

Guia XML-copyright.	3
Prefácio.	7
Recursos da Informatica.	7
Conceitos de XML.	9
Visão Geral dos Conceitos XML.	9
Arquivos XML.	9
Arquivos DTD.	13
Arquivos de Esquema XML.	15
Tipos de Metadados XML.	16
Cardinalidade.	18
Tipos de XML simples e complexos.	21
Elementos e atributos Any Type.	25
Grupos de componentes.	29
XML Path.	32
Páginas de Código.	32
Usando XML com o PowerCenter.	33
Visão Geral do Uso do XML com o PowerCenter.	33
Importando Metadados XML.	34
Noções Básicas sobre as Exibições XML.	40
Noções Básicas de Relacionamentos Hierárquicos.	41
Noções Básicas de Relacionamentos de Entidade.	45
Trabalhando com Referências Circulares.	51
Noções Básicas de Linhas de Exibição.	53
Colunas dinâmicas.	55
Trabalhando com Origens XML.	57
Trabalhando com a Visão Geral de Origens XML.	57
Importando uma Definição de Origem de XML.	58
Trabalhando com exibições XML.	60
Gerando Relacionamentos de Entidade.	62
Gerando relacionamentos de hierarquia.	62
Criando exibições XML personalizadas.	63

Sincronizando definições XML.	64
Editando Propriedades de Definição de Origem XML.	65
Criando definições XML a partir de definições de repositório.	67
Solucionando Problemas de Origens XML.	68
Usando o Editor de XML.	69
Usando a visão geral do Editor de XML.	69
Criando e editando exibições.	72
Criando um Predicado de Consulta do XPath.	78
Mantendo relacionamentos de exibição.	82
Exibindo componentes do esquema.	83
Definindo opções de exibição XML.	87
Solucionando problemas de trabalho com o Editor de XML.	95
Trabalhando com Destinos XML.	96
Visão geral do trabalho com destinos XML.	96
Importando uma definição de destino XML de um arquivo XML.	96
Criando um Destino a partir de uma Definição de Origem XML.	97
Editando Propriedades da Definição de Destino XML.	98
Validando Destinos XML.	100
Usando um Destino XML em um Mapeamento.	101
Solucionando Problemas dos Destinos XML.	104
Transformação de Qualificador de Origem XML.	106
Visão Geral da Transformação de Qualificador de Origem XML.	106
Adicionando um Qualificador de Origem XML a um Mapeamento.	106
Editando uma Transformação de Qualificador de Origem XML.	107
Usando o Qualificador de Origem XML em um Mapeamento.	109
Solucionando Problemas das Transformações de Qualificador de Origem XML.	112
Transformações XML de Midstream.	113
Visão Geral das Transformações XML de Midstream.	113
Transformação de Analisador de XML.	113
Transformação de gerador de XML.	118
Criando uma Transformação XML de Midstream.	118
Sincronizando uma definição XML de midstream.	119
Editando propriedades de transformação XML midstream.	119
Gerando portas de passagem.	122
Solucionando Problemas das Transformações XML de Midstream.	123
Referência de Tipo de Dados XML.	124

Tipos de dados XML e de transformação.	124
Referência de Funções de Consulta do XPath.	127
Visão Geral das Funções de Consulta do XPath.	127
Referência Rápida de Função.	127
boolean.	129
ceiling.	130
concat.	130
contains.	131
false.	132
floor.	133
lang.	133
normalize-space.	134
not.	135
number.	135
round.	136
starts-with.	137
string.	138
string-length.	139
subsequência.	139
subsequência-depois.	141
substring-before.	142
translate.	142
true.	143

Guia XML-copyright

Este software e a documentação são fornecidos somente sob um contrato de licença separado, contendo restrições sobre uso e divulgação. Não está permitida de forma alguma a reprodução ou a transmissão de qualquer parte deste documento (seja por meio eletrônico, fotocópia, gravação ou quaisquer outros meios) sem o consentimento prévio da Informatica LLC.

Informatica, o logotipo Informatica e PowerCenter são marcas comerciais ou marcas registradas da Informatica LLC nos Estados Unidos e em muitas jurisdições por todo o mundo. Uma lista atual das marcas comerciais da Informatica está disponível na Internet em <https://www.informatica.com/trademarks.html>. Os nomes de outras companhias e produtos podem ser nomes ou marcas comerciais de seus respectivos proprietários.

Partes desta documentação e/ou software estão sujeitas a copyright de terceiros, incluindo sem limitação: Copyright DataDirect Technologies. Todos os direitos reservados. Copyright © Sun Microsystems. Todos os direitos reservados. Copyright © RSA Security Inc. Todos os direitos reservados. Copyright © Ordinal Technology Corp. Todos os direitos reservados. Copyright © Aandacht

c.v. Todos os direitos reservados. Copyright Genivia, Inc. Todos os direitos reservados. Copyright Isomorphic Software. Todos os direitos reservados. Copyright © Meta Integration Technology, Inc. Todos os direitos reservados. Copyright © Intalio. Todos os direitos reservados. Copyright © Oracle. Todos os direitos reservados. Copyright © Adobe Systems Incorporated. Todos os direitos reservados. Copyright © DataArt, Inc. Todos os direitos reservados. Copyright © ComponentSource. Todos os direitos reservados. Copyright © Microsoft Corporation. Todos os direitos reservados. Copyright © Rogue Wave Software, Inc. Todos os direitos reservados. Copyright © Teradata Corporation. Todos os direitos reservados. Copyright © Yahoo! Inc. Todos os direitos reservados. Copyright © Glyph & Cog, LLC. Todos os direitos reservados. Copyright © Thinkmap, Inc. Todos os direitos reservados. Copyright © Clearpace Software Limited. Todos os direitos reservados. Copyright © Information Builders, Inc. Todos os direitos reservados. Copyright © OSS Nokalva, Inc. Todos os direitos reservados. Copyright Edifecs, Inc. Todos os direitos reservados. Copyright Cleo Communications, Inc. Todos os direitos reservados. Copyright © International Organization for Standardization 1986. Todos os direitos reservados. Copyright © ej-technologies GmbH. Todos os direitos reservados. Copyright © Jaspersoft Corporation. Todos os direitos reservados. Copyright © International Business Machines Corporation. Todos os direitos reservados. Copyright © yWorks GmbH. Todos os direitos reservados. Copyright © Lucent Technologies. Todos os direitos reservados. Copyright © University of Toronto. Todos os direitos reservados. Copyright © Daniel Veillard. Todos os direitos reservados. Copyright © Unicode, Inc. Copyright IBM Corp. Todos os direitos reservados. Copyright © MicroQuill Software Publishing, Inc. Todos os direitos reservados. Copyright © PassMark Software Pty Ltd. Todos os direitos reservados. Copyright © LogiXML, Inc. Todos os direitos reservados. Copyright © 2003-2010 Lorenzi Davide, todos os direitos reservados. Copyright © Red Hat, Inc. Todos os direitos reservados. Copyright © The Board of Trustees of the Leland Stanford Junior University. Todos os direitos reservados. Copyright © EMC Corporation. Todos os direitos reservados. Copyright © Flexera Software. Todos os direitos reservados. Copyright © Jinfonet Software. Todos os direitos reservados. Copyright © Apple Inc. Todos os direitos reservados. Copyright © Telerik Inc. Todos os direitos reservados. Copyright © BEA Systems. Todos os direitos reservados. Copyright © PDFlib GmbH. Todos os direitos reservados. Copyright © Orientation in Objects GmbH. Todos os direitos reservados. Copyright © Tanuki Software, Ltd. Todos os direitos reservados. Copyright © Ricebridge. Todos os direitos reservados. Copyright © Sencha, Inc. Todos os direitos reservados. Copyright © Scalable Systems, Inc. Todos os direitos reservados. Copyright © jqWidgets. Todos os direitos reservados. Copyright © Tableau Software, Inc. Todos os direitos reservados. Copyright © MaxMind, Inc. Todos os direitos reservados. Copyright © TMate Software s.r.o. Todos os direitos reservados. Copyright © MapR Technologies Inc. Todos os direitos reservados. Copyright © Amazon Corporate LLC. Todos os direitos reservados. Copyright © Highsoft. Todos os direitos reservados. Copyright © Python Software Foundation. Todos os direitos reservados. Copyright © BeOpen.com. Todos os direitos reservados. Copyright © CNRI. Todos os direitos reservados.

Este produto inclui software desenvolvido pela Apache Software Foundation (<http://www.apache.org/>) e/ou outros softwares licenciados nas várias versões da Licença Apache (a "Licença"). Você pode obter uma cópia dessas Licenças em <http://www.apache.org/licenses/>. A menos que exigido pela legislação aplicável ou concordado por escrito, o software distribuído em conformidade com estas Licenças é fornecido "NO ESTADO EM QUE SE ENCONTRA", SEM GARANTIA OU CONDIÇÃO DE QUALQUER TIPO, seja expressa ou implícita. Consulte as Licenças para conhecer as limitações e as permissões que regulam o idioma específico de acordo com as Licenças.

Este produto inclui software desenvolvido pela Mozilla (<http://www.mozilla.org/>), direitos autorais de software de The JBoss Group, LLC; todos os direitos reservados; software copyright © 1999-2006 de Bruno Lowagie e Paulo Soares e outros produtos de software licenciados sob a Licença Pública GNU Lesser General Public License Agreement, que pode ser encontrada em <http://www.gnu.org/licenses/>

lgpl.html. Os materiais são fornecidos gratuitamente pela Informatica, no estado em que se encontram, sem garantia de qualquer tipo, explícita nem implícita, incluindo, mas não limitando-se, as garantias implicadas de comerciabilidade e adequação a um determinado propósito.

O produto inclui software ACE(TM) e TAO(TM) com copyright de Douglas C. Schmidt e seu grupo de pesquisa na Washington University, University of California, Irvine e Vanderbilt University, Copyright (©) 1993-2006, todos os direitos reservados.

Este produto inclui o software desenvolvido pelo OpenSSL Project para ser usado no kit de ferramentas OpenSSL (copyright The OpenSSL Project. Todos os direitos reservados) e a redistribuição deste software está sujeita aos termos disponíveis em <http://www.openssl.org> e <http://www.openssl.org/source/license.html>.

Este produto inclui o software Curl com o Copyright 1996-2013, Daniel Stenberg, <daniel@haxx.se>. Todos os direitos reservados. Permissões e limitações relativas a este software estão sujeitas aos termos disponíveis em <http://curl.haxx.se/docs/copyright.html>. É permitido usar, copiar, modificar e distribuir este software com qualquer objetivo, com ou sem taxa, desde que a nota de direitos autorais acima e esta nota de permissão apareçam em todas as cópias.

O produto inclui software copyright 2001-2005 (©) MetaStuff, Ltd. Todos os direitos reservados. Permissões e limitações relativas a este software estão sujeitas aos termos disponíveis em <http://www.dom4j.org/license.html>.

Este produto inclui o copyright de software © 1996-2006 Per Bothner. Todos os direitos reservados. O direito de usar tais materiais é estabelecido na licença que pode ser encontrada em <http://www.gnu.org/software/kawa/Software-License.html>.

Este produto inclui o software OSSP UUID com Copyright © 2002 Ralf S. Engelschall, Copyright © 2002 e OSSP Project Copyright © 2002 Cable & Wireless Deutschland. Permissões e limitações relativas a este software estão sujeitas aos termos disponíveis em <http://www.opensource.org/licenses/mit-license.php>.

Este produto inclui software desenvolvido pela Boost (<http://www.boost.org/>) ou sob a licença de software Boost. Permissões e limitações relativas a este software estão sujeitas aos termos disponíveis em http://www.boost.org/LICENSE_1_0.txt.

Este produto inclui software copyright © 1997-2007 University of Cambridge. Permissões e limitações relativas a este software estão sujeitas aos termos disponíveis em <http://www.pcre.org/license.txt>.

Este produto inclui o copyright de software © 2007 The Eclipse Foundation. Todos os direitos reservados. As permissões e as limitações relativas a este software estão sujeitas aos termos disponíveis em <http://www.eclipse.org/org/documents/epl-v10.php> e em <http://www.eclipse.org/org/documents/edl-v10.php>.

Este produto inclui softwares licenciados de acordo com os termos disponíveis em <http://www.tcl.tk/software/tcltk/license.html>, <http://www.bosrup.com/web/overlib/?License>, <http://www.stlport.org/doc/license.html>, <http://asm.ow2.org/license.html>, <http://www.cryptix.org/LICENSE.TXT>, <http://hsqldb.org/web/hsqllLicense.html>, <http://httpunit.sourceforge.net/doc/license.html>, <http://jung.sourceforge.net/license.txt>, http://www.gzip.org/zlib/zlib_license.html, <http://www.openldap.org/software/release/license.html>, <http://www.libssh2.org>, <http://slf4j.org/license.html>, <http://www.sente.ch/software/OpenSourceLicense.html>, <http://fusesource.com/downloads/license-agreements/fuse-message-broker-v-5-3-license-agreement>; <http://antlr.org/license.html>; <http://aopalliance.sourceforge.net/>; <http://www.bouncycastle.org/licence.html>; <http://www.jgraph.com/>

jgraphdownload.html; <http://www.jcraft.com/jsch/LICENSE.txt>; http://jotm.objectweb.org/bsd_license.html; . <http://www.w3.org/Consortium/Legal/2002/copyright-software-20021231>; <http://www.slf4j.org/license.html>; <http://nanoxml.sourceforge.net/orig/copyright.html>; <http://www.json.org/license.html>; <http://forge.ow2.org/projects/javaservice/>; <http://www.postgresql.org/about/licence.html>; <http://www.sqlite.org/copyright.html>; <http://www.tcl.tk/software/tcltk/license.html>; <http://www.jaxen.org/faq.html>; <http://www.jdom.org/docs/faq.html>; <http://www.slf4j.org/license.html>; <http://www.iodbc.org/dataspace/iodbc/wiki/iODBC/License>; <http://www.keplerproject.org/md5/license.html>; <http://www.toedter.com/en/jcalendar/license.html>; <http://www.edankert.com/bounce/index.html>; <http://www.net-snmp.org/about/license.html>; <http://www.openmdx.org/#FAQ>; http://www.php.net/license/3_01.txt; <http://srp.stanford.edu/license.txt>; <http://www.schneier.com/blowfish.html>; <http://www.jmock.org/license.html>; <http://xsom.java.net>; <http://benalman.com/about/license/>; <https://github.com/CreateJS/EaselJS/blob/master/src/easeljs/display/Bitmap.js>; <http://www.h2database.com/html/license.html#summary>; <http://jsoncpp.sourceforge.net/LICENSE>; <http://jdbc.postgresql.org/license.html>; <http://protobuf.googlecode.com/svn/trunk/src/google/protobuf/descriptor.proto>; <https://github.com/rantav/hector/blob/master/LICENSE>; <http://web.mit.edu/Kerberos/krb5-current/doc/mitK5license.html>; <http://jibx.sourceforge.net/jibx-license.html>; <https://github.com/lyokato/libgeohash/blob/master/LICENSE>; <https://github.com/hjiang/jsonxx/blob/master/LICENSE>; <https://code.google.com/p/lz4/>; <https://github.com/jedisct1/libsodium/blob/master/LICENSE>; <http://one-jar.sourceforge.net/index.php?page=documents&file=license>; <https://github.com/EsotericSoftware/kryo/blob/master/license.txt>; <http://www.scala-lang.org/license.html>; <https://github.com/tinkerpop/blueprints/blob/master/LICENSE.txt>; <http://gee.cs.oswego.edu/dl/classes/EDU/oswego/cs/dl/util/concurrent/intro.html>; <https://aws.amazon.com/asl/>; <https://github.com/twbs/bootstrap/blob/master/LICENSE>; <https://sourceforge.net/p/xmlunit/code/HEAD/tree/trunk/LICENSE.txt>.

Este produto inclui software licenciado de acordo com a Academic Free License (<http://www.opensource.org/licenses/afl-3.0.php>), a Common Development and Distribution License (<http://www.opensource.org/licenses/cddl1.php>), a Common Public License (<http://www.opensource.org/licenses/cpl1.0.php>), a Sun Binary Code License Agreement Supplemental License Terms, a BSD License (<http://www.opensource.org/licenses/bsd-license.php>), a nova BSD License (<http://opensource.org/licenses/BSD-3-Clause>), a MIT License (<http://www.opensource.org/licenses/mit-license.php>), a Artistic License (<http://www.opensource.org/licenses/artistic-license-1.0>) e a Initial Developer's Public License Version 1.0 (<http://www.firebirdsql.org/en/initial-developer-s-public-license-version-1-0/>).

Este produto inclui copyright do software © 2003-2006 Joe Walnes, 2006-2007 XStream Committers. Todos os direitos reservados. Permissões e limitações relativas a este software estão sujeitas aos termos disponíveis em <http://xstream.codehaus.org/license.html>. Este produto inclui software desenvolvido pelo Indiana University Extreme! Lab. Para obter mais informações, visite <http://www.extreme.indiana.edu/>.

Este produto inclui software Copyright © 2013 Frank Balluffi e Markus Moeller. Todos os direitos reservados. As permissões e limitações relativas a este software estão sujeitas aos termos da licença MIT.

