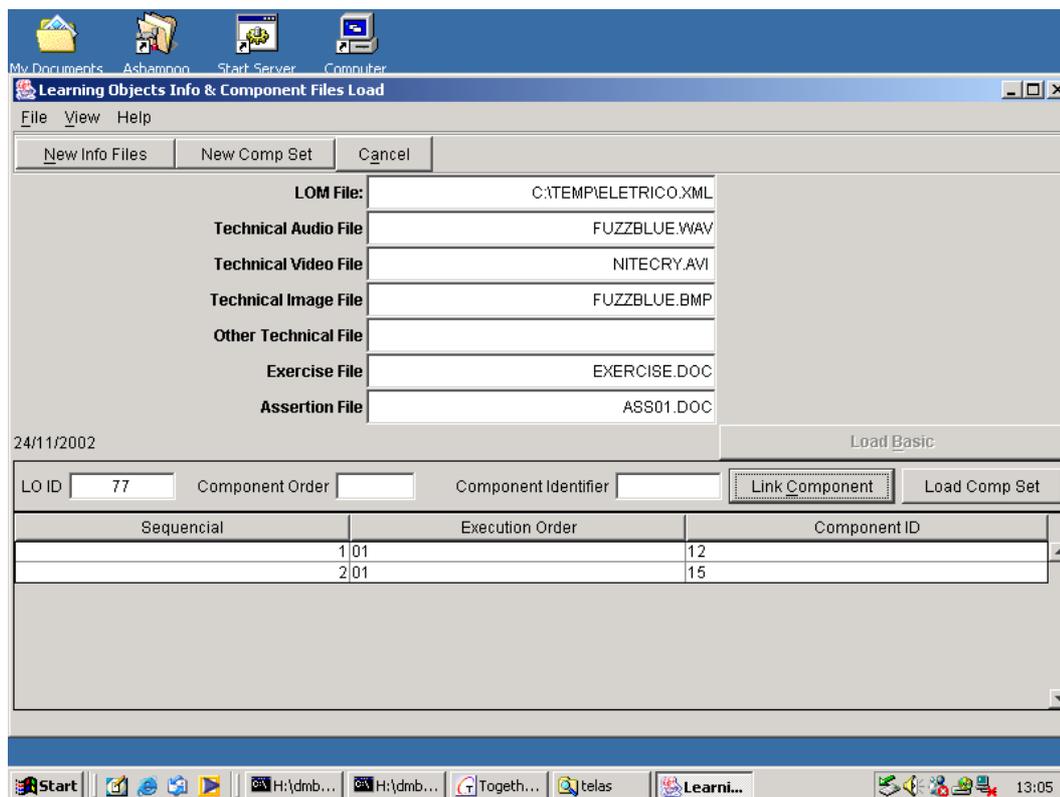


Figura 30. Tela de dados de composição

Onde foram inseridos os seguintes dados:

| Sequencial | Ordem de Execução | Identificador do Componente |
|------------|-------------------|-----------------------------|
| 1 | 01 | 12 |
| 2 | 01 | 15 |

Os demais exemplos apresentados no item **3.1 – Modelo de Objetos de Aprendizado** teriam o seguinte preenchimento:

$$LO_{32} = \{LO_{10}, LO_{11}\}$$

O objeto de aprendizagem LO_{32} é composto pelos objetos LO_{10} e LO_{11} explorados nesta ordem. O preenchimento seria:

| Sequencial | Ordem de Execução | Identificador do Componente |
|------------|-------------------|-----------------------------|
| 1 | 01 | 10 |
| 2 | 02 | 11 |

$$LO_{90} = \{LO_{88}, LO_{31}, LO_5\}$$

Este exemplo de composição tem a seguinte interpretação:

LO_{90} é composto por três elementos não ordenados. Foi assumido que dois seriam compostos: LO_{88} e LO_{31} e LO_5 seria objeto elementar. O preenchimento para LO_{90} será:

| Sequencial | Ordem de Execução | Identificador do Componente |
|------------|-------------------|-----------------------------|
| 1 | 01 | 88 |
| 2 | 01 | 31 |
| 3 | 01 | 5 |

LO_{88} é um conjunto ordenado composto por LO_1 , LO_2 e LO_3 e será preenchido:

| Sequencial | Ordem de Execução | Identificador do Componente |
|------------|-------------------|-----------------------------|
| 1 | 01 | 1 |
| 2 | 02 | 2 |
| 3 | 03 | 3 |

LO_{31} é um conjunto não ordenado composto por dois objetos elementares: LO_{41} e LO_{43} e um elemento composto LO_{87} . Logo o preenchimento de LO_{31} será:

| Sequencial | Ordem de Execução | Identificador do Componente |
|------------|-------------------|-----------------------------|
| 1 | 01 | 41 |
| 2 | 01 | 43 |
| 3 | 01 | 87 |

LO_{87} é composto pelo conjunto ordenado LO_{421} e LO_{422}

| Sequencial | Ordem de Execução | Identificador do Componente |
|------------|-------------------|-----------------------------|
| 1 | 01 | 421 |
| 2 | 02 | 422 |

Finalmente a terceira funcionalidade disponível na aplicação LO_DB permite obter a relação de todos os títulos cadastrados no servidor local. Um exemplo desta tela é apresentado na **Figura 31**.

| ID | LO_TITLE |
|---------|--|
| 77 | Microsoft SQL Server 7.0: Implementing a Database - Part 1 |
| 80 | Microsoft SQL Server 7.0: Implementing a Database - Part 2 |
| 81 | Microsoft SQL Server 7.0: Implementing a Database - Part 3 |
| 83 | Microsoft SQL Server 7.0: Implementing a Database - Part 4 |
| 84 | Generating Column Values |
| 85 | Tracing Column Values Update |
| 86 | Microsoft SQL Server 7.0: Performance and Tunning |
| 87 | Microsoft SQL Server 7.0: Performance and Tunning I |
| 88 | Microsoft SQL Server 7.0: Performance and Tunning II |
| 89 | Physlet Problems: Circuits II |
| 93 | Human Resource Management |
| 95 | Human Resource Management I |
| 96 | Physlet Problems: Circuits III |
| 97 | Physlet Problems: Circuits IV |
| 98 | Physlet Problems: Circuits V |
| 99 | Physlet Problems: Circuits VI |
| 100 | Physlet Problems: Circuits VII |
| 101 | Physlet Problems: Circuits VIII |
| 102 | Resumes and Interviews |
| 103 | Resumes and Interviews I |
| 104 | Resumes and Interviews II |
| 105 | Resumes and Interviews III |
| 106 | Resumes and Interviews IV |
| 107 | Resumes and Interviews V |
| 110 | Human Resource Management II |
| 111 | Human Resource Management III |
| 112 | Human Resource Management IV |
| SUMMARY | 27 |

Figura 31. Lista de Títulos Cadastrados

5.5. Criando Uma Base de Dados Federada no IBM DB2

Conforme apresentado na **Figura 26** e detalhado na **Figura 27** a configuração proposta compreende dois gerenciadores de bancos de dados locais e um gerenciador de banco de dados federado.

Nos passos anteriores foi descrita a implementação local a cada sítio de sua base de dados e a carga dos dados. Neste item serão apresentados os passos para implementar uma base de dados federada.

Esta base de dados será implementada utilizando os recursos disponíveis do gerenciador de banco de dados IBM DB2, não sendo necessária nenhuma adaptação.

Sua localização será no servidor KURIZKY, por facilidade nas questões de segurança relativas ao acesso.

A seguinte série de comandos foi executada:

a) Registrar, no servidor KURIZKY, o servidor URCA, via painel Centro de Controle, como sendo mais um sistema gerenciado remotamente:

| Parâmetro | Valor |
|-----------|----------|
| Hostname | URCA |
| Port | db2c_DB2 |

b) Registrar a base LO_DB remota, via *Configuration Assistant*, adição de base de dados:

| Parâmetro | | Valor |
|-----------------|------------------|---|
| Source set up | | <i>Manually</i> |
| Catalog options | | <i>Add database to your local machine</i> |
| Protocolo | | <i>TCP/IP</i> |
| TCP/IP | Host name | <i>139.82.120.59</i> |
| | Port number | <i>51164</i> |
| | Service name | <i>db2c_DB2</i> |
| Dbname | | <i>LO_DB</i> |
| Node options | Operating system | <i>Windows</i> |
| | Remote instance | <i>DB2</i> |
| System options | System name | <i>URCA</i> |

c) Criar, na base local o objeto federado relativo à base remota, via Centro de Controle:

- Incluir o invólucro;
- Incluir mapeamento de usuários
- Criar apelidos para tabelas, utilizando a seguinte convenção: *URCA.remote_name*

Tendo concluído estas tarefas é possível verificar o acesso transparente à nova base e tabelas selecionadas, efetuando-se a conexão à base local e submetendo comandos de consulta à base remota através da utilização dos apelidos.

5.5.1. Conversão de tipos de dados

Para utilizar a funcionalidade de suporte de sistema federado do IBM DB2 é necessário efetuar o mapeamento dos dados e das funções especializados, definidos localmente, para recursos regulares do gerenciador. Para tanto, são necessárias as seguintes tarefas básicas:

1. Em cada base componente de dados definir visões mapeando os recursos específicos para recursos regulares;
2. Na base de dados federada realizar as seguintes tarefas:
 - a. Definir sobrecarga de funções específicas utilizando dados convencionais em lugar de dados não convencionais;
 - b. Definir os nodos correspondentes aos endereços que serão interligados;
 - c. Definir *wrappers* para cada tipo de fonte de dados local a ser acoplado;
 - d. Definir as transformações de códigos de acesso entre o gerenciador federado e os gerenciadores a serem acoplados;
 - e. Definir os apelidos para as tabelas e visões a serem acopladas.

A **Tabela 6** apresenta a conversão de tipos de dados específicos criados pelos extensores XML (esquema *DB2XML*) e AVI (esquema *MMDBSYS*) do IBM DB2 para tipos de dados básicos (esquema *SYSIBM*). Esta tabela é baseada em informações armazenadas na visão *SYSCAT.DATATYPES*, armazenada internamente pelo IBM DB2.

Outros tipos de dados especializados (UDT) a serem explorados no nível da federação também deverão ser objeto de mapeamento. Estes dados podem ser facilmente identificados na seção *User Defined Distinct Datatypes* ao visualizar as informações sobre uma base de dados no painel Centro de Controle, conforme apresentado na **Figura 32**.

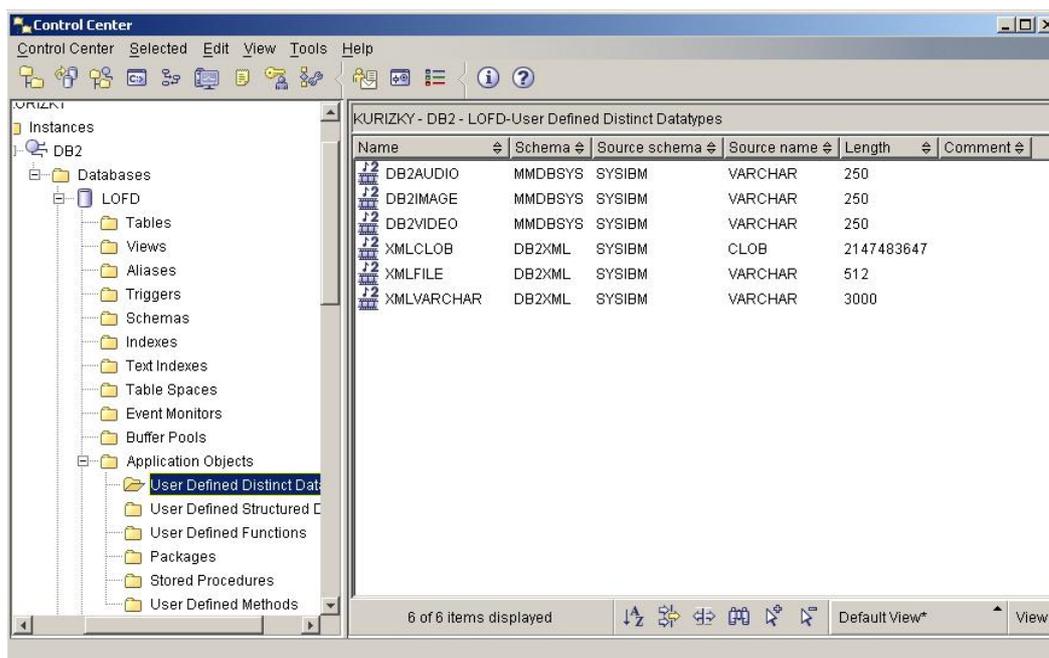


Figura 32. Tipos de dados não básicos

O conteúdo da **Tabela 4** deverá ser complementado com tipos de dados utilizados por outros gerenciadores ou outros tipos de dados especializados.

| Tipo de Dado Não Convencional | Tipo de Dado Básico |
|-------------------------------|---------------------|
| DB2VIDEO | VARCHAR (250) |
| DB2IMAGE | VARCHAR (250) |
| DB2AUDIO | VARCHAR (250) |
| XMLFILE | VARCHAR (512) |
| XMLCLOB | CLOB (2147483647) |
| XMLVARCHAR | VARCHAR (3000) |

Tabela 4. - Conversão para tipos de dados básicos

5.5.2. Adição de Uma Fonte de Dados DB2

Os passos descritos a seguir são executados via linha de comando do IBM DB2:

- a) Catalogando o nodo no diretório da federação:
 - a. Deve ser determinado o protocolo de comunicação a ser utilizado;
 - b. Executar o seguinte comando supondo que o protocolo, por exemplo, que seja TCP/IP:

```
CATALOG TCPIP NODE DB2NODE REMOTE URCA SERVER db2c_DB2
```

Onde:

DB2NODE é o nome a ser associado ao nodo remoto;

URCA é o nome local do servidor onde a fonte de dados reside;

Db2c_DB2 é o nome do serviço que gerencia a instância.

b) Catalogando a base de dados remota no sistema federado:

```
CATALOG DATABASE LO_DB AS LO_DB_EI AT NODE DB2NODE AUTHENTICATION  
SERVER
```

LO_DB é o nome remoto da base de dados;

LO_DB_EI é o apelido como a base externa será conhecida na federação;

c) Registrando o invólucro:

```
CREATE WRAPPER DRDA
```

d) Registrando as definições do servidor de dados:

```
CREATE SERVER PUC TYPE DB2/NT VERSION 7.1 WRAPPER DRDA  
AUTHORIZATION "db2admin" PASSWORD "pass" OPTIONS (NODE 'URCA', DBNAME  
'LO_DB')
```

e) Criando o mapeamento de usuários:

Este comando é necessário para permitir que uma conexão feita através da base de dados federada seja aceita na base de dados fonte. Seu objetivo é estabelecer uma associação entre usuários catalogados na base federada de dados e usuários catalogados nas fontes de dados.

```
CREATE USER MAPPING FOR db2admin SERVER URCA OPTIONS (REMOTE_AUTHID  
'db2admin', REMOTE_PASSWORD 'pass')
```

f) Testando a conexão:

```
SET PASSTHRU PUC  
SELECT count(*) FROM sysibm.systables  
SET PASSTHRU RESET
```

g) Registrando os apelidos para tabelas e visões:

Esta tarefa visa permitir que o catálogo federado acumule estatísticas que serão utilizadas posteriormente no otimizador de consultas

```
CREATE NICKNAME URCA_LO_SIDE_TAB FOR PUC.DB2ADMIN.LO_DB.LO_SIDE_TAB
```

h) Acessando os dados remotos:

Tendo concluído os passos acima é possível executar o comando de acesso aos dados como se os mesmos estivessem locais, sendo transparente qualquer diferença de localização:

```
SELECT * FROM LO_REL
```

5.5.3. Ajustes no Componente

Para conseguir que dados não convencionais, utilizados pelos extensores XML e AIV, possam ser explorados ao nível da federação foi definida uma visão mapeando-os nos equivalentes dos dados básicos do IBM DB2. Esta visão com dados básicos pode então ser exportada para a federação.

Foi mapeada a entidade que armazena as informações básicas sobre objetos de aprendizado.

| Atributo | Formato | Descrição |
|----------|---------|--|
| ID | INTEGER | Identificador único |
| LO_ASSE | BLOB | Arquivo sobre capacitação |
| LO_AUDIO | VARCHAR | Arquivo com conteúdo sonoro |
| LO_EXER | BLOB | Arquivo sobre exercícios |
| LO_IMAGE | VARCHAR | Arquivo com imagem |
| LO_OTHER | BLOB | Arquivo com conteúdo não áudio, vídeo ou imagem. |
| LO_VIDEO | VARCHAR | Arquivo com filme |
| LO_XML | CLOB | Arquivo com descrição XML |

O comando abaixo foi executado no servidor componente para criar uma visão utilizando tipos de dados primários:

```
CREATE VIEW X_LO_TAB (ID, LO_XML, LO_AUDIO, LO_VIDEO, LO_IMAGE) AS
SELECT ID, CAST(LO_XML AS CLOB(2147483647)), CAST(LO_AUDIO AS VARCHAR(250)),
CAST(LO_VIDEO AS VARCHAR(250)), CAST(LO_IMAGE AS VARCHAR(250))
FROM LO_TAB
```

5.5.4. Ajustes no Servidor Federado

Para que possam ser utilizadas com toda a sua capacidade as funcionalidades do IBM DB2 precisam ser adaptadas.