Consulte as patentes em <https://www.informatica.com/legal/patents.html>.

ISENÇÃO DE RESPONSABILIDADE: a Informatica LLC fornece esta documentação no estado em que se encontra, sem garantia de qualquer tipo, expressa ou implícita, incluindo, mas não limitando-se, as garantias implícitas de não infração, comercialização ou uso para um determinado propósito. A Informatica LLC não garante que este software ou documentação não contenha erros. As informações

fornecidas neste software ou documentação podem incluir imprecisões técnicas ou erros tipográficos. As informações deste software e documentação estão sujeitas a alterações a qualquer momento sem aviso prévio.

AVISOS

Este produto da Informatica (o "Software") traz determinados drivers (os "drivers da DataDirect") da DataDirect Technologies, uma empresa em funcionamento da Progress Software Corporation ("DataDirect"), que estão sujeitos aos seguintes termos e condições:

1. OS DRIVERS DA DATADIRECT SÃO FORNECIDOS NO ESTADO EM QUE SE ENCONTRAM, SEM GARANTIA DE QUALQUER TIPO, EXPRESSA OU IMPLÍCITA, INCLUINDO, MAS NÃO LIMITANDO-SE, AS GARANTIAS IMPLÍCITAS DE COMERCIALIZAÇÃO, ADEQUAÇÃO A UMA FINALIDADE ESPECÍFICA E NÃO INFRAÇÃO.
2. EM NENHUM CASO, A DATADIRECT OU SEUS FORNECEDORES TERCEIRIZADOS SERÃO RESPONSÁVEIS, EM RELAÇÃO AO CLIENTE FINAL, POR QUAISQUER DANOS DIRETOS, INDIRETOS, INCIDENTAIS, ESPECIAIS, CONSEQUENCIAIS OU DEMAIS QUE POSSAM ADVIR DO USO DE DRIVERS ODBC, SENDO OU NÃO ANTERIORMENTE INFORMADOS DAS POSSIBILIDADES DE TAIS DANOS. ESTAS LIMITAÇÕES SE APLICAM A TODAS AS CAUSAS DE AÇÃO, INCLUINDO, SEM LIMITAÇÕES, QUEBRA DE CONTRATO, QUEBRA DE GARANTIA, NEGLIGÊNCIA, RESPONSABILIDADE RIGOROSA, DETURPAÇÃO E OUTROS ATOS ILÍCITOS.

As informações contidas neste documento estão sujeitas a alteração sem aviso prévio. Se você encontrar quaisquer problemas nesta documentação, informe-os em infa_documentation@informatica.com.

Os produtos Informatica apresentam garantias segundo os termos e condições dos acordos em que são fornecidos. A INFORMATICA FORNECE AS INFORMAÇÕES NESTE DOCUMENTO "COMO ESTÃO" SEM GARANTIA DE QUALQUER TIPO, EXPRESSA OU IMPLÍCITA, INCLUINDO, SEM QUAISQUER GARANTIAS DE COMERCIALIZAÇÃO, ADEQUAÇÃO A UM DETERMINADO FIM E QUALQUER GARANTIA OU CONDIÇÃO DE NÃO-VIOLAÇÃO.

Prefácio

O *Guia de XML* foi elaborado para desenvolvedores e engenheiros de software responsáveis pelo trabalho com XML em um ambiente de data warehouse. Antes de usar o *Guia de XML*, verifique se você tem uma noção básica sólida dos conceitos de XML, sistemas operacionais, arquivos simples ou sistema mainframe em seu ambiente. Além disso, certifique-se de estar familiarizado com os requisitos de interface para os aplicativos de suporte.

Recursos da Informatica

A Informatica oferece uma variedade de recursos de produtos através da Rede da Informatica e outros portais on-line. Use os recursos para obter o máximo de seus produtos e soluções da Informatica e para aprender com outros usuários da Informatica e especialistas no assunto.

Rede da Informatica

A Rede da Informatica é a porta de entrada para muitos recursos, incluindo a Base de Dados de Conhecimento da Informatica e o Suporte Global a Clientes da Informatica. Para acessar a Rede da Informatica, visite <https://network.informatica.com>.

Como membro da Rede da Informatica, você tem as seguintes opções:

- Pesquisar por recursos do produto na Base de Dados de Conhecimento.
- Visualizar informações sobre disponibilidade de produtos.

- Criar e revisar seus casos de suporte.
- Encontrar a sua Rede de Grupo de Usuários da Informatica local e colaborar com seus colegas.

Base de Dados de Conhecimento da Informatica

Use a Base de Dados de Conhecimento da Informatica para encontrar recursos de produtos, como artigos de instruções, práticas recomendadas, tutoriais em vídeo e respostas a perguntas frequentes.

Para pesquisar na Base de Dados de Conhecimento, visite <https://search.informatica.com>. Em caso de dúvidas, comentários ou ideias sobre a Base de Dados de Conhecimento, entre em contato com a equipe da Base de Dados de Conhecimento da Informatica em KB_Feedback@informatica.com.

Documentação da Informatica

Use o Portal de Documentação da Informatica para explorar uma extensa biblioteca de documentação para versões de produtos atuais e recentes. Para explorar o Portal de Documentação, visite <https://docs.informatica.com>.

Em caso de dúvidas, comentários ou ideias sobre a documentação do produto, entre em contato com a equipe da Documentação da Informatica em infa_documentation@informatica.com.

Matrizes de Disponibilidade de Produto da Informatica

As Matrizes de Disponibilidade de Produto (PAMs) indicam as versões dos sistemas operacionais, os bancos de dados e tipos de fontes e destinos de dados com os quais uma versão de produto é compatível. Veja as PAMs da Informatica em <https://network.informatica.com/community/informatica-network/product-availability-matrices>.

Informatica Velocity

O Informatica Velocity é uma coleção de dicas e práticas recomendadas desenvolvidas pelos Serviços Profissionais da Informatica e baseada em experiências reais de centenas de projetos de gerenciamento de dados. O Informatica Velocity representa o conhecimento coletivo dos consultores da Informatica que trabalham com organizações em todo o mundo para planejar, desenvolver, implantar e manter soluções de gerenciamento de dados bem-sucedidas.

Encontre os recursos do Informatica Velocity em <http://velocity.informatica.com>. Se você tiver dúvidas, comentários ou ideias sobre o Informatica Velocity, entre em contato com os Serviços Profissionais da Informatica em ips@informatica.com.

Informatica Marketplace

O Informatica Marketplace é um fórum onde você pode encontrar soluções que ampliam e aprimoram suas implementações da Informatica. Aproveite as centenas de soluções dos desenvolvedores e parceiros da Informatica no Marketplace para melhorar sua produtividade e agilizar o tempo de implementação em seus projetos. Encontre o Informatica Marketplace em <https://marketplace.informatica.com>.

Suporte Global a Clientes da Informatica

Você pode entrar em contato com um Centro de Suporte Global por telefone ou por meio da Rede da Informatica.

Para descobrir o número de telefone local do Suporte Global a Clientes da Informatica, visite o site da Informatica no seguinte link:

<https://www.informatica.com/services-and-training/customer-success-services/contact-us.html>.

Para encontrar recursos de suporte on-line na Rede da Informatica, visite

<https://network.informatica.com> e selecione a opção eSupport.

Conceitos de XML

Visão Geral dos Conceitos XML

A XML (Linguagem de Marcação Extensível) é uma maneira flexível de criar formatos comuns de informações e de compartilhar os formatos e os dados entre aplicativos e na Internet.

Você pode importar definições XML para o PowerCenter® a partir dos seguintes tipos de arquivos:

- **Arquivo XML.** Um arquivo XML contém dados e metadados. Um arquivo XML pode fazer referência a um arquivo DTD (Definição de Tipo de Documento) ou XSD (definição de esquema XML) para validação.
- **Arquivo DTD.** Um arquivo DTD define os tipos de elementos, atributos e entidades em um arquivo XML. Um arquivo DTD gera algumas restrições na estrutura de arquivo XML, mas não contém dados.
- **Esquema XML.** Um esquema XML define elementos, atributos e definições de tipo. Os esquemas contêm tipos simples e complexos. Um tipo simples é um elemento ou atributo XML que contém texto. Um tipo complexo é um elemento XML que contém outros elementos e atributos.

Os esquemas oferecem suporte a grupos de elementos, atributos e substituições aos quais você pode fazer referência em todo um esquema. Use grupos de substituição para substituir um elemento por outro em um documento de instância XML. Os esquemas também oferecem suporte à herança de elementos, tipos complexos e grupos de elementos e atributos.

Arquivos XML

Os arquivos XML contêm marcas que identificam dados no arquivo XML, mas não o formato dos dados. O componente básico de um arquivo XML é um elemento. Um elemento XML inclui uma marca de início do elemento, o conteúdo do elemento e uma marca de fim do elemento. Todos os arquivos XML devem ter um elemento raiz definido por uma única marca no início e no fim do arquivo. O elemento raiz inclui todos os outros elementos do arquivo.

Um arquivo XML modela um banco de dados hierárquico. A posição de um elemento em uma hierarquia XML representa seus relacionamentos com outros elementos. Um elemento pode conter elementos filhos e herdar características de outros elementos.

Por exemplo, o seguinte arquivo XML descreve um manual:

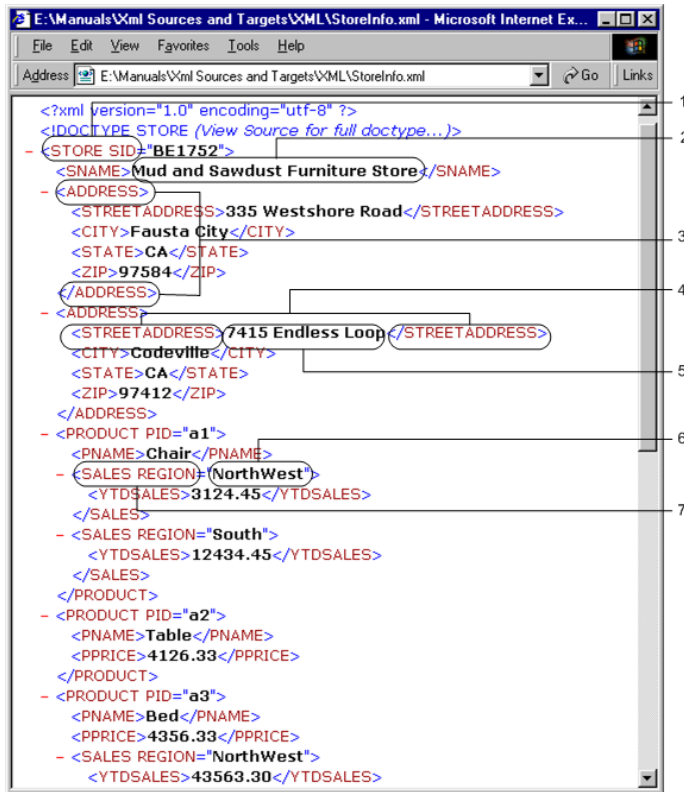
```
<book>
  <title>Fun with XML</title>
  <chapter>
    <heading>Understanding XML</heading>
    <heading>Using XML</heading>
  </chapter>
  <chapter>
    <heading>Using DTD Files</heading>
    <heading>Fun with Schemas</heading>
  </chapter>
</book>
```

Livro é o elemento raiz e contém os elementos título e capítulo. Livro é o elemento pai de título e capítulo, e capítulo é o pai de cabeçalho. Título e capítulo são elementos irmãos, pois eles têm o mesmo pai.

Um elemento pode ter atributos que fornecem informações adicionais sobre o elemento. No seguinte exemplo, o atributo `graphic_type` descreve o conteúdo da figura:

```
<picture graphic_type="gif">computer.gif</picture>
```

A figura seguinte mostra a estrutura, os elementos e os atributos em um arquivo XML:



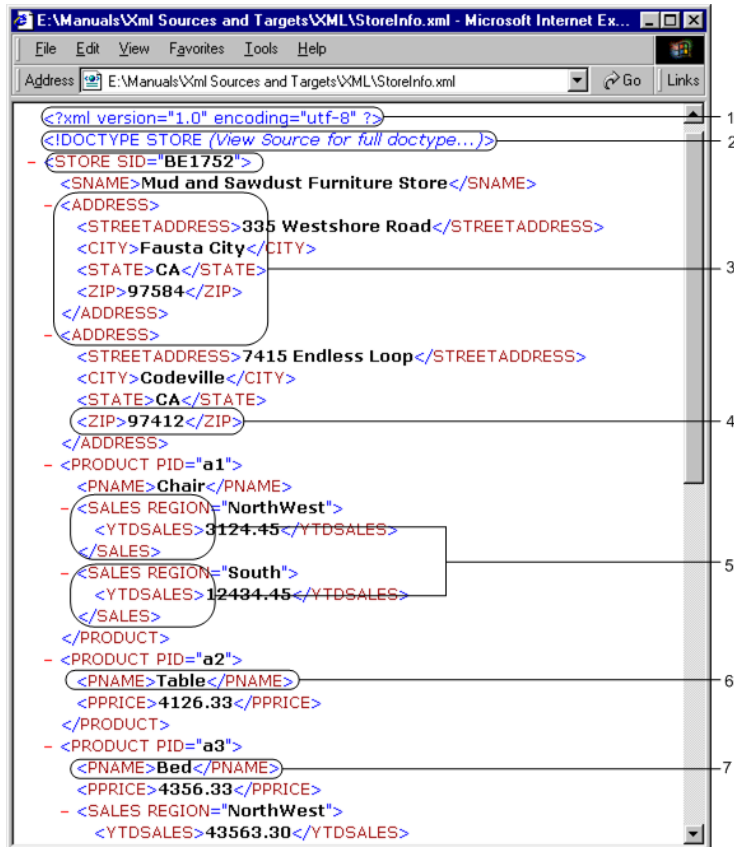
1. Elemento raiz.
2. Dados do elemento.
3. Elemento com compartimento.
4. Marcas dos elementos.
5. Dados do elemento.
6. Valor do atributo
7. Marca de atributo.

Um arquivo XML tem estrutura hierárquica. Uma hierarquia XML inclui os seguintes elementos:

- **Elemento filho.** Um elemento contido dentro de outro elemento.
- **Elemento com compartimento.** Um elemento que contém outros elementos, mas não contém dados. Um elemento com compartimentos pode incluir outros elementos com compartimentos.
- **Elemento global.** Um elemento que é filho direto do elemento raiz. Você pode mencionar elementos globais em todo o esquema XML.
- **Elemento folha.** Um elemento que não contém outros elementos. Um elemento folha é o elemento de nível mais baixo na hierarquia XML.
- **Elemento local.** Um elemento aninhado em outro elemento. Você pode mencionar elementos locais apenas no contexto do elemento pai.
- **Elemento com várias ocorrências.** Um elemento que ocorre mais de uma vez dentro de seu elemento pai. Os elementos com compartimentos podem ser elementos com várias ocorrências.
- **Cadeia pai.** A sucessão de elementos filho-pai que rastreia o caminho de um elemento para a raiz.

- **Elemento pai.** Um elemento que contém outros elementos.
- **Elemento com uma única ocorrência.** Um elemento que ocorre uma vez dentro de seu pai.

A seguinte figura mostra alguns elementos em uma hierarquia XML:



1. O atributo de codificação identifica a página de código.
2. DOCTYPE identifica um arquivo DTD associado.
3. Elemento com compartimentos: o elemento Address inclui os elementos StreetAddress, City, State e Zip. O elemento Address também é um elemento Pai.
4. Elemento folha: o element Zip, juntamente com todos os seus elementos irmãos, é o elemento de nível mais baixo dentro do elemento Address.
5. Elemento com Várias Ocorrências: o elemento Sales Region ocorre mais de uma vez dentro do elemento Product.
6. Elemento com uma única ocorrência: o elemento PName ocorre uma vez dentro do elemento Product.
7. Elemento filho: o elemento PName é um filho de Product, que é filho de Store.

Validando Arquivo XML com um DTD ou Esquema

Um arquivo XML válido está em conformidade com a estrutura de um arquivo DTD ou de esquema associado.

Para fazer referência ao local e ao nome de um arquivo DTD, use a declaração DOCTYPE em um arquivo XML. A declaração DOCTYPE também denomina o elemento raiz do arquivo XML.

Por exemplo, o seguinte arquivo XML faz referência ao local do arquivo note.dtd:

```
<?xml version="1.0"?>
<!DOCTYPE note SYSTEM
"http://www.w3schools.com/dtd/note.dtd">
<note>
  <body>XML Data</body>
</note>
```

Para fazer referência a um esquema, use a declaração schemaLocation. A declaração schemaLocation contém o local e o nome de um esquema.

O seguinte arquivo XML faz referência ao esquema note.xsd em um local externo:

```
<?xml version="1.0"?>
<note xsi:SchemaLocation="http://www.w3schools.com note.xsd">
  <body>XML Data</body>
</note>
```

Codificação Unicode

Um arquivo XML contém um atributo de codificação que indica a página de código no arquivo. As codificações mais comuns são UTF-16LE e UTF-16. UTF-16LE representa um caractere de um a quatro bytes, dependendo do símbolo Unicode. UTF-16 representa um caractere como uma palavra de 16 bits.

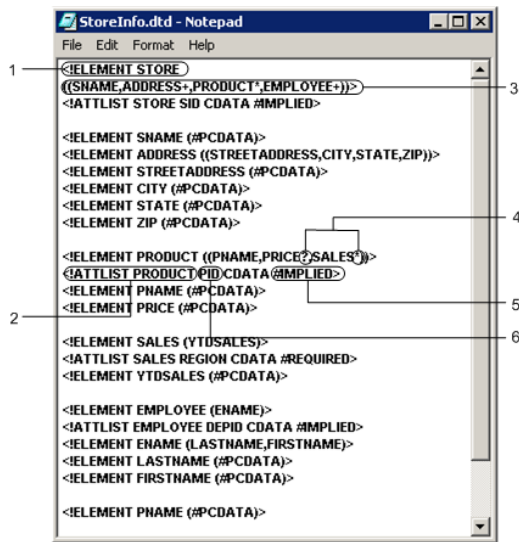
O exemplo a seguir mostra um atributo UTF-16LE em um arquivo XML:

```
<?xml version="1.0"encoding="UTF-16LE"?>
<note xsi:SchemaLocation="http://www.w3schools.com note.xsd">
  <body>XML Data</body>
</note>
```

Arquivos DTD

Um arquivo DTD (Definição de Tipo de Documento) define os tipos de elementos e atributos em um arquivo XML. Um arquivo DTD também fornece algumas restrições na estrutura de arquivos XML. Um arquivo DTD não contém dados ou tipos de dados de elementos.

A seguinte figura mostra elementos e atributos em um arquivo DTD:



1. Elemento
2. Atributo
3. Lista de elementos
4. Ocorrência do elemento
5. Opção de valor de atributo
6. Nome do atributo

Elementos DTD

No arquivo DTD, uma declaração de elemento define um elemento XML. Uma declaração de elemento tem a seguinte sintaxe:

```
<!ELEMENT product (#PCDATA)>
```

A descrição do DTD define a marca XML <product>. A descrição (#PCDATA) especifica os dados de caracteres analisados. Os dados analisados são o texto entre a marca inicial e a marca final de um elemento XML. Os dados de caracteres analisados são o texto sem os elementos filhos.

O exemplo a seguir mostra uma descrição DTD de um elemento com dois elementos filhos:

```
<!ELEMENT boat (brand, type) >
```

```
<!ELEMENT brand (#PCDATA) >
```

```
<!ELEMENT type (#PCDATA) >
```

Brand e type são elementos filhos de boat. Todos os elementos filhos podem conter caracteres. Neste exemplo, brand e type podem ocorrer uma vez dentro do elemento boat. A seguinte descrição de DTD especifica que brand deve ocorrer uma ou mais vezes para um boat:

```
<!ELEMENT boat (brand+) >
```

Atributos de DTD

Os atributos fornecem informações adicionais sobre elementos. Em um arquivo DTD, um atributo ocorre dentro da marca inicial de um elemento.

A seguinte sintaxe descreve um atributo em um arquivo DTD:

```
<!ATTLIST element_name attribute_name attribute_type "default_value">
```

Os parâmetros seguintes identificam um atributo em um arquivo DTD:

- **Nome do elemento** O nome do elemento que tem o atributo.
- **Nome do atributo** O nome do atributo.
- **Tipo do atributo.** O tipo de atributo. O tipo de atributo mais comum é CDATA. Um atributo CDATA são dados de caracteres.
- **Valor padrão** O valor do atributo se nenhum valor ocorrer no arquivo XML.

Use as seguintes opções com um valor padrão:

- **#REQUIRED.** O arquivo XML deve conter o valor de atributo.
- **#IMPLIED.** O valor de atributo é opcional.
- **#FIXED.** O arquivo XML deve conter o valor padrão do arquivo DTD. Um arquivo XML válido pode conter o mesmo valor de atributo que o DTD ou o arquivo XML pode não ter um valor de atributo. Você deve especificar um valor padrão com essa opção.

O seguinte exemplo mostra um atributo com um valor fixo:

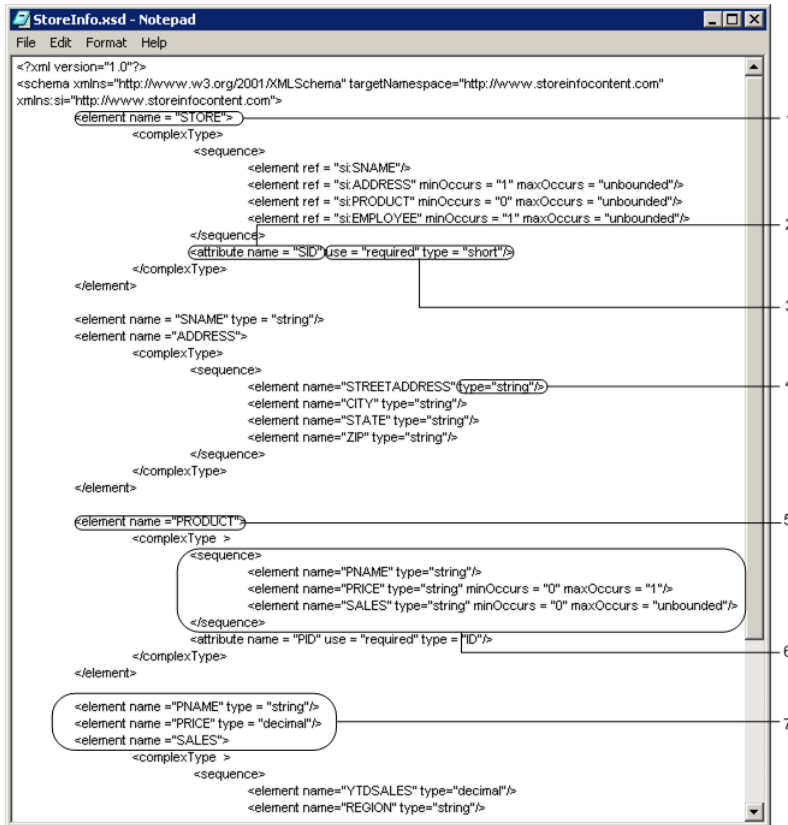
```
<!ATTLIST product product_name CDATA #FIXED "vacuum">
```

O nome do elemento é product. O atributo é product_name. O atributo tem um valor padrão, vacuum.

Arquivos de Esquema XML

Um esquema XML é um documento que define o conteúdo válido de arquivos XML. Um arquivo de esquema XML, como um arquivo DTD, contém apenas metadados. Um esquema XML define a estrutura e o tipo de elementos e atributos para um arquivo XML associado. Ao usar um esquema para definir um arquivo XML, você pode restringir dados, definir formatos de dados e converter dados entre tipos de dados. Um esquema XML oferece suporte a tipos complexos e à herança entre tipos. Um esquema fornece uma maneira de especificar grupos de elementos e atributos, conteúdo ANY e referências circulares.

A seguinte figura mostra componentes de esquema XML:



1. Nome do elemento
2. Atributo
3. Tipo de atributo e construção nula
4. Tipo de dados do elemento
5. Dados do elemento
6. Lista de elementos e ocorrência
7. Lista de elementos e tipo de dados

TÓPICOS RELACIONADOS:

- [“Tipos de XML simples e complexos” na página 21](#)
- [“Grupos de componentes” na página 29](#)

Tipos de Metadados XML

Você pode criar definições XML do PowerCenter a partir de arquivos XML, DTD ou de esquema XML. Os arquivos XML fornecem dados e metadados. Os arquivos DTD e de esquema XML fornecem metadados.

O PowerCenter extrai os seguintes tipos de metadados de arquivos XML, DTD e de esquema XML:

- **Namespace.** Uma coleção de elementos e nomes de atributos identificados por uma referência de URI (Uniform Resource Identifier) em um arquivo XML. O namespace diferencia entre elementos provenientes de diferentes origens.

- **Nome.** Uma marca que contém o nome de um elemento ou atributo.
- **Hierarquia.** A posição de um elemento no relacionamento com outros elementos em um arquivo XML.
- **Cardinalidade.** O número de vezes que um elemento ocorre em um arquivo XML.
- **Tipo de dados.** Uma classificação de elemento de dados, como numeric, string, Boolean ou time. O XML oferece suporte a heranças e tipos de dados personalizados.

Namespace

Um namespace contém um URI para identificar o local do esquema. Um URI é uma string de caracteres que identifica um recurso de Internet. Um URI é uma abstração de uma URL. Uma URL localiza um recurso, mas um URI identifica um recurso. Um arquivo DTD ou de esquema não precisa existir no local do URI.

Um namespace XML identifica grupos de elementos. Um namespace pode identificar elementos e atributos de diferentes arquivos XML ou distinguir significados entre elementos. Por exemplo, você pode distinguir significados para o elemento “table” declarando diferentes namespaces, como *math:table* e *furniture:table*. XML diferencia maiúsculas de minúsculas. O namespace *Math:table* é diferente do namespace *math:table*.

Você pode declarar um namespace no nível raiz de um arquivo XML ou pode declarar um namespace dentro de qualquer elemento em uma estrutura XML. Ao declarar vários namespaces no mesmo arquivo XML, você usa um prefixo de namespace para associar um elemento a um namespace. Uma declaração de namespace é exibida no arquivo XML como um atributo que começa com xmlns. Declare o prefixo do namespace com o atributo xmlns. Você pode criar um nome de prefixo de qualquer tamanho.

O seguinte exemplo mostra dois namespaces em um documento de instância XML:

```
<example>
  xmlns:math = "http://www.mathtables.com"
  xmlns:furniture = "http://www.home.com">
  <math:table>4X6</math:table>
  <furniture:table>Brueners </furniture:table>
</example>
```

Um namespace tem elementos math e o outro namespace tem elementos furniture. Cada namespace tem um elemento chamado “table”, mas os elementos contêm tipos diferentes de dados. O prefixo do namespace distingue entre a tabela math e a tabela furniture.

O seguinte texto mostra uma declaração de esquema comum:

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.w3XML.com"
  xmlns="http://www.w3XML.com"
  elementFormDefault="qualified">...
...</xs:schema>
```

A seguinte tabela descreve cada parte da declaração do namespace:

Declaração de Esquema	Descrição
<code>xmlns:xs="http://www.w3.org/2001/XMLSchema"</code>	Namespace que contém o esquema e os tipos de dados XML nativos. Neste exemplo, cada componente do esquema tem o prefixo "xs".
<code>targetNamespace="http://www.w3XML.com"</code>	Namespace que contém o esquema.
<code>xmlns="http://www.w3XML.com"</code>	Declaração de namespace padrão. Todos os elementos no esquema que não têm prefixo pertencem ao namespace padrão. Declare um namespace padrão usando um atributo xmlns sem prefixo.
<code>elementFormDefault="qualified"</code>	Especifica que qualquer elemento no esquema deve ter um namespace no arquivo XML.

Nome

Em um arquivo XML, cada marca é o nome de um elemento ou atributo. Em um arquivo DTD, a marca `<!ELEMENT>` especifica o nome de um elemento e a marca `<!ATTLIST>` indica o conjunto de atributos de um elemento. Em um arquivo de esquema, `<element name>` especifica o nome de um elemento e `<attribute name>` especifica o nome de um atributo.

Quando você importa uma definição XML, as marcas de elementos se tornam os nomes de coluna na definição do PowerCenter, por padrão.

Hierarquia

Um arquivo XML modela um banco de dados hierárquico. A posição de um elemento em uma hierarquia XML representa seu relacionamento com os outros elementos. Por exemplo, um elemento pode conter elementos filho, e elementos podem herdar características de outros elementos.

Cardinalidade

Cardinalidade de elemento em um arquivo DTD ou de esquema é o número de vezes que um elemento ocorre em um arquivo XML. A cardinalidade de elemento afeta como você estrutura grupos em uma definição XML. A cardinalidade absoluta e a cardinalidade relativa de elementos afetam a estrutura de uma definição XML.

Cardinalidade Absoluta

A cardinalidade absoluta de um elemento é o número de vezes que um elemento ocorre dentro de seu elemento pai em uma hierarquia XML. Os arquivos DTD e de esquema XML descrevem a cardinalidade absoluta dos elementos na hierarquia. Um arquivo DTD usa símbolos e um arquivo de esquema XML usa os atributos `<minOccurs>` e `<maxOccurs>` para descrever a cardinalidade absoluta de um elemento.

Por exemplo, um elemento terá uma cardinalidade absoluta de um (1) se o elemento uma vez dentro de seu elemento pai. No entanto, o elemento poderá ocorrer várias vezes em uma hierarquia XML se o elemento pai tiver cardinalidade e um ou mais (+).

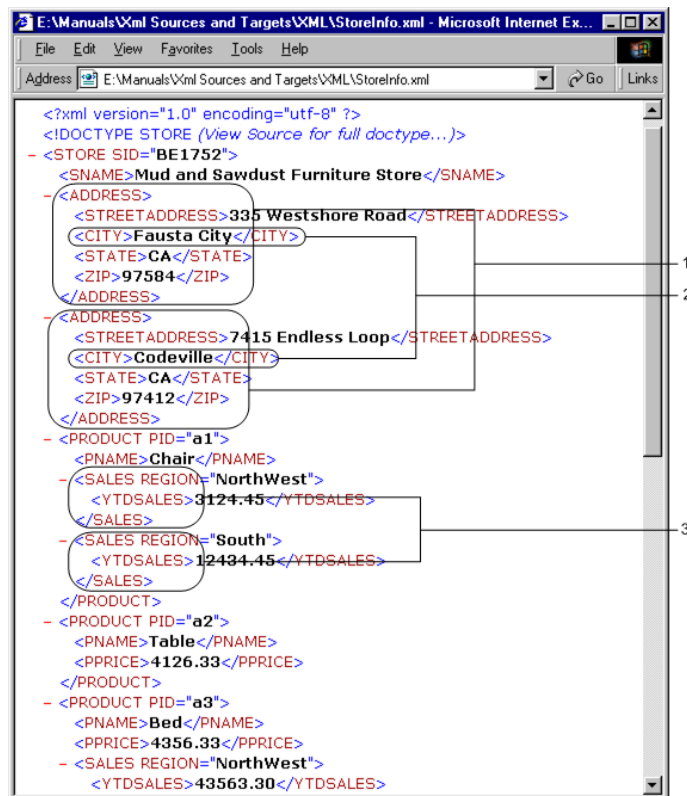
A cardinalidade absoluta de um elemento determina sua restrição nula. Um elemento que tem cardinalidade absoluta de um ou mais (+) não pode ter valores nulos, mas um elemento com cardinalidade de zero ou mais (*) pode ter valores nulos. Um atributo marcado como fixo ou requerido em um arquivo de esquema XML ou DTD não pode ter valores nulos, mas um atributo implícito pode ter valores nulos.

A tabela a seguir descreve como os arquivos de esquema XML e DTD representam a cardinalidade:

Cardinalidade Absoluta	DTD	Esquema
Zero ou uma vez	?	minOccurs=0 maxOccurs=1
Zero ou uma ou mais vezes	*	minOccurs=0 maxOccurs=não vinculado minOccurs=0 maxOccurs=n
Uma Vez	-	minOccurs=1 maxOccurs=1
Uma ou mais vezes	+	minOccurs=1 maxOccurs=não vinculado minOccurs=1 maxOccurs=n

Nota: Você pode declarar um número máximo de ocorrências ou ocorrências ilimitadas em um esquema.

A figura seguinte mostra a cardinalidade absoluta dos elementos em um arquivo XML de amostra:



1. O elemento *Address* ocorre mais de uma vez em *Store*. Sua cardinalidade absoluta é uma ou mais(+).
2. O elemento *City* ocorre uma vez em seu elemento pai *Address*. Sua cardinalidade absoluta é única(1).
3. O elemento *Sales* ocorre zero ou mais vezes em seu elemento pai *Product*. Sua cardinalidade absoluta é zero ou mais(*).

Cardinalidade relativa

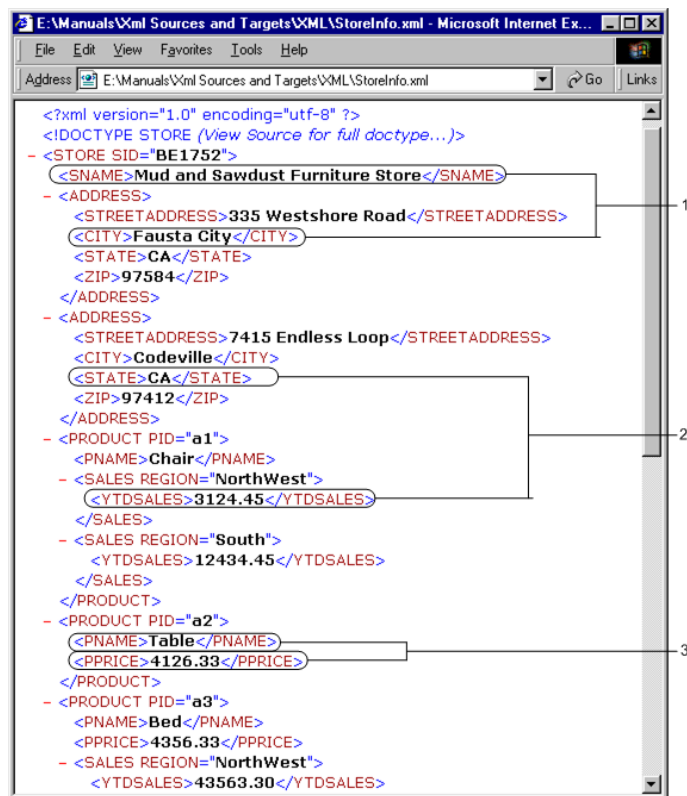
Cardinalidade relativa é o relacionamento de um elemento com outro na hierarquia de XML. Um elemento pode ter um relacionamento um para um, um para muitos ou muitos para muitos com outro elemento na hierarquia.

O elemento terá um relacionamento um para um com outro elemento se todas as ocorrências de um elemento puderem ter uma ocorrência do outro elemento. Por exemplo, um elemento employee pode ter um elemento social security number. Employee e social security number tem um relacionamento um para um.

O elemento terá um relacionamento um para muitos com outro elemento se todas as ocorrências de um elemento puderem ter várias ocorrências do outro elemento. Por exemplo, um elemento employee pode ter vários endereços de email. Employee e email address têm um relacionamento um para muitos.

O elemento terá um relacionamento muitos para muitos com outro elemento se um arquivo XML puder ter várias ocorrências dos dois elementos. Por exemplo, um employee pode ter vários endereços de email e vários endereços. Email address e street address têm um relacionamento muitos para muitos.

A seguinte figura mostra a cardinalidade relativa entre os elementos em um arquivo XML de amostra:



1. Relacionamento um para muitos. Para cada ocorrência de SNAME, pode haver muitas ocorrências de ADDRESS e, dessa forma, muitas ocorrências de CITY.
2. Relacionamento muitos para muitos. Para cada ocorrência de STATE, pode haver várias ocorrências de YTDSALES. Para cada ocorrência de YTDSALES, pode haver várias ocorrências de STATE.
3. Relacionamento um para um. Para cada ocorrência de PNAME, existe uma ocorrência de PPRICE.

Tipos de XML simples e complexos

A linguagem do esquema XML tem mais de 40 tipos de dados internos, incluindo numérico, strings, hora, XML e binário. Esses tipos de dados são denominados tipos simples. Eles contêm texto, mas nenhum outro elemento e atributo. Você pode derivar novos tipos simples dos tipos simples XML básicos.

Você pode criar tipos de dados XML complexos. Um tipo de dados complexo é aquele que contém mais de um tipo simples. Um tipo de dados complexo pode conter também outros tipos e atributos complexos.

Para obter mais informações sobre os tipos de dados XML, consulte as especificações W3C para tipos de dados XML em <http://www.w3.org/TR/xmlschema-2>.

Tipos simples

Um tipo de dado simples é um elemento ou atributo XML que contém texto. Um tipo simples é indivisível. Os tipos simples não podem ter atributos, mas os atributos são tipos simples.

O PowerCenter oferece suporte aos seguintes tipos simples:

- **Tipos atômicos.** Um tipo de dado básico, como booleano, strings ou inteiro.
- **Listas.** Uma coleção de matrizes de tipos atômicos.
- **Unões.** Uma combinação de um ou mais tipos atômicos ou de listas que são mapeados para um tipo simples em um arquivo XML.

Tipos Atômicos

Um tipo de dados atômico é um tipo de dados básico, como Boolean, string, integer, decimal ou date. Para definir tipos de dados atômicos personalizados, adicione restrições a um tipo de dados atômico para limitar o conteúdo. Use uma faceta para definir os valores que devem ser restringidos ou permitidos.

Faceta é uma expressão que define valores mínimos ou máximos, valores específicos ou um padrão de dados de valores válidos. Por exemplo, uma faceta de padrão restringe um elemento a uma expressão de valores de dados. Uma faceta de enumeração lista os valores legais de um elemento.

O seguinte exemplo contém uma faceta de padrão que restringe um elemento a uma letra minúscula entre a e z:

```
<xs:element name="letter">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:pattern value="[a-z]" />
    </xs:restriction>
  </xs:simpleType></xs:element>
```

O seguinte exemplo contém uma faceta de enumeração que restringe uma string a a, b ou c:

```
<xs:element name="letter">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="a" />
      <xs:enumeration value="b" />
      <xs:enumeration value="c" />
    </xs:restriction>
  </xs:simpleType></xs:element>
```

```
</xs:restriction>
</xs:simpleType></xs:element>
```

Listas

Lista é uma coleção de matrizes de tipos atômicos, como uma lista de strings que representam nomes. A lista `itemType` define o tipo de dados dos componentes da lista.

O seguinte exemplo mostra uma lista denominada `names`:

```
<xs:simpleType name="names">
  <xs:list itemType="xs:string" />
</xs:simpleType>
```

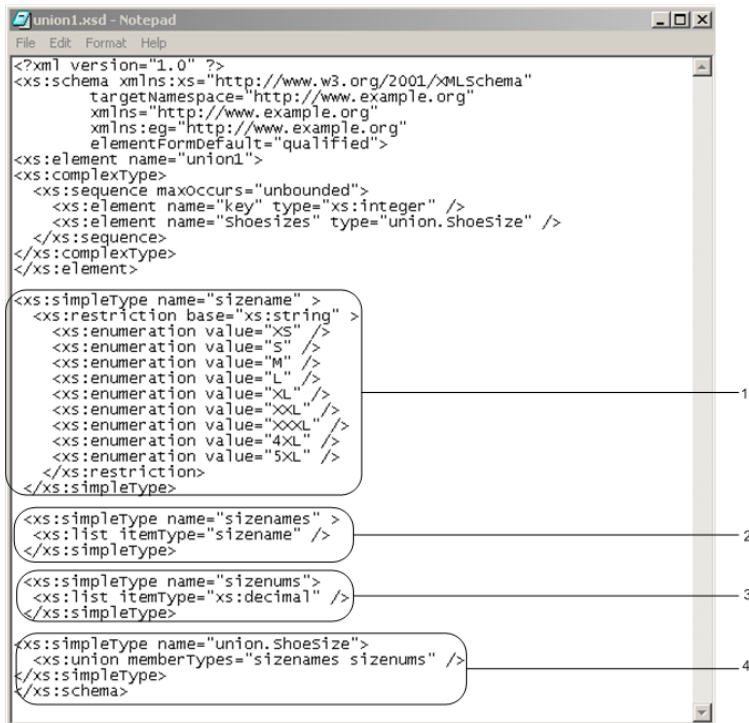
Um arquivo XML pode conter os seguintes dados na lista `names`:

```
<names>Joe Bob Harry Atlee Will</names>
```

Unões

Uma união é a combinação de um ou mais tipos atômicos ou de listas que são mapeados para um tipo simples em um arquivo XML. Ao definir um tipo `union`, você especifica os tipos que combinam. Por exemplo, você pode criar um tipo chamado `size`. `Size` pode incluir dados da string, como S, M e L, ou pode conter tamanhos decimais, como 30, 32 e 34. Se você definir um elemento do tipo `union`, o arquivo XML poderá incluir um tipo `sizenam` para tamanhos de string e um tipo `sizeum` para tamanhos numéricos.

A seguinte figura mostra um arquivo de esquema que contém uma união shoesize com as listas sizenames e sizenums:



1. Sizename é um tipo de string restrito.
2. O tipo sizenames aceita uma lista de strings.
3. O tipo sizenums aceita uma lista decimais.
4. A união shoesize aceita as listas decimal e string.

A união define sizenames e sizenums como tipos de membro union. Sizenames define uma lista de valores de string. Sizenums define uma lista de valores decimais.

Tipos Complexos

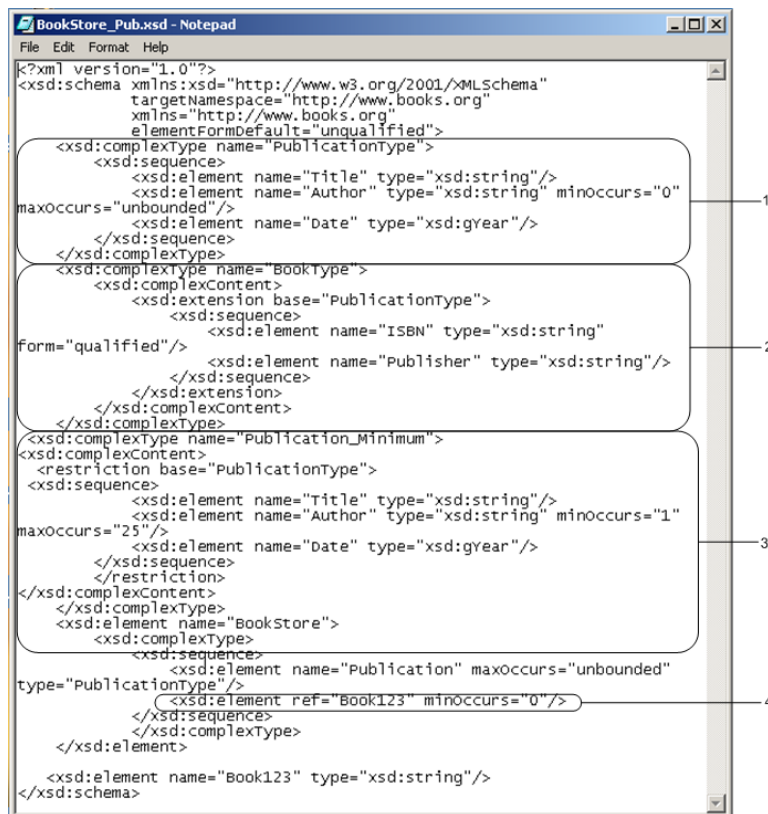
Um tipo complexo agrega uma coleção de tipos simples em uma unidade lógica. Por exemplo, um tipo de cliente pode incluir o número do cliente, o nome, o endereço, o bairro, a cidade e o CEP. Um tipo complexo também pode fazer referência a outros tipos complexos ou grupos de elementos e atributos.

O XML suporta a herança de tipo complexo. Ao definir um tipo complexo, você pode criar outros tipos complexos que herdam os componentes do tipo base. Em um relacionamento de tipo, o tipo base é o tipo complexo a partir do qual você deriva outro tipo. Um tipo complexo derivado herda elementos do tipo base.

Um tipo complexo estendido é um tipo derivado que herda elementos de um tipo base e inclui elementos adicionais. Por exemplo, o tipo `customer_purchases` pode herdar sua definição do tipo complexo `customer`, mas o tipo `customer_purchases` adiciona os elementos `item`, `cost` e `date_sold`.

Um tipo complexo restrito é um tipo derivado que restringe alguns elementos do tipo base. Por exemplo, `mail_list` pode herdar elementos de `customer`, mas restringir o elemento `phone_number` definindo os limites de `minoccurs` e `maxoccurs` como zero.

A seguinte figura mostra os tipos complexos derivados que restringem e estendem o tipo complexo base:



1. Tipo complexo base
2. Tipo complexo estendido
3. Tipo complexo restrito
4. Referência de elemento

Na figura acima, o tipo base é `PublicationType`. `BookType` estende `PublicationType` e inclui os elementos `ISBN` e `Publisher`. `Publication_Minimum` restringe `PublicationType`. `Publication_Minimum` requer entre 1 e 25 `Authors` e restringe a data ao ano.

Elementos abstratos

Às vezes, um esquema contém um tipo base que define a estrutura básica de um elemento complexo, mas não contém todos os componentes. Os tipos complexos derivados estendem o tipo base com mais componentes. Como o tipo base não é uma definição completa, convém não o usar em um arquivo XML. Você pode declarar que o elemento de tipo base é abstrato. Um elemento abstrato não é válido em um arquivo XML. Somente os elementos derivados são válidos.

Para definir um elemento abstrato, adicione um atributo `abstract` com o valor `"true."` O padrão é `false`.

Por exemplo, `PublicationType` é um elemento abstrato. `BookType` herda os elementos em `PublicationType`, mas não inclui os elementos `ISBN` e `Publisher`. Como `PublicationType` é abstrato, um elemento `PublicationType` não é válido no arquivo XML. Um arquivo XML pode conter o tipo derivado, `BookType`.

O seguinte esquema contém PublicationType e BookType:

```
<xsd:complexType name="PublicationType" abstract="true">
  <xsd:sequence>
    <xsd:element name="Title" type="xsd:string"/>
    <xsd:element name="Author" type="xsd:string" minOccurs="0"
maxOccurs="unbounded"/>
    <xsd:element name="Date" type="xsd:gYear"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="BookType">
  <xsd:complexContent>
    <xsd:extension base="PublicationType" >
      <xsd:sequence>
        <xsd:element name="ISBN" type="xsd:string"/>
        <xsd:element name="Publisher" type="xsd:string"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

Elementos e atributos Any Type

Alguns elementos e atributos de esquema permitem qualquer tipo de dados em um arquivo XML. Use esses elementos e atributos quando precisar validar um arquivo XML que tenha tipos não identificados de elementos e atributos.

Use os seguintes elementos e atributos que permitem qualquer tipo de dados:

- **Elementos anyType.** Permite que um elemento seja qualquer tipo de dados no arquivo XML associado.
- **Elemento anySimpleType.** Permite que um elemento seja qualquer simpleType no arquivo XML associado.
- **Elemento de conteúdo ANY.** Permite que um elemento seja qualquer elemento já definido no esquema.
- **Atributo anyAttribute.** Permite que um elemento seja qualquer atributo já definido no esquema.

Elementos anyType

Um elemento anyType pode ser qualquer tipo de dado em um documento de instância XML. Declare um elemento como anyType quando ele contiver diferentes tipos de dados.

O esquema seguinte descreve uma pessoa com nome, sobrenome e um elemento age que é anyType:

```
<xs:element name="person">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="firstname" type="xs:string"/>
      <xs:element name="lastname" type="xs:string"/>
      <<xs:element name="age" type="xs:anyType"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

O seguinte documento de instância XML inclui um tipo de dados e um número no elemento age:

```
<person>
  <firstname>Danny</firstname>
  <lastname>Russell</lastname>
  <age>1959-03-03</age>
</person>
<person>
  <firstname>Carla</firstname>
  <lastname>Havers</lastname>
  <age>46</age>
</person>
```

Os dois tipos são válidos para o esquema. Se você não declarar um tipo de dados para um elemento no esquema, o elemento assumirá anyType como padrão quando você importar o esquema no Designer.

Elementos anySimpleType

Um elemento anySimpleType pode conter qualquer tipo atômico. Um tipo atômico é um tipo de dados básico, como Boolean, string, integer, decimal ou date.

O seguinte esquema descreve uma pessoa com nome, sobrenome e outro elemento que é anySimpleType:

```
<xs:element name="person">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="firstname" type="xs:string"/>
      <xs:element name="lastname" type="xs:string"/>
      <<xs:element name="other" type="xs:anySimpleType"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

O seguinte documento de instância XML substitui o elemento anySimpleType por um tipo de dados de string:

```
<person>
  <firstname>Kathy</firstname>
```

```

    <lastname>Russell</lastname>

    <other>Cissy</other>

</person>

```

O seguinte documento de instância XML substitui o elemento anySimpleType por um tipo de dados numérico:

```

<person>

    <firstname>Kathy</firstname>

    <lastname>Russell</lastname>

    <other>34</other>

</person>

```

Elementos de Conteúdo ANY

O elemento de conteúdo ANY aceita qualquer conteúdo em um arquivo XML. Ao declarar um elemento de conteúdo ANY em um esquema, você poderá substituí-lo por um elemento de qualquer nome e tipo em um documento de instância XML. O elemento substituto deve existir no esquema.

Ao especificar o conteúdo ANY, você usa a palavra-chave ANY, em vez de um nome de elemento e um tipo de elemento.

O esquema seguinte descreve uma pessoa com nome, sobrenome e um elemento que é conteúdo ANY:

```

<xs:element name="person">

    <xs:complexType>

        <xs:sequence>

            <xs:element name="firstname" type="xs:string"/>

            <xs:element name="lastname" type="xs:string"/>

            <xs:any minOccurs="0"/>

        </xs:sequence>

    </xs:complexType>

</xs:element>
<xs:element name="son" type="xs:string"/>

<xs:element name="daughter" type="xs:string"/>

```

O esquema inclui um elemento son e um elemento daughter. Você pode substituir o elemento ANY pelo elemento son ou daughter no documento de instância XML:

```

<person>

    <firstname>Danny</firstname>

    <lastname>Russell</lastname>

```

```

    <son>Atlee</son>

</person>
<person>

    <firstname>Christine</firstname>

    <lastname>Slade</lastname>

    <daughter>Susie</daughter>

</person>

```

Atributos AnyAttribute

O atributo anyAttribute aceita qualquer atributo em um arquivo XML. Ao declarar um atributo como anyAttribute, você pode substituir o elemento anyAttribute por qualquer atributo do esquema.