5.5.4.1. Sobrecarga das Funções de Usuário

Conforme é realizada a conversão de tipos de dados específicos para dados básicos é necessário também redefinir as funções que utilizem parâmetros de tipos específicos para funções similares utilizando tipos básicos.

Um exemplo deste carregamento é apresentado na **Figura 33**, onde está definida a função *ExtractVarchar*, utilizada para extrair uma cadeia de caracteres de um documento XML. Esta função na sua versão original utiliza como parâmetros de entrada um campo do tipo *xmlclob* e foi redefinida para utilizar um tipo *clob*.

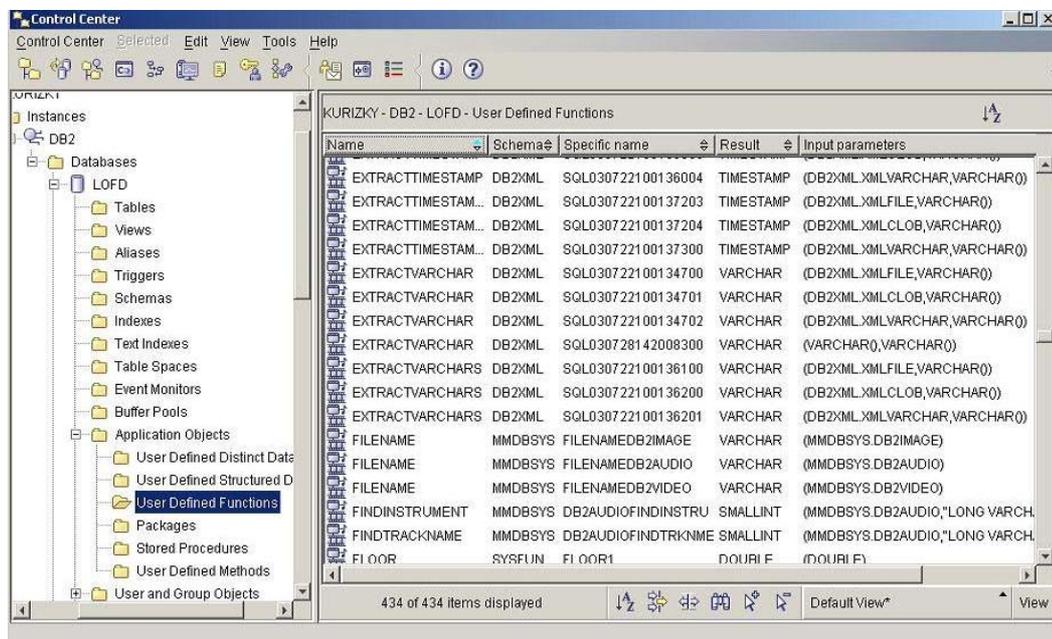


Figura 33. Funções de Usuário

Para redefinir funções utilizando novos tipos de dados foram utilizados os comandos abaixo exemplificados:

```

CREATE FUNCTION
  DB2XML.EXTRACTVARCHAR (VARCHAR(), VARCHAR())
  RETURNS VARCHAR ()
SOURCE
  DB2XML.EXTRACTVARCHAR (DB2XML.XMLVARCHAR, VARCHAR ())

```

Outras funções também foram redefinidas:

```

CREATE FUNCTION MMDBSYS.THUMBNAIL_V (VARCHAR())
  RETURNS LONG VARCHAR
SOURCE MMDBSYS.THUMBNAIL (MMDBSYS.DB2VIDEO)
CREATE FUNCTION MMDBSYS.THUMBNAIL_I (VARCHAR())
  RETURNS LONG VARCHAR
SOURCE MMDBSYS.THUMBNAIL (MMDBSYS.DBIMAGE)
CREATE FUNCTION MMDBSYS.CONTENT_I (VARCHAR(), VARCHAR ())
  RETURNS BLOB
SOURCE MMDBSYS.CONTENT (MMDBSYS.DB2IMAGE, VARCHAR())
CREATE FUNCTION MMDBSYS.FORMAT_I (VARCHAR())
  RETURNS LONG VARCHAR ()
SOURCE MMDBSYS.FORMAT (MMDBSYS.DB2IMAGE)

```

Neste caso foram criadas novas funções uma vez que as originais já utilizavam sobrecarga para tratar as diferenças entre os tipos de dados áudio, vídeo e imagem.

Após a sobrecarga das funções é possível verificar o correto funcionamento executando o comando abaixo, no servidor federado:

```

SELECT id, db2xml.extractvarchar(lo_xml, '/lom/general/title/langstring')
  FROM x_lo_tab
SELECTt id, mmdbsys.format_i(lo_image)
  FROM x_lo_tab

```

As funções abaixo foram transformadas apenas para a versão 7:

```

CREATE FUNCTION mmdbsys.comment_i (varchar())
  RETURNS long varchar
SOURCE mmdbsys.comment(mmdbsys.db2image)
CREATE FUNCTION mmdbsys.comment_a (varchar())
  RETURNS long varchar
SOURCE mmdbsys.comment(mmdbsys.db2audio)
CREATE FUNCTION mmdbsys.comment_v (varchar())
  RETURNS long varchar
SOURCE mmdbsys.comment(mmdbsys.db2video)

```

5.5.4.2. Visão Federada

Uma visão que acessa dados cujas tabelas estejam localizadas em fontes remotas é denominada visão federada. Ao criar uma visão federada as tabelas componentes são referenciadas pelos apelidos e não pelos seus nomes originais.

Quando uma visão federada é composta por mais de uma fonte de dados é permitida apenas a leitura dos dados remotos, ao passo que sendo a visão sobre uma fonte única de dados são permitidas também atualizações de dados.

Para criar uma visão federada uma sintaxe especial (*federated view*) é utilizada, conforme o exemplo de comando apresentado a seguir:

```
CREATE federated view lo_db_vf AS
  SELECT * FROM lo_side_tab
  UNION
  SELECT * FROM urca_lo_side_tab
```

Este comando tem por objetivo criar uma visão que disponibilize para a união dos dados da relação local **LO_SIDE_TAB** com a relação remota **URCA_LO_SIDE_TAB**. O nome **URCA_LO_SIDE_TAB** é o apelido criado para a relação remota **PUC.DB2ADMIN.LO_DB.LO_SIDE_TAB**.

Embora o exemplo apresente a união de todos os atributos de duas relações, a visão federada pode ser composta de outras formas, selecionando diferentes atributos de uma ou mais relações.

5.6. Navegação nos Dados

A exploração dos dados armazenados no IBM DB2, através de navegador, é possível através da utilização de uma codificação em macro que é submetida à camada de produto denominada Net.Data.

Um exemplo da codificação para o Net.Data é apresentado no item **Apêndices – 10.4. Exemplo de Macro Net.Data**.

A exploração aos dados, por browser, é possível com o seguinte comando:

```
//139.82.120.59/cgi-bin/db2www.exe/lo_db.d2w/StartHere
```

ou

```
//localhost:8080/cgi-bin/db2www.exe/lo_db.d2w/StartHere
```

Este comando inicia, dentro da macro apresentada no item **Apêndices –**

10.4. Exemplo de Macro Net.Data, o bloco HTML denominado *startHere*.

```
%function (DTW_SQL) startHereSQL(){
select id, lo_title, lo_description from db2admin.fv_lo_side_tab
%REPORT{
<table border="2" bgcolor="#b1b1b1">
<tr><th>Id <th> Title <th> Description </tr>
%ROW{ <tr><td><a href="addThumbs?id=$(V_id)">$(V_id)</a> <td> $(V_lo_title)
<td> $(V_lo_description) </tr>
% }
</table>
% }
```

Este bloco, por sua vez, executa a função *startHereSQL* que efetivamente executa o comando de leitura de uma visão federada e como resultado do conjunto é apresentada a tela da **Figura 34**.

| Id | Title | Description |
|---------------------|--|--|
| 125 | Microsoft SQL Server 7.0: Implementing a Database - Part 1 | This is the first course in a five part series that will provide students with the knowledge to implement a database solution with Microsoft SQL Server 7.0. |
| 126 | Physlet Problems: Circuits II | Physlet problems that relate to electric circuits. |
| 127 | Resumes and Interviews V | Description: An interview with Lorna Court, Personnel Consultant, on how to impress employers with your resume. Court gives practical advice on what to put in your resume in order to secure an interview. |
| 128 | Tracing Column Values Update | For each table, a single identifier column can be created that contains system-generated sequential values that uniquely identify each row within the table. |
| 129 | Preassessment Unit 3 | Preassessment Instructions |
| 130 | Physlet Problems: Circuits | Physlet problems that relate to electric circuits. |
| 131 | Actin-myosin crossbridge animation | The actin-myosin crossbridge system is complex, and we are really only speculating on the details in many ways. However, if a picture is worth a thousand words, this one second, 15 frame, animation is worth at least 15 thousand. |
| 225 | Microsoft SQL Server 7.0: Implementing a Database - Part 1 | This is the first course in a five part series that will provide students with the knowledge to implement a database solution with Microsoft SQL Server 7.0. |
| 325 | Microsoft SQL Server 7.0: Implementing a Database - Part 1 | This is the first course in a five part series that will provide students with the knowledge to implement a database solution with Microsoft SQL Server 7.0. |
| 326 | Microsoft SQL Server 7.0: Implementing a Database - Part 2 | This is the SECOND course in a five part series that will provide students with the knowledge to implement a database solution with Microsoft SQL Server 7.0. |