O seguinte esquema descreve uma pessoa com nome, sobrenome e um atributo que é anyAttribute:

```

<xs:element name="person">

  <xs:complexType>

    <xs:sequence>

      <xs:element name="firstname" type="xs:string"/>

      <xs:element name="lastname" type="xs:string"/>

    </xs:sequence>

    <xs:anyAttribute/>

  </xs:complexType>

</xs:element>

```

O seguinte esquema inclui um atributo gender:

```

    <xs:attribute name="gender">

      <xs:simpleType>

        <xs:restriction base="xs:string">

          <xs:pattern value="male|female"/>

        </xs:restriction>

      </xs:simpleType>

    </xs:attribute>

```

O seguinte documento de instância XML substitui anyAttribute pelo atributo gender:

```

<person gender="female">

```

```

    <firstname>Anita</firstname>

    <lastname>Ficks</lastname>

</person>

<person gender="male">

    <firstname>Jim</firstname>

    <lastname>Geimer</lastname>

</person>

```

Grupos de componentes

É possível criar os seguintes grupos de componentes em um esquema XML:

- **Grupo de elementos e atributos.** Grupo de elementos ou atributos aos quais você pode fazer referência em todo um esquema.
- **Grupo de substituição.** Grupo de elementos que você pode substituir por outros elementos do mesmo grupo.

Grupos de Elementos e Atributos

Você pode colocar elementos e atributos em grupos aos quais pode fazer referência em um esquema. Você deve declarar o grupo de elementos ou atributos antes de fazer referência ao grupo.

O exemplo a seguir mostra a sintaxe do esquema para um grupo de elementos:

```

<xs:group name="Songs">

    <xs:element name="songTitle" type="xs:string" />

    <xs:element name="artist" type="xs:string" />

    <xs:element name="publisher" type="xs:string" />

</xs:group>

```

O exemplo a seguir mostra a sintaxe do esquema para um grupo de atributos:

```

<xs:attributeGroup name="Songs">

    <xs:attribute name="songTitle" type="xs:string" />

    <xs:attribute name="artist" type="xs:string" />

    <xs:attribute name="publisher" type="xs:string" />

</xs:attributeGroup>

```

Os grupos de elementos a seguir oferecem restrições aos dados XML:

- **Grupo de sequência.** Todos os elementos em um arquivo XML devem ocorrer na ordem que o esquema os lista. Por exemplo, OrderHeader exige customerName primeiro, em seguida orderNumber e depois orderDate:

```
<xs:group name="OrderHeader">

  <xs:sequence>

    <xs:element name="customerName" type="xs:string" />

    <xs:element name="orderNumber" type="xs:number" />

    <xs:element name="orderDate" type="xs:date" />

  </xs:sequence>

</xs:group>
```

- **Grupo de escolha.** Um elemento no grupo pode ocorrer em um arquivo XML. Por exemplo, o grupo CustomerInfo lista uma escolha de elementos para o arquivo XML:

```
<xs:group name="CustomerInfo">

  <xs:choice>

    <xs:element name="customerName" type="xs:string" />

    <xs:element name="customerID" type="xs:number" />

    <xs:element name="customerNumber" type="xs:integer" />

  </xs:choice>

</xs:group>
```

- **Grupo Todos.** Todos os elementos devem ocorrer no arquivo XML ou nenhum. Os elementos podem ocorrer em qualquer ordem. Por exemplo, CustomerInfo exige todos ou nenhum dos três elementos:

```
<xs:group name="CustomerInfo">

  <xs:all>

    <xs:element name="customerName" type="xs:string" />

    <xs:element name="customerAddress" type="xs:string" />

    <xs:element name="customerPhone" type="xs:string" />

  </xs:all>

</xs:group>
```

Grupos de substituição

Use grupos de substituição para substituir um elemento por outro em um arquivo XML. Por exemplo, se tiver endereços do Canadá e dos Estados Unidos, você poderá criar um tipo de endereço para o Canadá e outro tipo para os Estados Unidos. Você pode criar um grupo de substituição que aceite qualquer tipo de endereço.

O seguinte fragmento de esquema mostra um tipo base Address e os tipos derivados CAN_Address e USA_Address:

```
<xs:complexType name="Address">
  <xs:sequence>
    <xs:element name="Name" type="xs:string" />
    <xs:element name="Street" type="xs:string"
      minOccurs="1" maxOccurs="3" />
    <xs:element name="City" type="xs:string" />
  </xs:sequence>
</xs:complexType>
<xs:element name="MailAddress" type="Address" />
<xs:complexType name="CAN_Address">
  <xs:complexContent>
    <xs:extension base="Address">
      <xs:sequence>
        <xs:element name="Province" type="xs:string" />
        <xs:element name="PostalCode" type="CAN_PostalCode"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="USA_Address">
  <xs:complexContent>
    <xs:extension base="Address">
      <xs:sequence>
        <xs:element name="State" type="USPS_StateCode" />
        <xs:element name="ZIP" type="USPS_ZIP"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:element name="AddrCAN" type="CAN_Address"
  substitutionGroup="MailAddress"/>
<xs:element name="AddrUSA" type="USA_Address"
```

```
substitutionGroup="MailAddress"/>
```

CAN_Address inclui Province e PostalCode, e USA_Address inclui State e Zip. O grupo de substituição MailAddress inclui os dois tipos de endereços.

TÓPICOS RELACIONADOS:

- [“Usando grupos de substituição em uma definição XML” na página 50](#)

XML Path

O XPath (XPath) é uma linguagem que descreve uma forma de localizar itens em um arquivo XML. O XPath usa uma sintaxe de endereçamento baseada na rota por meio da hierarquia da raiz até um elemento ou um atributo. Um caminho XML pode conter nomes de componente de esquema longo.

O XPath usa uma barra invertida (/) para distinguir entre elementos na hierarquia. Os atributos XML são precedidos por “@” no XPath.

Você pode criar uma consulta em um elemento ou um atributo XPath para filtrar dados XML.

TÓPICOS RELACIONADOS:

- [“Usando Predicados de Consulta do XPath” na página 54](#)

Páginas de Código

Os arquivos XML contêm uma declaração de codificação que indica a página de código usada no arquivo. As páginas de código mais comuns no XML são UTF-8 e UTF-16. Todos os analisadores de XML oferecem suporte a essas páginas de código. Para obter informações sobre a especificação da codificação de caracteres XML, consulte o site do W3C em <http://www.w3c.org>.

O PowerCenter oferece suporte ao mesmo conjunto de páginas de código para arquivos XML que ele aceita para bancos de dados relacionais e outros arquivos simples. O PowerCenter não oferece suporte a uma página de código definida pelo usuário.

Quando você cria uma definição de origem ou de destino XML, o Designer atribui à definição a página de código do Cliente do PowerCenter. Se você importar um esquema XML que contenha uma atribuição de página de código, o Assistente de XML exibirá a página de código do esquema. No entanto, o Assistente de XML não aplica essa página de código à definição XML que você cria no repositório.

Você não pode configurar a página de código para uma definição de origem XML. O Serviço de Integração converte arquivos de origem XML em Unicode quando os analisa.

Você pode configurar a página de código para uma definição XML de destino no Designer. Você também pode alterar a página de código para uma instância de destino XML nas propriedades da sessão.

Usando XML com o PowerCenter

Visão Geral do Uso do XML com o PowerCenter

Você pode criar uma definição XML no PowerCenter a partir de um arquivo XML, um arquivo DTD, um esquema XML, uma definição de arquivo simples ou uma definição de tabela relacional. Quando você cria uma definição XML, o Designer extrai os metadados XML e cria um esquema no repositório. O esquema fornece a estrutura a partir da qual você edita e valida a definição XML.

Uma definição XML pode conter vários grupos. Em uma definição XML, os grupos são chamados exibições. O relacionamento entre os elementos na hierarquia XML define o relacionamento entre as exibições.

Quando você cria uma definição XML, o Designer cria exibições de elementos com várias ocorrências e tipos complexos em um esquema por padrão. A cardinalidade relativa de elementos em uma hierarquia XML afeta como o PowerCenter cria exibições em uma definição XML. A cardinalidade relativa determina se os elementos podem fazer parte da mesma exibição.

O Designer define relacionamentos entre as exibições em uma definição XML por chaves. As definições de origem não necessitam de chaves, mas as exibições de destino devem tê-las. Cada exibição tem uma chave primária que é um elemento XML ou uma chave gerada.

Ao criar uma definição XML, você pode criar um modelo hierárquico ou um modelo de relacionamento de entidade dos dados XML. Ao criar um modelo hierárquico, você cria uma hierarquia normalizada ou desordenada. Uma hierarquia normalizada contém exibições separadas para elementos com várias ocorrências. Uma hierarquia desordenada tem uma exibição com dados duplicados para elementos com várias ocorrências.

Se você criar um modelo de entidade, o Designer criará exibições para tipos complexos e elementos com várias ocorrências. O Designer cria uma definição XML que modela os relacionamentos circulares e de herança que o esquema fornece.

O PowerCenter pode trabalhar com esquemas XML contendo menos de 400 elementos. O perfil do PowerCenter pode conter até três níveis de hierarquia e incluir os seguintes elementos de tipo complexo:

- Sequência
- Qualquer
- Opção

O assistente de importação de XML do PowerCenter pode criar até 400 exibições.

Limitações

As seguintes limitações se aplicam ao tratamento de XML no PowerCenter:

- O esquema XML deve ter menos de 400 elementos.
- O arquivo de esquema XML deve ter menos de 100 KB.
- O arquivo XML deve ter 10 MB ou menos.
- O perfil de complexidade está limitado a três níveis de hierarquia.
- O Assistente de Importação de XML cria no máximo 400 exibições.

O PowerCenter não oferece suporte às seguintes funções:

- **Colunas concatenadas.** Uma coluna não pode ser uma concatenação de dois elementos. Por exemplo, você não pode criar uma coluna FULLNAME que se refira a uma concatenação de dois elementos FIRSTNAME e LASTNAME.
- **Chaves compostas.** Uma chave não pode ser uma concatenação de dois elementos. Por exemplo, você não pode criar uma chave CUSTOMERID que se refira a uma concatenação de dois elementos LASTNAME e PHONENUMBER.
- **Listas analisadas.** O PowerCenter armazena um tipo de lista como uma cadeia que contém todos os elementos de matriz. O PowerCenter não analisa os respectivos tipos simples das strings.

Para criar uma transformação com outros tipos de elementos e para transformar arquivos de entrada XML maiores, use uma transformação de Processador de Dados. Para obter mais informações sobre como criar transformações de Processador de Dados, consulte o *Guia do Usuário do Informatica Data Transformation* e o *Guia de Introdução do Informatica Data Transformation*.

Importando Metadados XML

Quando você importa uma definição XML, o Designer cria um esquema da definição no repositório. O esquema de repositório fornece a estrutura a partir da qual você edita e valida a definição XML.

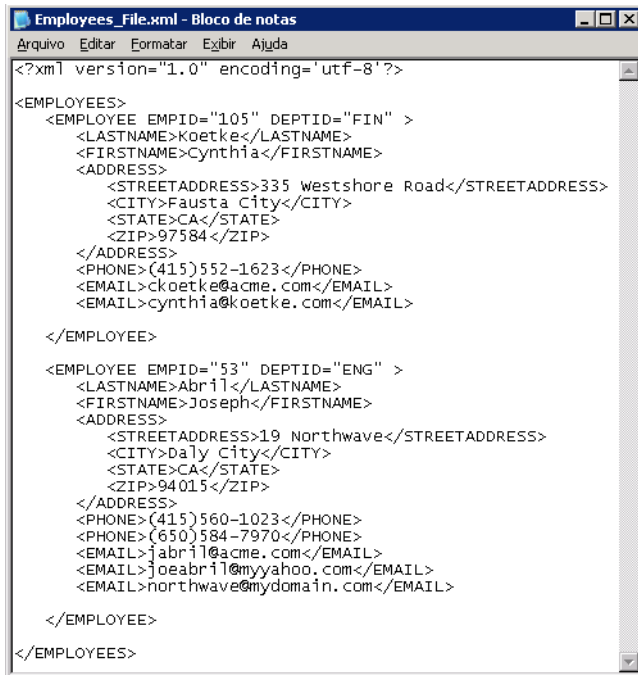
É possível criar metadados a partir dos seguintes tipos de arquivos:

- Arquivos XML
- Arquivos DTD
- Arquivos de esquema XML
- Tabelas relacionais
- Arquivos simples

Importando metadados de um arquivo XML

Em um arquivo XML, um par de marcas indica o começo e o fim de cada elemento de dados. Essas marcas são a base para os metadados que o PowerCenter extrai do arquivo XML. Se você importar um arquivo XML sem um DTD ou esquema XML associado, o Designer lerá as marcas XML para determinar os elementos, suas possíveis ocorrências e sua posição na hierarquia. O Designer verifica os dados dentro das marcas do elemento e atribui um tipo de dados dependendo da representação dos dados. Você pode alterar os tipos de dados para esses elementos na definição XML.

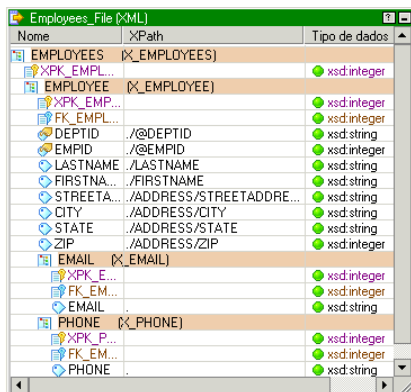
A figura a seguir mostra um exemplo de arquivo XML:



O elemento raiz é Funcionários. Funcionário é um elemento com várias ocorrências. O elemento Funcionário contém LastName, FirstName e Endereço. O elemento Funcionário também contém elementos com várias ocorrências: Telefone e E-mail.

O Designer determina uma estrutura de esquema dos dados XML.

A figura a seguir mostra a definição de origem XML padrão com exibições separadas para o elemento raiz e os elementos com várias ocorrências:



Quando importa um arquivo XML, você não precisa de todos os dados XML para criar uma definição XML. Você precisa de dados suficientes para mostrar de forma precisa a hierarquia do arquivo XML.

O Designer pode criar uma definição XML a partir de um arquivo XML que faça referência a um arquivo DTD ou esquema XML. Se um arquivo XML tiver uma referência para um esquema DTD ou XML em outro nó, o nó que hospeda o Cliente do PowerCenter deve ter acesso ao nó onde o esquema reside de forma que o Designer possa ler o esquema. O arquivo XML contém um URI (Identificador Universal de Recursos), que é o endereço do DTD ou o esquema XML.

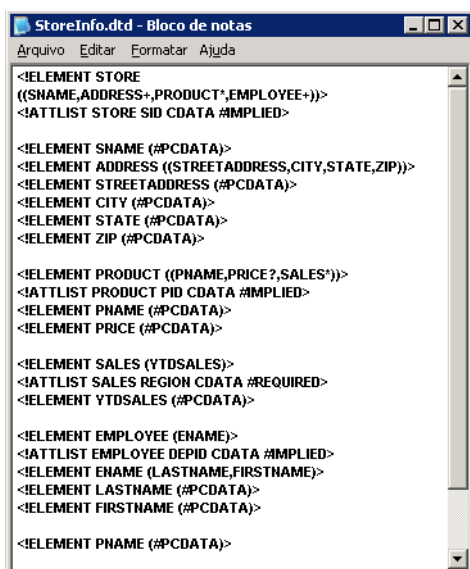
Importando metadados de um arquivo DTD

Um arquivo DTD fornece restrições em uma estrutura de documento XML. Um arquivo DTD lista elementos, atributos, entidades e notações para um documento XML. Um arquivo DTD especifica relacionamentos entre componentes. Um DTD especifica cardinalidade e restrição nula. Porém, um arquivo DTD não contém dados nem tipos de dados.

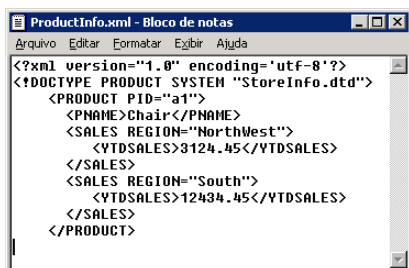
Ao importar um arquivo DTD, você pode alterar os tipos de dados para os elementos na definição XML. Você pode alterar a restrição nula, mas não pode alterar a cardinalidade do elemento.

Se importar um arquivo XML com um DTD associado, o Designer criará uma definição com base na estrutura do DTD.

A figura a seguir mostra um exemplo de um arquivo XML em que StoreInfo.dtd contém o elemento Loja e Produto é um dos elementos filhos de Loja:

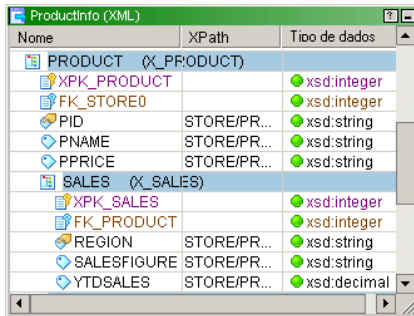


A figura a seguir mostra o DTD associado:



O DTD associado, ProductInfo.xml, usa o elemento Produto de StoreInfo.dtd. Produto inclui o elemento Vendas com várias ocorrências.

A figura a seguir mostra a definição de origem que o Designer cria:



Nome	XPath	Tipo de dados
PRODUCT (X_PRODUCT)		
PK_PRODUCT		xsd:integer
FK_STORE0		xsd:integer
PID	STORE/PR...	xsd:string
PNAME	STORE/PR...	xsd:string
PPRICE	STORE/PR...	xsd:string
SALES (X_SALES)		
PK_SALES		xsd:integer
FK_PRODUCT		xsd:integer
REGION	STORE/PR...	xsd:string
SALESFIGURE	STORE/PR...	xsd:string
YTD SALES	STORE/PR...	xsd:decimal

A definição ProductInfo contém os grupos Produto e Vendas. O arquivo XML determina quais elementos serão incluídos na definição. O arquivo DTD determina a estrutura da definição de XML.

Importando Metadados de um Esquema XML

Um arquivo de esquema define a estrutura de elementos e atributos em um arquivo XML. Um arquivo de esquema contém descrições do tipo de elementos e atributos no arquivo. Quando você importa um esquema XML, o Designer determina o tipo de dados, a precisão e a cardinalidade dos elementos. Você não poderá alterar uma definição de elemento no PowerCenter se ela for proveniente de um esquema.

Quando você importa metadados de um esquema XML, o arquivo .xsd pode conter instruções de importação e de inclusão que fazem referência a outros arquivos .xsd. Quando você importa um esquema que inclui outros esquemas, os outros esquemas não devem fazer referência ao mesmo espaço de nome.

Exemplo:

```
<IMPORT
  schemaLocation="../../../administration/process/bo/LocationTextBO.xsd"
  namespace="http://EnterpriseLibrary/com/acs/enterprise/common/program/administration/
  process/bo">
<IMPORT
  schemaLocation="../../../administration/process/bo/LineOfBusinessBO.xsd"
  namespace="http://EnterpriseLibrary/com/acs/enterprise/common/program/administration/
  process/bo">
<IMPORT
  schemaLocation="../../../administration/process/bo/ClaimExceptionBO.xsd"
  namespace="http://EnterpriseLibrary/com/acs/enterprise/common/program/administration/
  process/bo">
```

Você pode substituir várias instruções "import schemalocation" com uma instrução:

```
<xsd:import schemalocation="imported.xsd" namespace=" http://EnterpriseLibrary/com/acs/enterprise/
common/program/administration/process/bo" />
```

O arquivo imported.xsd inclui os outros arquivos XSD usando a seguinte sintaxe:

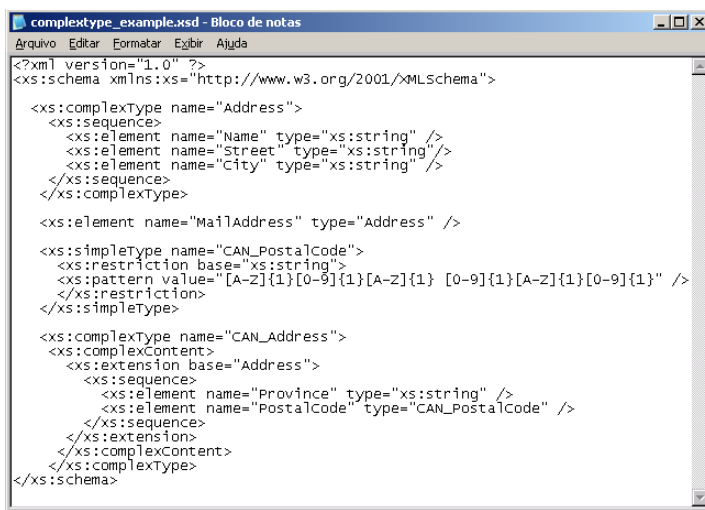
```
<xsd:schema targetNamespace="http://EnterpriseLibrary/com/acs/enterprise/common/
program/administration/process/bo" elementFormDefault="qualified" >
  <xsd:include schemaLocation=" LocationTextBO.xsd" />
  <xsd:include schemaLocation=" LineOfBusinessBO.xsd" />
  <xsd:include schemaLocation=" ClaimExceptionBO.xsd" />
</xsd:schema>
```

Para obter mais informações, consulte o artigo 158334 da Base de Dados de Conhecimento.

Cada definição de tipo simples em um esquema XML é uma restrição de outra definição de tipo simples nos tipos de dados Atômicos do esquema, como Booleano, cadeia ou número inteiro, que restringem o tipo de dados anySimpleType. Ao definir um tipo de dados simples em um esquema XML, você deriva um novo tipo de dados de um tipo de dados existente. Por exemplo, você pode derivar um tipo integer restrito que retém apenas números de 1 a 20. O tipo base é integer.

Ao derivar um tipo de dados complexo de outro tipo de dados, você cria um novo tipo de dados que contém os elementos do tipo base. Você pode adicionar novos elementos ao tipo derivado ou criar restrições nos elementos herdados. O Designer cria exibições para tipos derivados sem duplicar as colunas que representam os componentes herdados. Isso reduz os metadados e diminui o tamanho da definição XML no repositório.

A seguinte imagem mostra um esquema com tipos derivados simples e complexos:



O elemento MailAddress é um tipo de endereço que é um tipo complexo. Um tipo derivado, CAN_Address, herda o Name, City e Street do tipo Address e estende Address adicionando Province e PostalCode. PostalCode é um tipo simples chamado CAN_PostalCode.

Quando você importa um esquema XML, todos os tipos ou atributos simples em um tipo complexo podem se tornar uma coluna em uma definição XML. Os tipos complexos se tornam exibições.

A seguinte figura mostrará uma definição XML do esquema se você importar o esquema com as opções padrão:

Nome	Tipo de dados
MailAddress (X_MailAddress)	
XPX_MailAddress	xsd:integer
MailAddress	Address
Address (X_Address)	
XPX_Address	xsd:integer
Name	xsd:string
Street	xsd:string
City	xsd:string
CAN_Address (X_CAN_Address)	
XPX_CAN_Address	xsd:integer
FK_Address	xsd:integer
Province	xsd:string
PostalCode	CAN_PostalCode

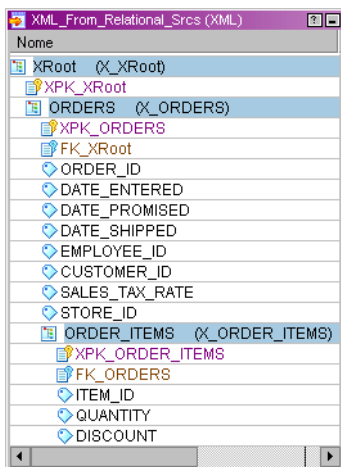
A exibição CAN_Address contém os elementos exclusivos para seu tipo. O elemento raiz é MailAddress. O tipo Address contém Name, Street e City. O CAN_Address tem uma chave externa para Address. CAN_Address inclui Province e PostalCode.

A exibição não contém Name, Street e City que ela herda de MailAddress.

Criando Metadados a partir de Definições Relacionais

Você pode criar uma definição XML selecionando várias definições relacionais e criando relacionamentos entre elas. O Designer cria uma exibição XML para cada definição relacional importada. O Designer converte todas as colunas na definição relacional e gera relacionamentos de chave primária para chave externa. Você pode optar por criar uma raiz de exibição.

A figura a seguir mostra um exemplo de definição de destino XML das definições relacionais, Pedidos e Order_Items:

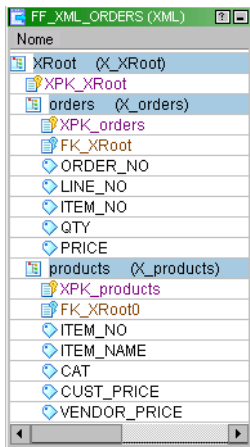


A raiz é XRoot. XRoot inclui Pedidos e Itens de Pedidos. Order_Items tem uma chave externa que aponta para Pedidos.

Criando Metadados de Arquivos Simples

Você pode criar uma definição XML importando uma definição de arquivo simples do repositório. Se você importar mais de uma definição de arquivo simples, o Designer criará uma definição XML com uma exibição para cada arquivo simples. As exibições não têm relacionamento entre si na definição XML. Se você optar por criar uma exibição raiz, o Designer criará as exibições com chaves externas para a raiz.

A figura a seguir mostra um exemplo de definição de origem XML dos arquivos simples pedidos e produtos:



Produtos e Pedidos têm uma chave externa para a exibição raiz e são realçados.

Noções Básicas sobre as Exibições XML

O relacionamento entre elementos na hierarquia XML define o relacionamento entre exibições XML em uma definição do PowerCenter. Em uma definição de origem, uma exibição não deve estar relacionada a nenhuma outra exibição. Portanto, as exibições em uma definição de origem não exigem chaves primárias ou externas. Uma exibição desordenada pode ser independente de qualquer outra exibição. No entanto, o Designer gerará chaves se você não designar colunas de chaves quando as exibições estiverem relacionadas a outras exibições.

Cada exibição em uma definição de destino deve estar relacionada a, pelo menos, um outro grupo. Portanto, cada exibição precisa de, pelo menos, uma chave para estabelecer seu relacionamento com outra exibição. Se você não designar as chaves, o Designer gerará chaves primárias e externas nas exibições de destino. Você poderá definir chaves primárias e externas para exibições se criar as exibições e os relacionamentos no Editor de XML, em vez de permitir que o Designer faça isso para você.

Quando o Designer cria uma coluna de chaves primárias ou externas, ele atribui um nome de coluna com um prefixo. Em uma definição XML, os prefixos são XPK_ para uma coluna de chaves primárias geradas e XFK_ para uma coluna de chaves externas geradas. O Designer usa o prefixo FK_ para uma chave externa que aponta para uma chave primária.

Por exemplo, quando o Designer cria uma coluna de chave primária para o grupo Vendas, o nome atribuído à coluna é XPK_Sales. Quando o Designer cria uma coluna de chaves externas que conecta um grupo de vendas a outro, o nome atribuído à coluna é XFK_Sales. Você poderá renomear qualquer nome de coluna que o Designer criar.

Se um mapeamento contiver uma origem XML, o Serviço de Integração criará o valor para as colunas de chaves primárias geradas na definição de origem quando você executar a sessão. É possível configurar valores iniciais para as chaves geradas.

Criando exibições XML personalizadas

Exibições personalizadas são grupos que você cria com o Assistente de XML ou o Editor de XML. Se você usar o Assistente de XML para criar exibições personalizadas, o assistente criará as exibições que contêm todos os componentes no esquema. Se usar o Editor de XML, você poderá definir cada exibição e escolher os componentes.

Os elementos nas exibições e o relacionamento entre as exibições dependem do esquema que o Designer cria no repositório quando você importa a definição. O Editor de XML valida as definições XML usando as regras para exibições válidas.

Regras e Diretrizes para Exibições XML

Considere as seguintes regras e diretrizes quando você trabalha com chaves de exibição e relacionamentos:

- Uma definição XML do PowerCenter pode ter até 400 exibições.
- Uma exibição pode ter uma chave primária.
- Uma exibição pode ser relacionada a várias outras exibições e uma exibição pode ter várias chaves externas.
- Uma coluna não pode ser uma chave primária e uma chave externa.
- Uma exibição em uma definição de origem não requer uma chave.
- Uma exibição em uma definição de destino requer, pelo menos, uma chave.
 - A exibição raiz de destino requer uma chave primária, mas a raiz de destino não requer uma chave externa.
 - Uma exibição folha de destino requer uma chave externa, mas ela não requer uma chave primária.
- Um elemento com compartimentos não pode ser uma chave.
- Uma chave externa sempre se refere a uma chave primária em outro grupo. Você não pode usar chaves de autorreferência.
- Uma coluna de chave externa gerada sempre se refere a uma coluna de chave primária gerada.
- A cardinalidade relativa de elementos em uma hierarquia XML afeta como o PowerCenter cria exibições em uma definição XML. As seguintes regras determinam quando os elementos podem fazer parte da mesma exibição:
 - Elementos que têm um relacionamento um para um podem fazer parte da mesma exibição.
 - Elementos que têm um relacionamento um para muitos podem fazer parte da mesma exibição normalizada ou desordenada.
 - Os elementos que têm relacionamentos muitos para muitos não podem ser parte da mesma exibição.

Noções Básicas de Relacionamentos Hierárquicos

Uma definição XML com relacionamentos da exibição hierárquica tem cada elemento na hierarquia que aparece sob seu elemento pai em uma exibição. Os elementos com várias ocorrências podem ser tornar exibições. Os tipos complexos não se tornam exibições e os elementos exclusivos para tipos complexos derivados não ocorrem em qualquer exibição.

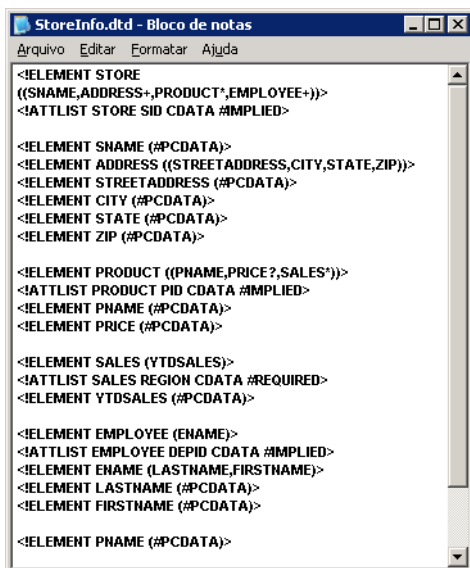
Você pode gerar os seguintes tipos de exibição hierárquica:

- **Exibições normalizadas.** Uma definição XML com exibições normalizadas reduz a redundância separando dados com várias ocorrências em exibições distintas. As exibições são relacionadas por chaves primárias e externas.
- **Exibições desordenadas.** Uma definição XML com uma exibição desordenada tem todos os elementos da hierarquia que são exclusivos para os tipos complexos derivados na exibição. Uma definição de origem ou de destino pode conter uma exibição desordenada.

Exibições normalizadas

Quando o Designer gera uma exibição normalizada, ele estabelece o elemento raiz e os elementos com várias ocorrências que se tornam exibições em uma definição XML.

A seguinte figura mostra um arquivo DTD e os elementos que se tornam exibições em uma definição XML normalizada:



Loja é o elemento raiz. Endereço, produto, funcionário e vendas são os elementos com várias ocorrências.

A seguinte figura mostra uma definição de origem baseada no arquivo DTD da figura acima:

Nome	XPath	Tipo de dados
STORE (X_STORE)		
XPK_STORE		xsd:integer
SID	STORE/@SID	xsd:string
SNAME	STORE/SN...	xsd:string
ADDRESS (X_ADDRESS)		
XPK_AD...		xsd:integer
FK_STO...		xsd:integer
STREET...	STORE/AD...	xsd:string
CITY	STORE/AD...	xsd:string
STATE	STORE/AD...	xsd:string
ZIP	STORE/AD...	xsd:string
PRODUCT (X_PRODUCT)		
XPK_PR...		xsd:integer
FK_STO...		xsd:integer
PID	STORE/PR...	xsd:string
PNAME	STORE/PR...	xsd:string
PPRICE	STORE/PR...	xsd:string
SALES (X_SALES)		
XPK_S...		xsd:integer
FK_P...		xsd:integer
REGION	STORE/PR...	xsd:string
SALES...	STORE/PR...	xsd:string
EMPLOYEE (X_EMPLOYEE)		
XPK_EM...		xsd:integer
FK_STO...		xsd:integer
DEPID	STORE/EM...	xsd:string
LASTNA...	STORE/EM...	xsd:string

A definição tem exibições normalizadas. A exibição raiz é Loja. As exibições Endereço, Produto e Vendas têm chaves externas para Loja. A exibição Vendas tem uma chave externa para a exibição Produto.

A tabela a seguir mostra as linhas em uma visualização de dados de exibição Loja:

XPK_si_STORE	si_SID	si_NAME
1	BE1752	Mud and Sawdust Furniture Store

A tabela a seguir mostra as linhas em uma visualização de dados para a exibição Endereço:

XPK_si_ADDRESS	FK_si_ADDRESS	si_STREETADDRESS	si_CITY	si_STATE	si_ZIP
1	1	335 Westshore Road	Fausta City	CA	95784
2	1	7415 Endless Loop	Codeville	CA	97412

A tabela a seguir mostra as linhas em uma visualização de dados para a exibição Produto:

XPK_si_PRODUCT	FK_si_PRODUCT	si_PNAME	si_PRICE	si_PID
1	1	Cadeira	5690.00	A1
1	1	Tabela	1240.00	a2
1	1	Cama	1364.99	a3

A tabela a seguir mostra as linhas em uma visualização de dados para a exibição Vendas:

XPK_si_SALES	FK_si_SALES	si_REGION	si_YTDSALES
1	a1	Noroeste	4565.44
2	a2	Sul	8793.99
3	a3	Oriental	23110.00
4	a4	Sul	5500.00
5	a5	Noroeste	10095.34
6	a6	Oriental	200.00

A tabela a seguir mostra as linhas em uma visualização de dados para a exibição Funcionário:

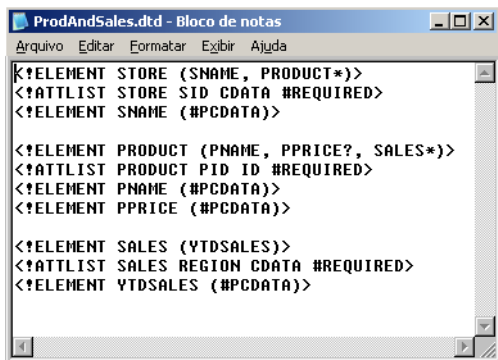
XPK_si_EMPLOYEE	FK_si_EMPLOYEE	si_FIRSTNAME	si_LASTNAME
1	1	James	Bond
2	1	Austin	Powers
3	1	Indiana	Jones
4	1	Foxie	Marrom
5	1	Bonnie	Bell
6	1	Laura	Croft

Exibições desordenadas

Quando o Designer gera uma exibição desordenada, ele cria uma exibição e coloca nela todos os elementos da hierarquia. Todos os elementos em uma exibição desordenada pertencem à mesma cadeia pai. Exibições desordenadas, como tabelas desordenadas, geram dados duplicados.