Figura 34. Tela Lista de Objetos de Aprendizado

Ao clicar sobre um identificador de um objeto de aprendizado é visualizada a tela dos detalhes deste objeto (**Figura 35**), armazenados em XML segundo o padrão LOM, bem como a indicação de existência de dados AVI associados. Isto é resultado o processamento do bloco HTML *addThumbs*. Este bloco efetua a chamada da função *addThumbsSQL* que recupera de uma outra visão o conjunto de dados associado ao identificador selecionado.

```
%function (DTW_SQL) addThumbsSQL(){
select
db2xml.extractVarchar (LO_XML,'/lom/general/title/langstring'),
db2xml.extractVarchar (LO_XML,'/lom/general/description/langstring') ,
db2xml.extractVarchar (LO_XML,'/lom/rights/description/langstring') ,
cast(mmdbsys.thumbnail_i(lo_image) as blob(10000)),
cast(mmdbsys.thumbnail_v(lo_video) as blob(10000)),
mmdbsys.comment(lo_audio), id
from db2admin.x_lo_tab
where id = '${id}'
%REPORT{
<h2>$(V1)</h2>
<h4>Description:</h4> <blockquote>$(V2) </blockquote>
<h4>Copyrights:</h4> <blockquote>$(V3) </blockquote>
<a href="showImage?id=$(V_id)"></a>
<a href="getVideo?id=$(V_id)"></a>
<a href="getAudio?id=$(V_id)&filename=$V6">[Listen]</a>
% }
% }
```

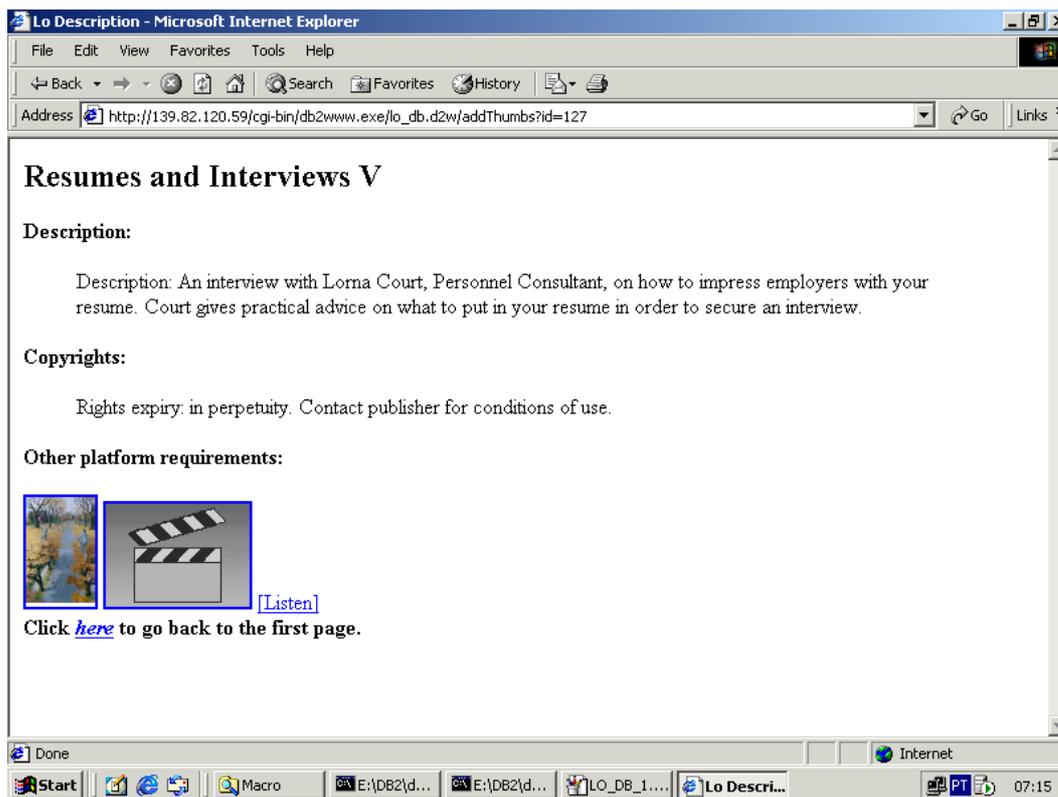


Figura 35. Tela Detalhes de Um Objeto de Aprendizado

Teclando sobre a imagem reduzida da figura será apresentada a tela exposta na **Figura 36. Tela Imagem de um Objeto de Aprendizado**. Ao teclar sobre o símbolo de vídeo ou sobre a música, tem-se acesso respectivamente às páginas que apresentam o clipe do vídeo ou finalmente o som associado ao objeto de aprendizagem. Estas páginas são resultado do processamento dos blocos *showImage*, *getVideo* e *getAudio*, que fazem chamadas às funções *showImageSQL*, *showVideoSQL* e *showAudioSQL*.

```

%function (DTW_SQL) showImageSQL(){
select      cast(mmdbsys.content_i(lo_image,      'GIF')      as      blob(150000)),
mmdbsys.format_i(lo_image)
from db2admin.x_lo_tab
where id = '$(id)'
%REPORT{
%ROW{  <br><br><b>Original image format: $(V2)</b>% }
% }
    
```

```
%function (DTW_SQL) showVideoSQL(){
  select          mmdbsys.comment(lo_video),          mmdbsys.content_v(lo_video,
mmdbsys.comment(lo_video), 1),
  mmdbsys.format_v(lo_video)
  from db2admin.x_lo_tab
  where id = '$(id)'
  %REPORT{
  %ROW{ <a href="/tmplobs/$(V1)"><i><b> Play Video Clip</b></i></a>
<br><br><b>Format: $(V3)
  % }
```

```
%function (DTW_SQL) showAudioSQL(){
  select          mmdbsys.comment(lo_audio),          mmdbsys.content(lo_audio,
mmdbsys.comment(lo_audio), 1),
  mmdbsys.format(lo_audio)
  from db2admin.x_lo_tab
  where id = '$(id)'
  %REPORT{
  %ROW{ <a href="/tmplobs/$(V1) " ><i><b>Play Audio Clip</b></i></a>
<br><br><b>Format: $(V3)</b>
  % }
```

As funções extraem os dados e produzem a tela correspondente no bloco *REPORT*.

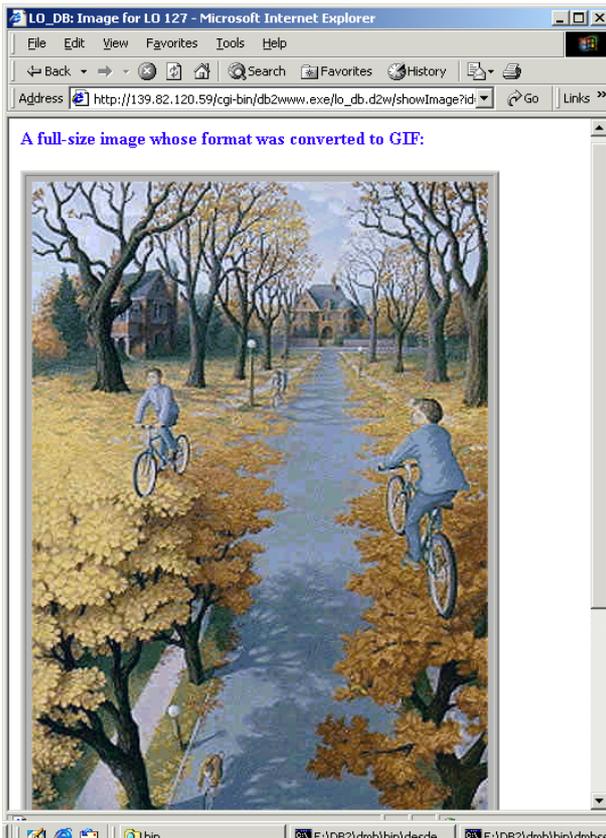


Figura 36. Tela Imagem de um Objeto de Aprendizado

É importante observar que para obter acesso aos dados multimídia é importante que os extensores de áudio, vídeo e imagem do IBM/DB2 estejam ativados. Este processo acontece, nas versões utilizadas, apenas quando existe uma sessão ativa no servidor (logada) e após a tela de ativação, conforme apresentada na **Figura 37. Tela de ativação do extensor AVI**.

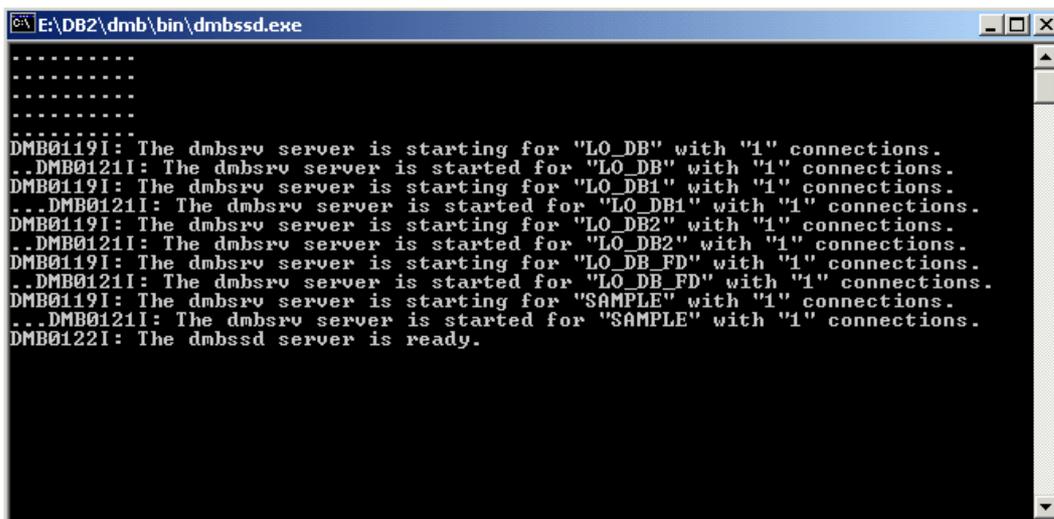


Figura 37. Tela de ativação do extensor AVI

Se o extensor AVI estiver desativado, a identificação de existência de dados AVI associados não será apresentada na tela apresentada na **Figura 35**. Neste caso a imagem do mesmo objeto de aprendizado será a apresentada conforme a **Figura 38**.

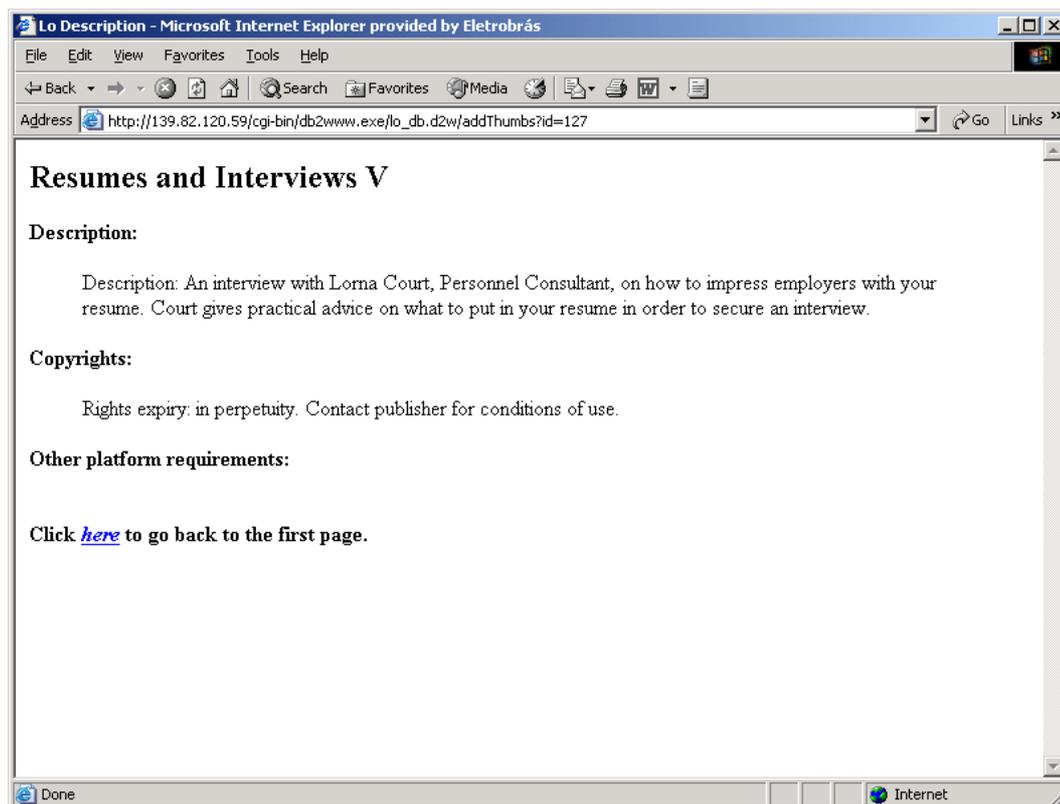


Figura 38. Tela Detalhes de Um Objeto de Aprendizado com Extensor AVI Desativado.

Observe-se também que os itens de cada tela são variáveis conforme a existência ou não dos elementos descritivos correspondentes. Isto pode ser verificado pelas **Figuras 39 e 40**.

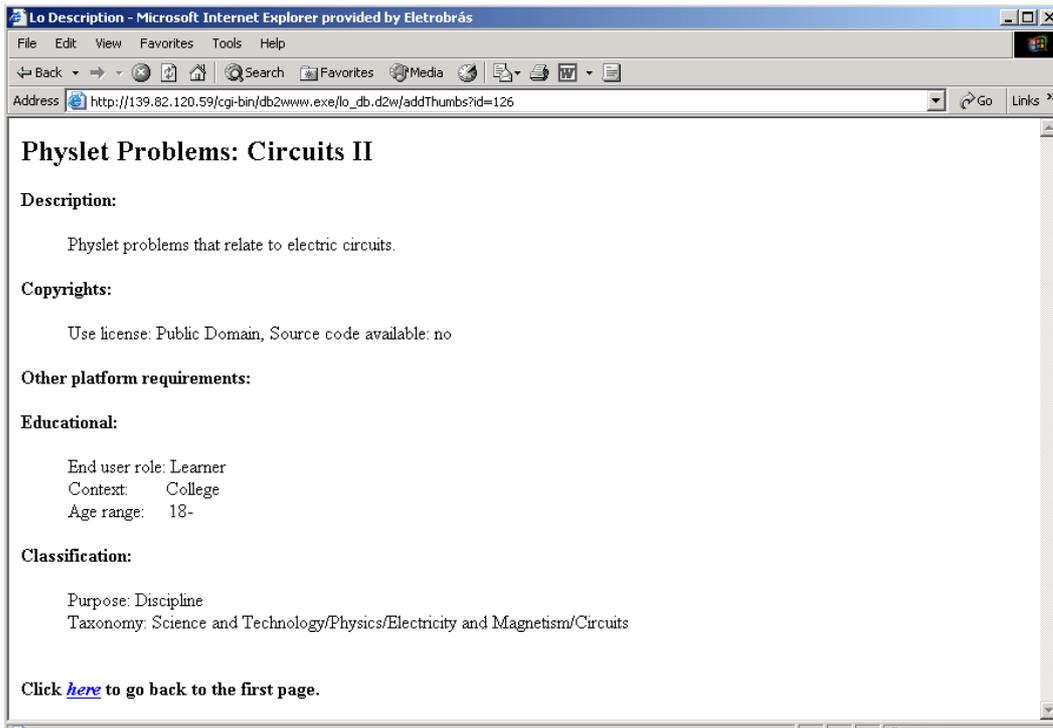


Figura 39. Tela Detalhes de Um Objeto de Aprendizado (2).

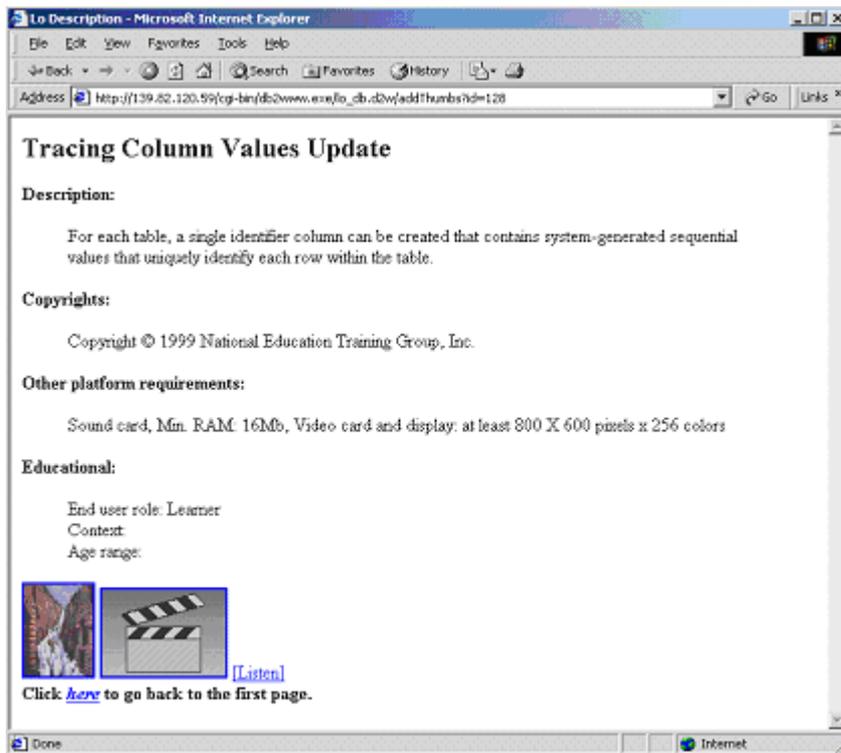


Figura 40. Tela Detalhes de Um Objeto de Aprendizado (3).