O Designer poderá gerar exibições desordenadas para definições XML que contenham mais de um elemento com várias ocorrências se os elementos com várias ocorrências tiverem um relacionamento um para muitos ou se todos fizerem parte da mesma cadeia pai.

A figura a seguir mostra um arquivo DTD que contém elementos com várias ocorrências, nesse caso Product e Sales:



Product e Sales são elementos com várias ocorrências. Como os elementos com várias ocorrências têm um relacionamento um para muitos, o Designer pode criar uma só exibição desordenada que inclua todos os elementos.

A seguinte figura mostra a exibição desordenada para ProdAndSales.dtd em uma definição de origem:

Nome	XPath	Tipo de dados
SALES (X_STORE)		
YTDSALES	./YTDSALES	xsd:string
REGION	./REGION	xsd:string
PRICE	./PRICE	xsd:string
PNAME	./PNAME	xsd:string
PID	./PID	xsd:string
SNAME	./SNAME	xsd:string
SID	./SID	xsd:string

O Designer cria uma única exibição para todos os elementos na hierarquia ProdAndSales. Como um arquivo DTD não define tipos de dados, o Designer atribui um tipo de dados de string a todas as colunas. A exibição desordenada não precisa de uma chave primária ou externa.

A seguinte figura mostra uma visualização de dados para a exibição desordenada:

SID	SNAME	PID	PNAME	PPRICE	REGION	YTDSALES
BE1752	Mud and Sawdust Furniture Store	a1	Chair	NULL	NorthWest	3124.45
BE1752	Mud and Sawdust Furniture Store	a1	Chair	NULL	South	12434.45
BE1752	Mud and Sawdust Furniture Store	a2	Table	4126.33	South	8252.66
BE1752	Mud and Sawdust Furniture Store	a3	Bed	4356.33	NorthWest	43563.30
BE1752	Mud and Sawdust Furniture Store	a3	Bed	4356.33	South	21781.65
BE1752	Mud and Sawdust Furniture Store	a3	Bed	4356.33	East	26137.98
BE1752	Mud and Sawdust Furniture Store	a4	Etagere	5000.00	South	20000.00
BE1752	Mud and Sawdust Furniture Store	a4	Etagere	5000.00	East	5000.00
BE1752	Mud and Sawdust Furniture Store	a4	Etagere	5000.00	NorthWest	15000.00

Noções Básicas de Relacionamentos de Entidade

Você pode criar relacionamentos de entidade a partir de um esquema XML. Quando você cria uma definição XML que contém relacionamentos de entidade, o Designer gera exibições separadas para elementos com várias ocorrências, grupos de elementos e tipos complexos. O Designer inclui exibições para todos os tipos complexos derivados. O Designer cria links e chaves entre as exibições com base em relacionamentos de hierarquia e tipo.

Ao trabalhar com esquemas XML, você pode fazer referência a partes do esquema, em vez de repetir as mesmas informações em componentes de esquema. Um componente pode herdar os elementos e os atributos de outro componente e restringir ou estender os elementos do componente. Por exemplo, você pode usar um tipo complexo como uma base para criar um novo tipo complexo. É possível adicionar mais elementos ao novo tipo para criar um tipo complexo estendido. Ou você pode criar um tipo complexo restrito, que é um subconjunto de outro tipo complexo.

Se criar exibições manualmente ou recriar relacionamentos de entidade no Editor de XML, você optará como deseja estruturar os metadados. Ao criar uma definição XML baseada em um esquema XML que usa herança, você pode gerar exibições separadas para o tipo base e o tipo derivado. Você poderá criar relacionamentos de herança se planejar mapear os dados XML para tabelas relacionais normalizadas.

Um relacionamento de herança Tipo I de XML é um relacionamento entre duas exibições. Cada raiz de exibição é um tipo complexo global. Uma exibição é derivada da outra.

Você pode criar um relacionamento de herança entre uma coluna e uma exibição. Este é um relacionamento de herança Tipo II de XML.

O Designer gera exibições separadas de grupos de substituição.

Regras e Diretrizes para Relacionamentos de Entidade

O Designer gera entidades baseadas nas diretrizes a seguir:

- Uma entidade representa uma parte de uma hierarquia XML, DTD ou esquema XML. Essa hierarquia não precisa começar na raiz do arquivo XML.
- O Designer usa entidades definidas em um arquivo DTD para criar relacionamentos de entidade.
- O Designer usa estruturas de tipo definidas em um esquema XML para gerar relacionamentos de entidade.
- O Designer cria uma nova entidade quando encontra um elemento de várias ocorrências em um elemento pai.
- O Designer gera uma exibição separada para cada membro de um grupo de substituição.
- O Designer gera chaves primárias e chaves externas para relacionar entidades separadas.

Exemplo de Relacionamento de Entidade Tipo 1

Um relacionamento de entidade Tipo 1 de XML é um relacionamento entre duas exibições. Cada exibição deve ser raiz em um tipo complexo global. Uma exibição deve ser derivada da outra.

O esquema seguinte contém PublicationType, BookType e MagazineType. PublicationType é o tipo de base. Uma publicação contém Title, Author e Date. BookType e MagazineType são tipos derivados que estendem PublicationType. Livro tem ISBN e Editor, e Revista tem Volume e Edição.

```
<xsd:complexType name="PublicationType">
  <xsd:sequence>
    <xsd:element name="Title" type="xsd:string"/>
    <xsd:element name="Author" type="xsd:string" maxOccurs="unbounded"/>
    <xsd:element name="Date" type="xsd:string"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:element name="Publication" type="PublicationType"/>
<xsd:complexType name="BookType">
  <xsd:complexContent>
```

```

<xsd:extension base="PublicationType">
<xsd:sequence>
  <xsd:element name="ISBN" type="xsd:string"/>
  <xsd:element name="Publisher" type="xsd:string"/>
</xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="MagazineType">
<xsd:complexContent>
<xsd:extension base="PublicationType">
<xsd:sequence>
  <xsd:element name="Volume" type="xsd:string"/>
  <xsd:element name="Edition" type="xsd:string"/>
</xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>
</xsd:schema>

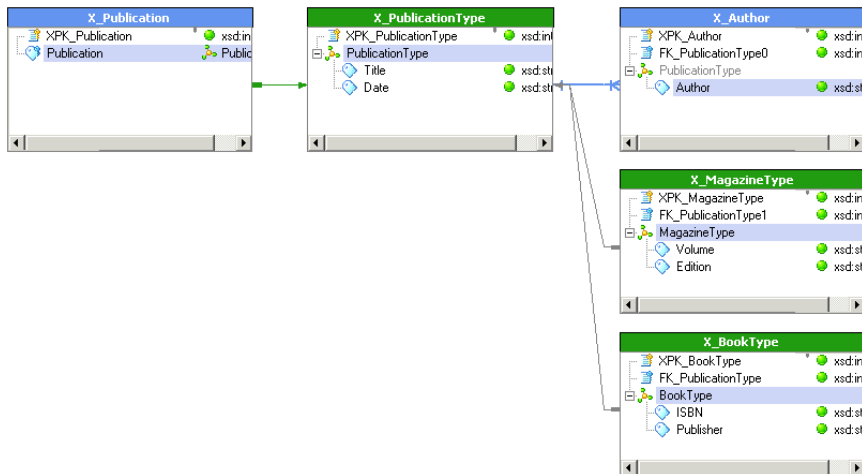
```

Quando você cria exibições XML como entidades em uma definição XML, os metadados de Title e Date de PublicationType não se repetem em BookType ou MagazineType. Em vez disso, essas exibições contêm os metadados que as distinguem de PublicationType: ISBN e Editor para BookType e volume e edição para MagazineType. Elas têm chaves externas que as vinculam a PublicationType.

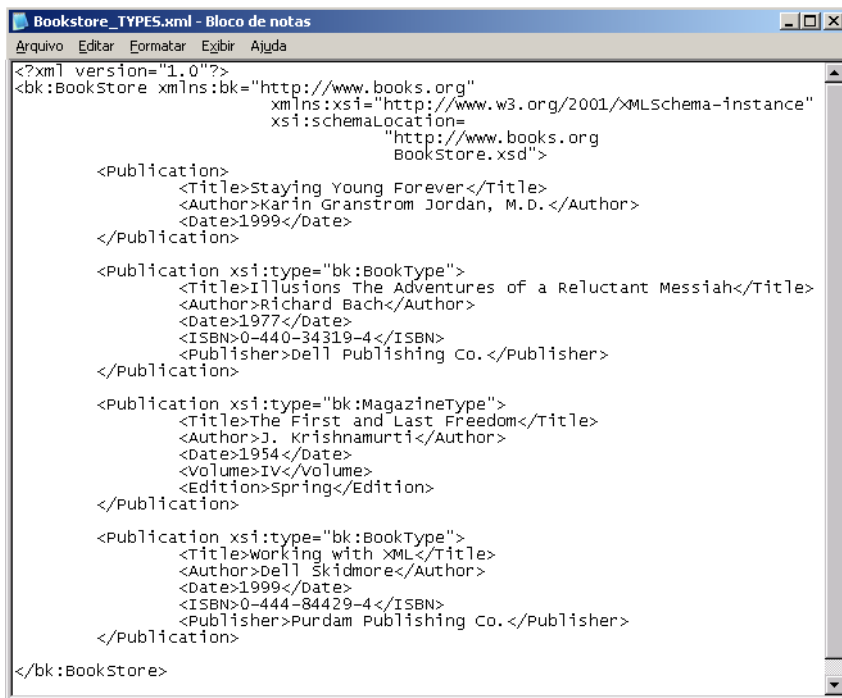
Esse exemplo usa explosão de metadados reduzidos porque nenhum dos elementos do tipo de base é repetido nos tipos derivados.

Autor é um elemento que ocorre várias vezes em Publicação. Author se torna uma exibição XML.

A figura a seguir mostra as exibições padrão que o Designer gera a partir do esquema:



A figura a seguir mostra um arquivo XML que tem uma publicação, uma revista e livros:



Se processar o arquivo XML de exemplo usando a definição XML da figura precedente, você criará dados nas seguintes exibições:

- **Exibição PublicationType.** Contém o título e a data de cada publicação.

A figura a seguir mostra a exibição PublicationType:

XPK_boa_PublicationType	Title	Date
1	Staying Yo...	1999
2	Illusions Th...	1977
3	The First a...	1954
4	Working wi...	1999

- **Exibição BookType.** Contém o ISBN e o editor. BookType contém uma chave externa para PublicationType.

A figura a seguir mostra a exibição BookType:

XPK_boa_BookType	FK_boa_PublicationType1	ISBN	Publisher
1	2	0-440-34319-4	Dell Publishing Co.
2	4	0-444-84429-4	Purdam Publishing Co.

- **Exibição MagazineType.** Contém volume e edição. MagazineType também contém uma chave externa para o PublicationType.

A figura a seguir mostra a exibição MagazineType:

XPK_boa_MagazineType	FK_boa_PublicationType	Volume	Edition
1	3	IV	Spring

- **Exibição Author.** Contém autores de todas as publicações. O Designer gera uma exibição separada para Autor porque Autor é um elemento com várias ocorrências. Cada publicação pode conter vários autores.

A figura a seguir mostra a exibição Autor:

XPK_Author	FK_boo_PublicationType0	Author
1	1	Karin Granstrom Jordan, M.D.
2	2	Richard Bach
3	3	J. Krishnamurti
4	4	Dell Skidmore

Exemplo de Relacionamento de Entidade Tipo II

Você pode criar um relacionamento de herança entre uma coluna e uma exibição de tipo complexo. A coluna deve ser um elemento de um tipo complexo local. A exibição deve ser a raiz em um tipo complexo global. O tipo complexo local deve ser derivado do tipo complexo global.

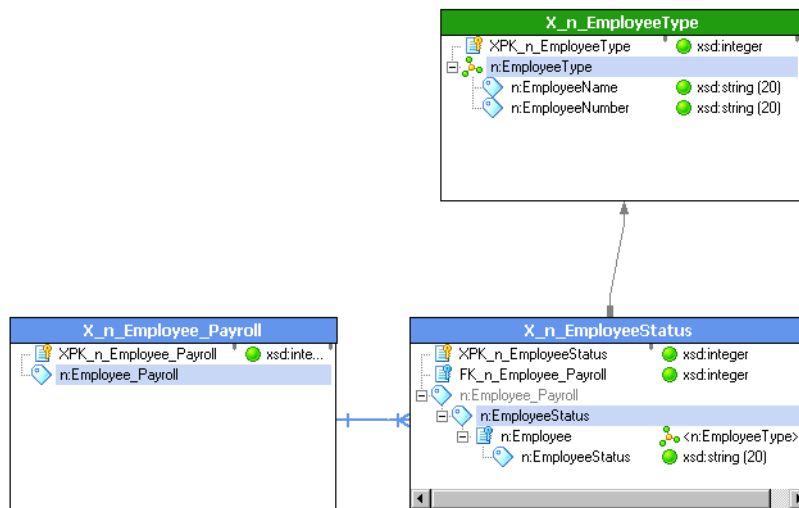
Por exemplo, o esquema a seguir define um tipo complexo chamado EmployeeType. EmployeeType contém os elementos EmployeeNumber e EmployeeName.

EmployeeStatusType inclui um elemento chamado Employee que estende EmployeeType. Employee inclui um elemento EmployeeStatus.

```
<xs:element name="Employee_Payroll">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="EmployeeStatus" type="EmpStatusType"
        maxOccurs="unbounded"></xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:complexType name="EmpStatusType">
  <xs:sequence>
    <xs:element name="Employee" minOccurs="0" maxOccurs="1">
      <xs:complexType>
        <xs:complexContent>
          <xs:extension base="EmployeeType">
            <xs:sequence>
              <xs:element name="EmployeeStatus" type="xs:string">
            </xs:element>
            </xs:sequence>
          </xs:extension>
        </xs:complexContent>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="EmployeeType">
  <xs:sequence>
    <xs:element name="EmployeeName" type="xs:string"></xs:element>
    <xs:element name="EmployeeNumber" type="xs:string"></xs:element>
  </xs:sequence>
</xs:complexType>
</xs:schema>
```

Quando você importa o esquema, o Designer cria uma exibição para Employee_Payroll, EmployeeType e EmployeeStatus. A exibição EmployeeStatus contém a coluna chamada Employee. Employee deriva de EmployeeType.

A figura a seguir mostra a exibição Employee_Payroll, a exibição EmployeeType e a exibição XML EmployeeStatus:



A exibição EMPLOYEE_Payroll contém o elemento Employee_Payroll e uma chave primária, PK_Employee_Payroll. A exibição Employee _ Payroll está conectado à exibição EmployeeStatus por uma linha azul que indica uma sequência de relacionamento um a muitos entre as exibições. Employee_Payroll contém várias ocorrências de EmployeeStatus.

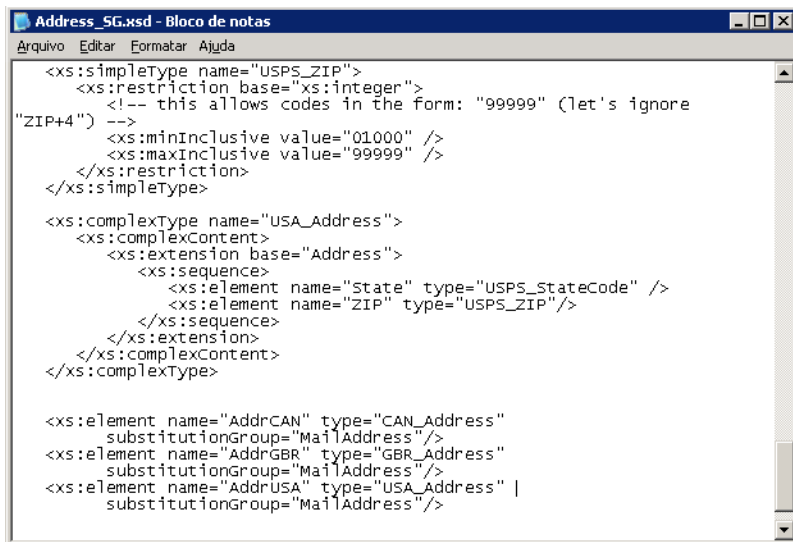
A exibição EmployeeStatus contém um elemento Employee do tipo EmployeeType. O elemento Employee estende EmployeeType, incluindo um elemento EmployeeStatus. A exibição EmployeeStatus também contém uma chave externa para Employee_Payroll. A exibição EmployeeStatus está conectada a uma exibição com uma seta cinza EmployeeType. A seta indica um tipo de relacionamento entre as exibições.

A exibição EmployeeType contém um EmployeeType que consiste em EmployeeName e employeeNumber.