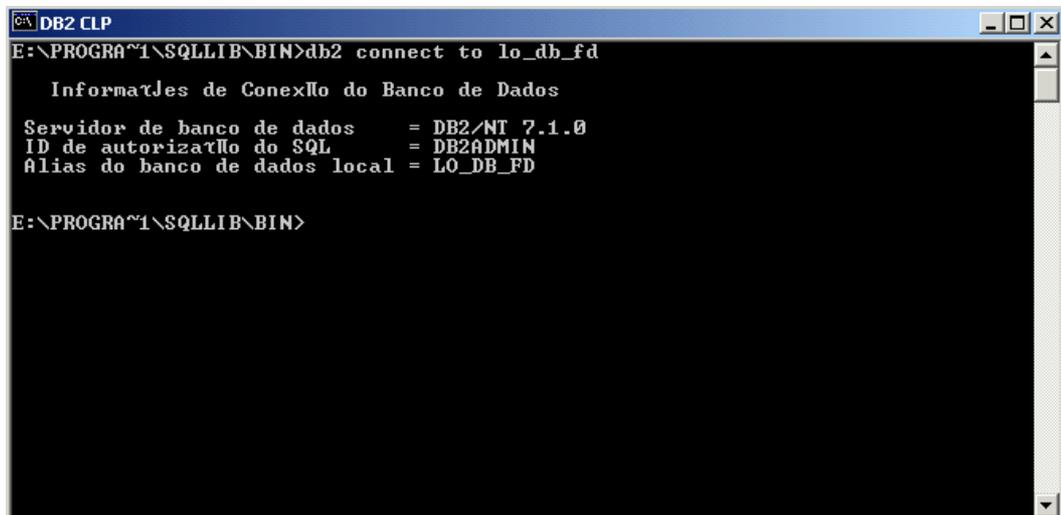
5.7. Validação do Processo de Carga

Este item tem por objetivo apresentar um resumo sobre o processo de carga de dados das bases componentes e da base federada, uma vez que nos itens anteriores foi dada ênfase à solução de possíveis problemas comuns a este tipo de implementação.

Os mesmos descritores de objetos de aprendizado foram utilizados para carga nas bases componentes. O que variou foi o número de objetos carregados e os arquivos de AVI associados.

O material carregado pode ser validado seguindo os seguintes passos:

a) Conexão à base federada



```
DB2 CLP
E:\PROGRAMAS\SQLLIB\BIN>db2 connect to lo_db_fd

Informações de Conexão do Banco de Dados
Servidor de banco de dados      = DB2/NT 7.1.0
ID de autorização do SQL       = DB2ADMIN
Alias do banco de dados local   = LO_DB_FD

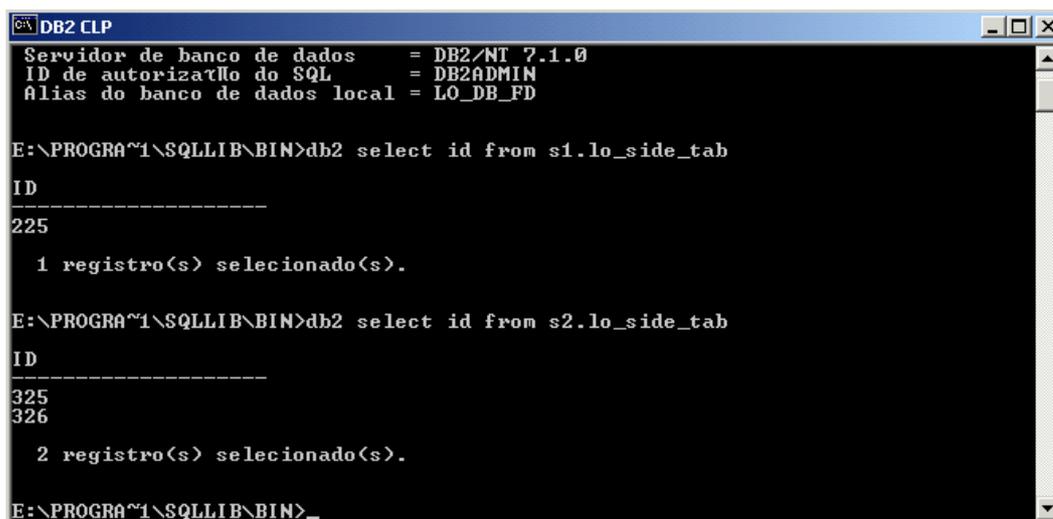
E:\PROGRAMAS\SQLLIB\BIN>
```

Figura 41. Conexão à Base Federada.

b) Consulta ao conteúdo de cada site. A **Figura 42** apresenta o conteúdo das relações componentes chaves de cada site:

s1.lo_side_tab

s2.lo_side_tab



```
DB2 CLP
Servidor de banco de dados = DB2/NT 7.1.0
ID de autorização do SQL = DB2ADMIN
Alias do banco de dados local = LO_DB_FD

E:\PROGRAMAS\SQLLIB\BIN>db2 select id from s1.lo_side_tab
ID
-----
225

1 registro(s) selecionado(s).

E:\PROGRAMAS\SQLLIB\BIN>db2 select id from s2.lo_side_tab
ID
-----
325
326

2 registro(s) selecionado(s).

E:\PROGRAMAS\SQLLIB\BIN>
```

Figura 42. Consulta ao conteúdo de cada site.

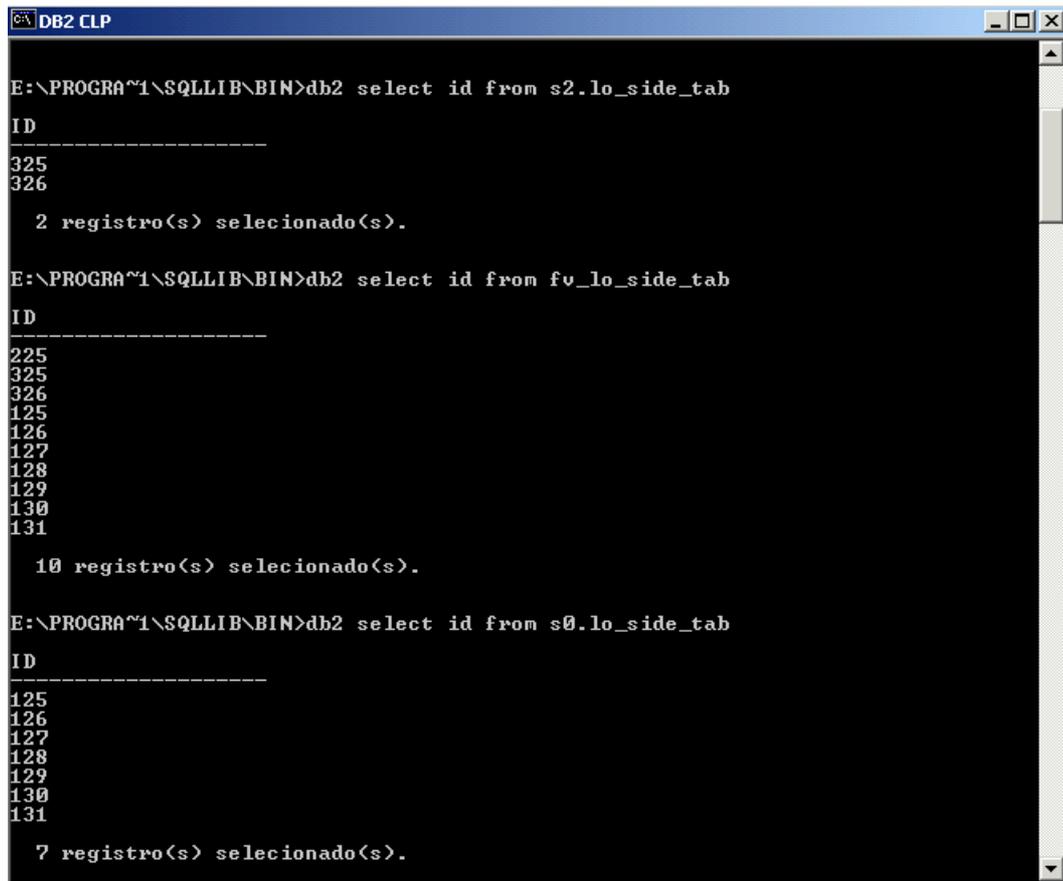
- c) Consulta ao conteúdo da federação. A **Figura 43** apresenta o conteúdo da federação e também da base local que também fez parte da federação. Esta base foi incluída para aumentar o conteúdo do exemplo e não estava prevista inicialmente.

Assim, temos o seguinte mapeamento das relações listadas na

Figura 43:

fv_lo_side_tav: visão federada

s0.lo_side_tab: relação local



```
DB2 CLP
E:\PROGRAMA~1\SQLLIB\BIN>db2 select id from s2.lo_side_tab
ID
-----
325
326
      2 registro(s) selecionado(s).

E:\PROGRAMA~1\SQLLIB\BIN>db2 select id from fv_lo_side_tab
ID
-----
225
325
326
125
126
127
128
129
130
131
     10 registro(s) selecionado(s).

E:\PROGRAMA~1\SQLLIB\BIN>db2 select id from s0.lo_side_tab
ID
-----
125
126
127
128
129
130
131
      7 registro(s) selecionado(s).
```

Figura 43. Consulta ao conteúdo da federação.

- d) Verificação da implementação pelo painel de controle do IBM DB2.

As **Figuras 44 e 45** apresentam o esquema da base de dados federada através do painel de controle do IBM DB2. Nesta figura podemos observar os dois sites referenciados no item de elementos federados e as relações que foram escolhidas para acesso federado.

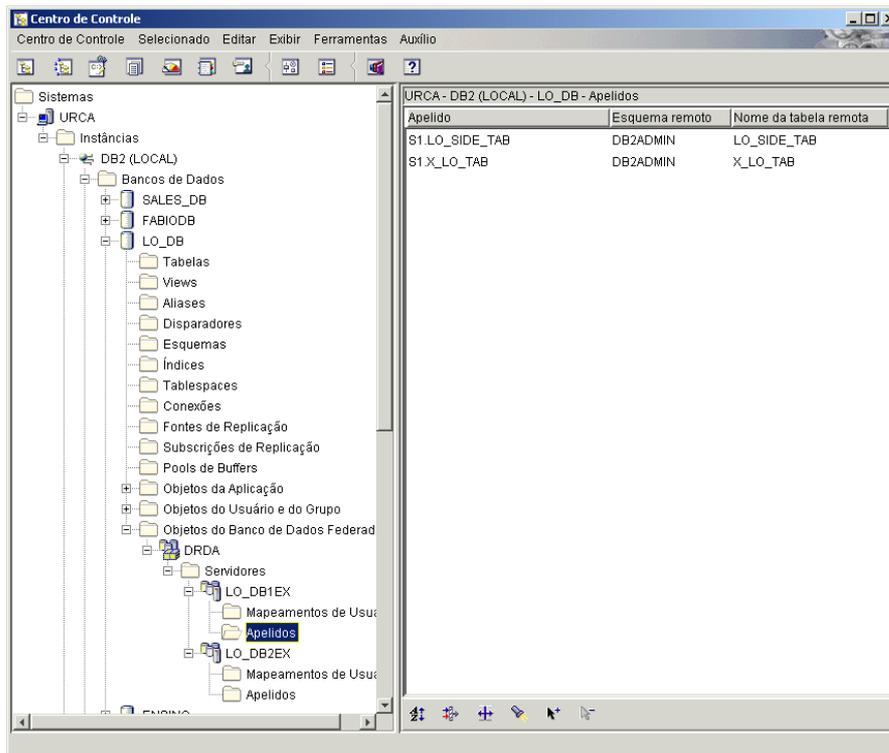


Figura 44. Base Federada através do IBM DB2 – site 1.

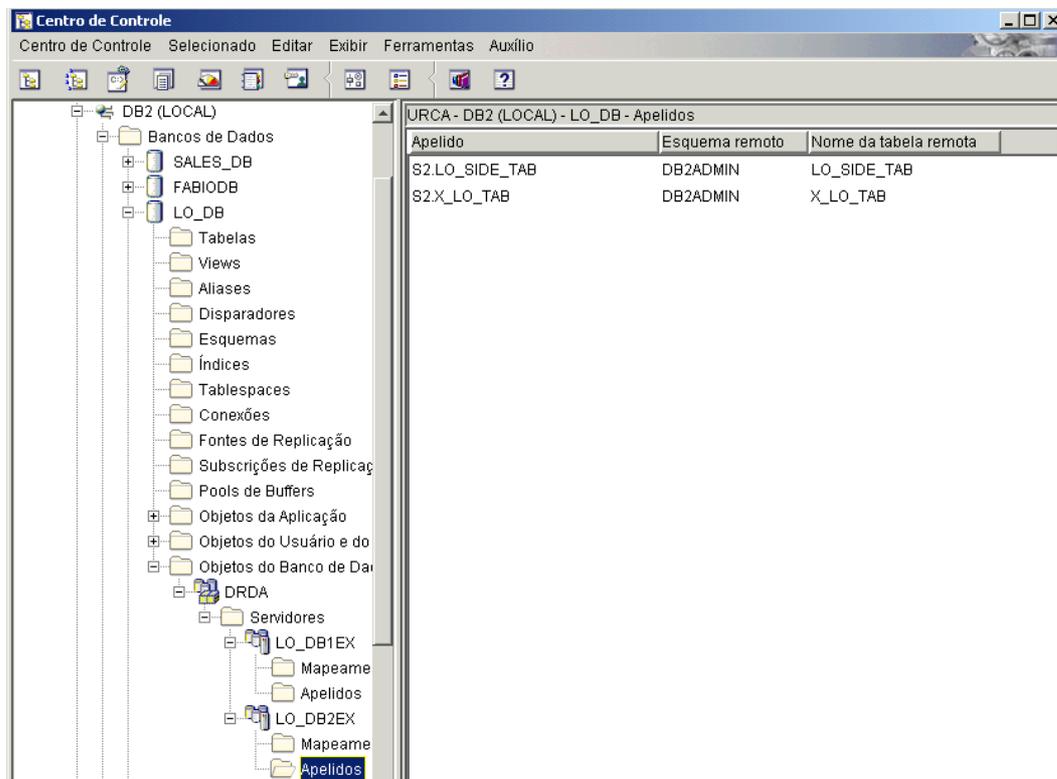


Figura 45. Base Federada através do IBM DB2 – site 2.

Nas **Figuras 46 e 47** é apresentada a composição da visão federada.

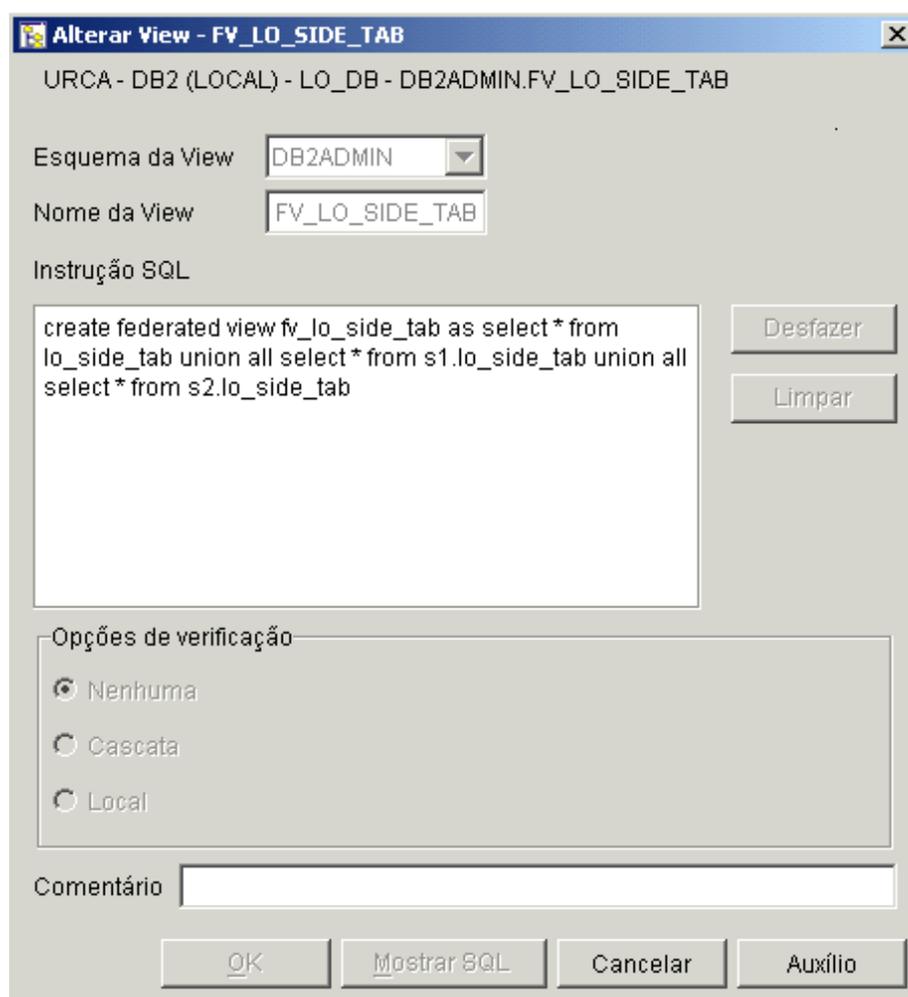


Figura 46. Base Federada através do IBM DB2 – Visão Federada – Relação de LO.

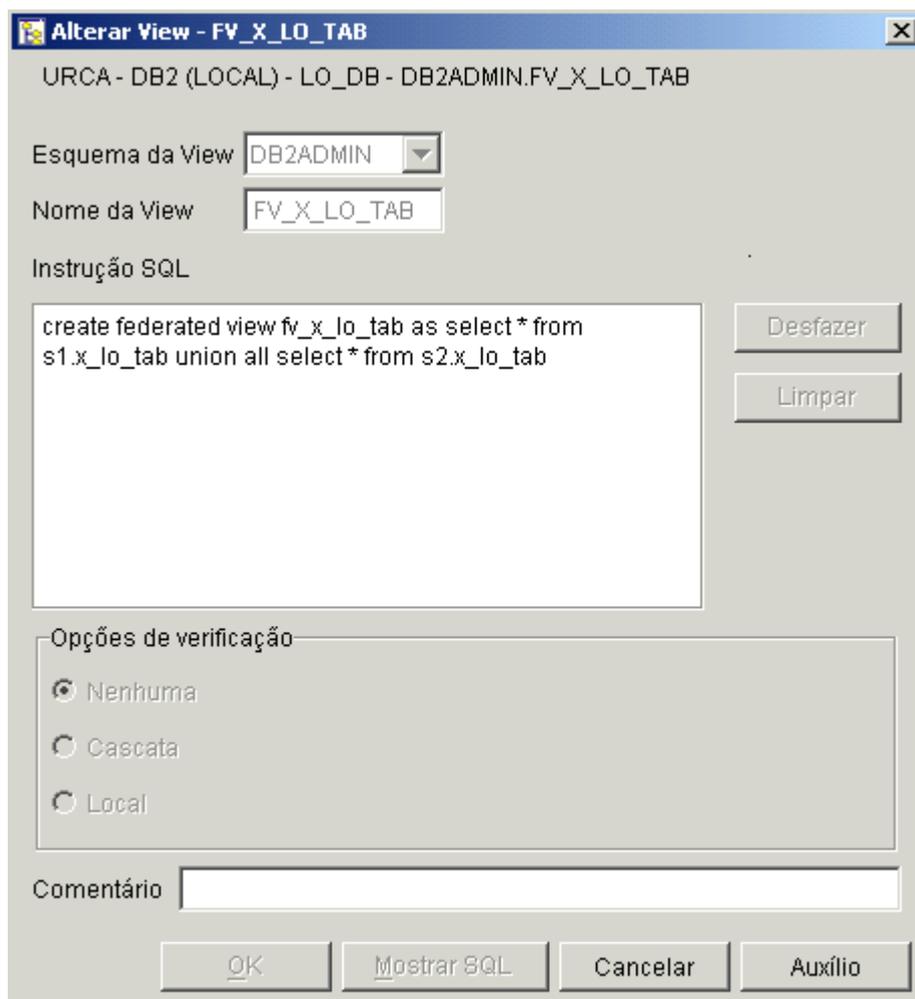


Figura 47. Base Federada através do IBM DB2 – Visão Federada – Detalhes de LO.

Na **Figura 48** é apresentada a definição de uma visão local demonstrando a sobrecarga de definições.

5.8. Versões do IBM DB2

Para desenvolvimento do protótipo foram utilizadas diferentes versões do produto IBM DB2. Isto foi necessário porque, ao utilizar versões demonstração, nem todas as funcionalidades estão disponíveis. Neste item são apresentadas algumas particularidades que tiveram impacto no protótipo.

1. DB2 Enterprise Server Edition 8.1, com fixpack 2 aplicado:
 - o Conjunto de *wrappers*: DRDA
 - o Mais de uma base federada no servidor: não
 - o Licença: 3 meses

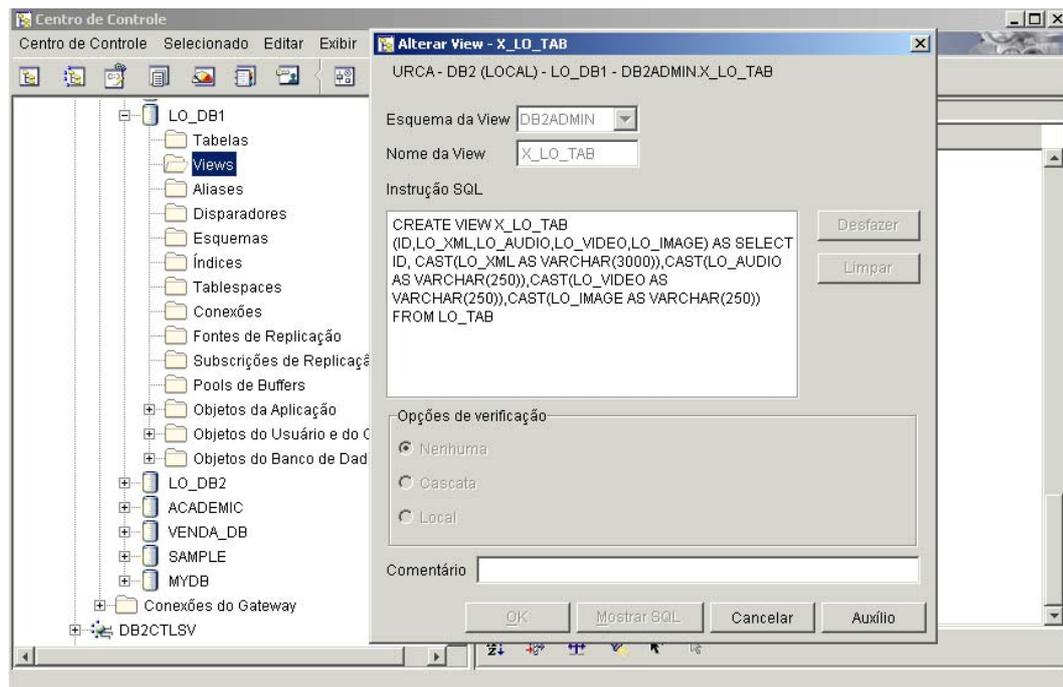


Figura 48. Visão com Sobrecarga – Detalhes de LO.

2. DB2 Enterprise Server Edition 7.1.3 (ou 7.2):
 - Conjunto de *wrappers*: DRDA, SQLNET, NET8, DBLIB, CTLIB(SYBASE), DJXMSSQL3
 - Mais de uma base federada no servidor: sim
 - Aceita base local como membro: não
 - Licença: 3 meses
3. DB2 Enterprise Edition 7.2.8, (7.1.3 com fixpack 8):
 - Conjunto de *wrappers*: DRDA, INFORMIX, OLEDB
 - Licença: Permanente
 - Para aplicar o *service pack*, a máquina deve ser iniciada na conta de instalação;
4. DB2 Enterprise Edition 7.2.10, (7.2 com *fixpack* 10 service level wr 21329):
 - Conjunto de *wrappers*: DRDA, INFORMIX, OLEDB
 - Licença: Permanente
 - Para aplicar o *service pack* deve estar conectado na conta de instalação;
5. PUC - DB2 Enterprise Edition 7.1:

- Conjunto de *wrappers*: DRDA, SQLNET, NET8
- Licença: Permanente

Para *download* dos produtos e documentações foi utilizado o seguinte endereço:

ftp://ftp.software.ibm.com/

/ps/products/db2/fixes/english-us/db2docv8/Windows/FP2_U486569

IP: 207.25.253.61

user:anonymous

password: e-mail

5.9. Observações Operacionais

Ao longo dos trabalhos foram observados os seguintes comportamentos não documentados claramente nos produtos:

- a) Ao utilizar uma base produzida na versão 7.2 com gerenciador na versão 8.2.1 é necessário executar novamente os comandos de atribuição dos extensores:

Enable_db for avi, xml

- b) Uma instância na versão 8.2.1 não consegue ter como membro uma base na versão 7.1;
- c) O parâmetro *federated* é ao nível da instância;
- d) Para resolver o problema identificado pelo erro “*not enough memory*” deve ser ajustado o valor do parâmetro APPLHEAPSZ para 512:

SQL1022C There is not enough memory available to process this command SQLSTATE=57011