Usando grupos de substituição em uma definição XML

Quando você cria uma definição XML que contém relacionamentos de entidade, o Designer gera exibições separadas para grupos de elementos e tipos complexos. Quando você importa um esquema XML que usa grupos de substituição, o Designer importa cada membro do grupo de substituição como uma entidade separada. O Designer cria uma exibição separada para cada grupo.

A figura a seguir mostra um parte de amostra de um esquema XML que contém o grupo de substituição MailAddress:



A figura a seguir mostra uma definição XML com uma exibição para cada membro do grupo de substituição, incluindo AddrCAN, AddrGBR, AddrUSA, ShortAddress e Street:

Nome	XPath	Tipo de dados
AddrCAN (X_AddrCAN)		
XPk_Addr...		xsd:integer
AddrCAN	AddrCAN	xsd:integer
Province	AddrCAN/Pr...	xsd:string
PostalCode	AddrCAN/P...	CAN_Postal...
AddrGBR (X_AddrGBR)		
XPk_Addr...		xsd:integer
AddrGBR	AddrGBR	xsd:integer
County	AddrGBR/C...	xsd:string
Postcode	AddrGBR/P...	GBR_Postc...
AddrUSA (X_AddrUSA)		
XPk_Addr...		xsd:integer
AddrUSA	AddrUSA	xsd:integer
State	AddrUSA/St...	USPS_State...
ZIP	AddrUSA/ZIP	USPS_ZIP
ShortAddress (X_ShortAddress)		
XPk_Short...		xsd:integer
Name	type(ShortA...	xsd:string
City	type(ShortA...	xsd:string
Street (X_Street2)		
XPk_Str...		xsd:integer
FK_Shor...		xsd:integer
Street	type(ShortA...	xsd:string

Trabalhando com Referências Circulares

Um relacionamento circular é um relacionamento de hierarquia circular entre duas exibições em uma definição XML ou dentro de uma única exibição em uma definição. Por exemplo, um elemento complexo chamado Part pode conter um ID, um nome de peça e uma referência à outra peça.

O seguinte exemplo mostra os componentes do elemento Part:

```

<xs:element name="Part">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ID" type="xs:string"/>

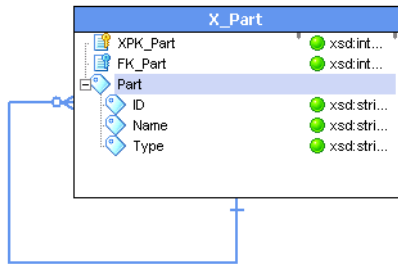
```

```

    <xs:element name="Name" type="xs:string"/>
    <xs:element name="Type" type="xs:string"/>
    <xs:element ref="Part" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
</xs:element>

```

A figura a seguir mostra uma referência circular no Editor de XML no espaço de trabalho com um elemento complexo chamado Part:



Você pode usar a definição XML Part para ler o seguinte arquivo XML em uma sessão:

```

<Part>
  <ID>1</ID>
  <Name>Big Part</Name>
  <Type>L</Type>
  <Part>
    <ID>1.A</ID>
    <Name>Middle Part</Name>
    <Type>M</Type>
    <Part>
      <ID>1.A.B</ID>
      <Name>Small Part</Name>
      <Type>S</Type>
    </Part>
  </Part>
</Part>

```

No arquivo XML, Part 1 contém Part 1.A, e Part 1.A contém Part 1.A.B.

A seguinte figura mostra os dados e as chaves que uma sessão pode gerar a partir da origem XML:

Visualizar Dados XML					
Visualizar dados com arquivo circle reference.xml					
XPK_Part	FK_Part	ID	Name	Type	
1	NULL	1	Big Part	L	
2	1	1.A	Middle Part	M	
3	2	1.A.B	Small Part	S	

Número de Linhas: 3

OK Ajuda

Nota: Você não poderá executar uma sessão que contenha uma referência de XML circular se a sessão estiver ativada para carregamento baseado em restrições. A sessão rejeita todas as linhas.

Noções Básicas de Linhas de Exibição

Para extrair dados de um documento XML, você especifica as linhas a serem geradas, as colunas de dados a serem incluídas e quando gerar as linhas. Ao definir uma exibição no Editor de XML, você cria a linha de exibição, um elemento ou um tipo complexo global que o Serviço de Integração requer para gerar uma linha de dados.

O Serviço de Integração usa uma linha de exibição para determinar quando ler e gravar dados de uma exibição XML. Você pode definir uma linha de exibição em qualquer elemento com uma única ocorrência ou com várias ocorrências. Depois que você definir a linha de exibição, todos os elementos adicionados à exibição terão uma correspondência um para um à linha de exibição.

Por exemplo, a exibição Employees contém os elementos Employee, Name, Firstname e Lastname. Quando você define a linha de exibição como Employee, o Serviço de Integração extrai dados usando o seguinte algoritmo:

```
For every (Employees/Employee)

  extract ./Name/Firstname/Lastname
```

Um esquema XML Employees pode conter os seguintes elementos:

```
EMPLOYEES
  EMPLOYEE+
    ADDRESS+
    NAME
      FIRSTNAME
      LASTNAME
    EMAIL+
```

Employee, Address e Email são elementos com várias ocorrências. Você pode criar uma exibição que contenha os seguintes elementos:

```
EMPLOYEE
  ADDRESS
  NAME
```

Se você definir a linha de exibição como Address, o Serviço de Integração extrairá um Name para cada Employee/Address nos dados XML. Não é possível adicionar Email a essa exibição, pois você cria um relacionamento muitos para muitos entre Address e Email.

Você pode adicionar uma coluna dinâmica com várias ocorrências à exibição. Uma linha de exibição pode conter a coluna dinâmica.

Por exemplo, você pode adicionar uma instância de Email como uma coluna dinâmica à exibição Employee. A exibição conterá os seguintes elementos:

```
EMPLOYEE
  ADDRESS
  NAME
  EMAIL[1]
```

A exibição pode ter a linha de exibição, EMPLOYEE/ADDRESS/EMAIL[1]. O Serviço de Integração extrai os dados para a primeira instância de Employee/Address/Email.

Usando Predicados de Consulta do XPath

Use uma consulta em uma exibição XML para filtrar dados de origem XML. O Serviço de Integração extrai dados de um arquivo XML de origem baseado na consulta. Se a consulta for verdadeira, o Serviço de Integração extrairá dados da exibição.

Para criar consultas em exibições XML, você cria um predicado de consulta do XPath no Editor XML. XPath é uma linguagem que descreve uma maneira de localizar itens em um documento XML. O XPath usa uma sintaxe de endereçamento baseada no caminho de passagem pela hierarquia XML de um componente raiz. Você pode criar um predicado de consulta do XPath para elementos na linha de exibição ou elementos e atributos com um XPath que inclua a linha de exibição.

Um predicado de consulta do XPath inclui um elemento ou atributo a ser extraído e o predicado de consulta que determina os critérios. Você pode verificar o valor de um elemento ou atributo ou pode verificar se um elemento ou atributo existe nos dados XML de origem.

Regras e Diretrizes para o Uso de Linhas de Exibição

Use as seguintes regras e diretrizes para usar linhas de exibição em uma definição XML:

- Uma linha de exibição deve ser um tipo ou um elemento. Uma linha de exibição não pode ser um atributo.
- Cada exibição deve ter uma linha, que deve ser um elemento ou um tipo complexo.
- A raiz da exibição é o elemento de nível superior em uma exibição. A raiz da exibição é o pai para todos os outros elementos na exibição.
- A linha de exibição pode ser igual à raiz da exibição, a menos que a exibição seja desordenada.
- Duas exibições podem ter a mesma linha de exibição em uma origem XML ou em uma transformação de Analisador de XML.
- O elemento de linha de exibição deve ser o elemento mais baixo com várias ocorrências na exibição. Uma exibição não pode conter relacionamentos muitos para muitos.
- Se adicionar um elemento com várias ocorrências a uma exibição sem outro elemento com várias ocorrências, você alterará a linha de exibição no novo elemento por padrão. Se a exibição já tiver um elemento com várias ocorrências, você não poderá adicionar outro elemento com várias ocorrências.
- Você não precisa especificar uma linha de exibição quando cria uma exibição vazia. No entanto, assim que você adicionar uma coluna à exibição, o Designer criará a linha de exibição. Isso será verdadeiro mesmo se você adicionar apenas a chave primária.
- Você poderá alterar uma linha de exibição posteriormente, mas não poderá alterar uma raiz de exibição, a menos que não haja componentes do esquema na exibição.
- Você pode especificar uma linha de exibição que consista em um elemento dinâmico, como:

```
Product/Order[2]/Customer
```

- Em uma exibição, uma linha de exibição efetiva é o caminho das linhas de exibição do início de um relacionamento de hierarquia até o fim da linha de exibição. Uma exibição pode ter várias linhas de exibição efetivas, pois a exibição pode ter vários relacionamentos de hierarquia na definição XML.

Você pode especificar opções no Editor de XML que afetam como as linhas de exibição e as linhas de exibição efetivas afetam a saída de dados.

Colunas dinâmicas

Geralmente, um elemento que ocorre várias vezes é um conjunto dos mesmos elementos que contêm valores diferentes. Por exemplo, um elemento denominado Sales que ocorre 12 vezes pode conter os números de vendas referentes a cada mês do ano. Ou então, um elemento denominado Endereço que ocorre duas vezes pode ser um endereço residencial e um endereço comercial.

Se você tiver esse tipo de elemento em uma origem XML, use a dinamização para tratar ocorrências de elementos como colunas separadas em um grupo. Para dinamizar as ocorrências de um elemento em uma exibição XML, crie uma coluna para cada ocorrência que você deseja representar na definição. No exemplo de vendas mensais, se você quiser representar as 12 ocorrências como colunas, crie 12 colunas de vendas na exibição. Se quiser representar as vendas de um trimestre, crie três colunas. Quando você executa uma sessão, o Serviço de Integração ignora todos os dados XML das ocorrências que não são incluídas na definição.

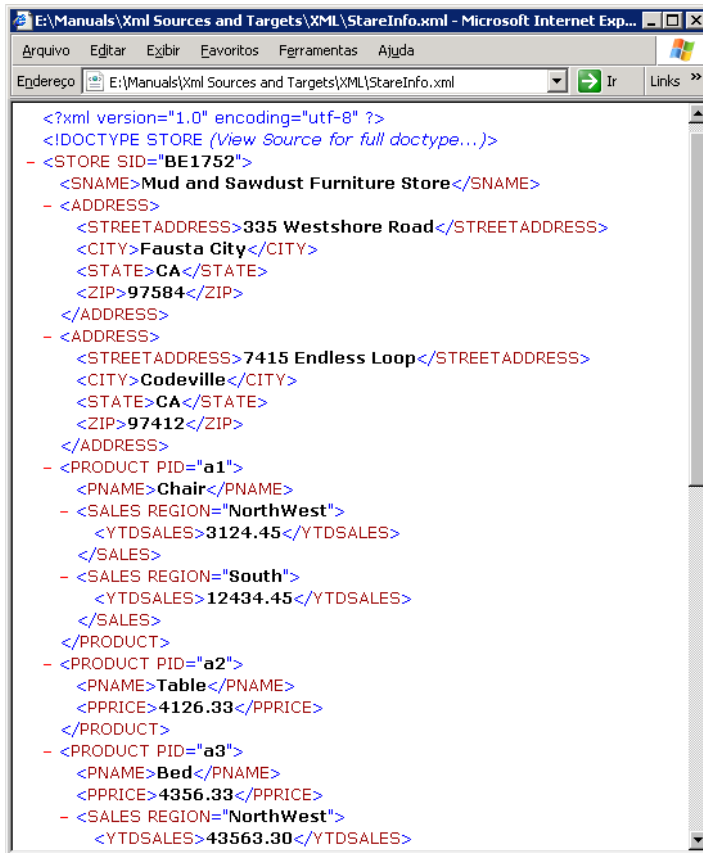
Você pode dinamizar as colunas ao adicionar ou editar uma exibição na definição de origem XML.

Você pode dinamizar os tipos simples e complexos. Não é possível dinamizar uma coluna de chaves primárias. Quando você dinamiza as colunas de uma exibição, a estrutura de grupos resultante deve seguir as regras para uma exibição normalizada ou desordenada válida. O Designer exibirá avisos e erros se a coluna dinâmica invalidar uma exibição.

A dinamização afeta um elemento da exibição no local em que você dinamiza o elemento. Ao dinamizar um elemento da exibição, você não altera o mesmo elemento em outra exibição.

Nota: Você pode dinamizar as colunas em um destino XML.

A figura a seguir mostra duas ocorrências do elemento Endereço no arquivo XML StoreInfo:



A primeira ocorrência de Address torna dinâmicas as colunas de endereços residenciais com o prefixo HOM_. A segunda ocorrência de Address torna dinâmicas as colunas de endereços comerciais com o prefixo OFC_. XPath mostra os dois conjuntos de colunas que se originam nos mesmos elementos.

A figura a seguir mostra o elemento ADDRESS do arquivo XML StoreInfo dinamizado em dois conjuntos de colunas de endereço:

Nome da Coluna PowerCe...	Tipo	Não ...	XPath
SID	xsd:short	<input type="checkbox"/>	./@SID
si_SNAME	xsd:string (30)	<input type="checkbox"/>	./si:SNAME
HOM_STREETADDRESS	xsd:string (30)	<input type="checkbox"/>	./si:ADDRESS[1]/STREETADDRESS
HOM_CITY	xsd:string (30)	<input type="checkbox"/>	./si:ADDRESS[1]/CITY
HOM_STATE	xsd:string (30)	<input type="checkbox"/>	./si:ADDRESS[1]/STATE
HOM_ZIP	xsd:string (30)	<input type="checkbox"/>	./si:ADDRESS[1]/ZIP
OFC_STREETADDRESS	xsd:string (30)	<input type="checkbox"/>	./si:ADDRESS[2]/STREETADDRESS
OFC_CITY	xsd:string (30)	<input type="checkbox"/>	./si:ADDRESS[2]/CITY
OFC_STATE	xsd:string (30)	<input type="checkbox"/>	./si:ADDRESS[2]/STATE
OFC_ZIP	xsd:string (30)	<input type="checkbox"/>	./si:ADDRESS[2]/ZIP

Na figura a seguir, a primeira e segunda ocorrências de Address (com HOM_ e OFC_ prefixos) são exibidos como colunas no grupo:

GPK_ADDRESS	FK_SID	HOM_STREETADDRESS	HOM_CITY	HOM_STATE	HOM_ZIP	OFC_STREETADDRESS	OFC_CITY	OFC_STATE	OFC_ZIP
1	1	335 West Bayshore Road	Fausta city	CA	97584	7415 Endless Loop	Codeville	CA	97412

Usando Dinâmicos de Vários Níveis

Você pode tornar dinâmico mais de um nível de elementos em uma exibição especificando deslocamentos fixos para elementos com várias ocorrências no XPath de uma coluna. Por exemplo, você pode ter os seguintes elementos em uma exibição:

```
STORE
    PRODUCT+
        PNAME
        ORDER+
            ORDERNAME
            CUSTOMER+
                CUSTNAME
```

O XPath `STORE/PRODUCT[2]/ORDER[1]/ORDERNAME` refere-se ao nome da ordem para a primeira ordem do segundo produto da loja. O XPath `STORE/PRODUCT[2]/ORDER/CUSTOMER[1]` refere-se ao primeiro cliente de todas as ordens do segundo produto.

Se você tornar dinâmica uma linha de exibição, qualquer coluna na exibição XML que ocorra abaixo da linha de exibição deverá ter um XPath correspondente ao XPath da linha de exibição.

Por exemplo, uma exibição pode ter a seguinte linha:

```
Transaction/Trade[1]
```

As seguintes colunas têm a mesma ocorrência de Trade no XPath:

```
Transaction/Trade[1]/Date
```

```
Transaction/Trade[1]/Price
```

```
Transaction/Trade[1]/Person[1]/Firstname
```

Não é possível criar uma coluna com o seguinte XPath na exibição:

```
Transaction/Trade[2]/Date
```

Trabalhando com Origens XML

Trabalhando com a Visão Geral de Origens XML

O Designer fornece um Assistente de XML que você pode usar para criar definições XML no repositório. Você pode importar arquivos de uma URL ou um nó local para criar uma definição XML. Também é possível importar definições relacionais ou de arquivo simples de um repositório do PowerCenter. Você pode criar definições XML a partir dos seguintes tipos de arquivos:

- Arquivos XML
- Arquivos de esquema XML
- Arquivos DTD

- Definições relacionais
- Definições de arquivo simples

Ao criar definições XML, você importa arquivos com o Assistente de XML e organiza metadados em exibições XML. As exibições XML são grupos de colunas que contêm os elementos e os atributos no arquivo XML. O assistente pode gerar as exibições ou você pode criar exibições personalizadas.

Você pode criar relacionamentos entre exibições no Assistente de XML. Você pode criar relacionamentos de hierarquia ou de entidade.

Você poderá sincronizar uma definição XML com relação a um arquivo de esquema XML se a estrutura do esquema for alterada.

Importando uma Definição de Origem de XML

Quando você importa uma definição de origem de um esquema XML ou arquivo DTD, o Designer pode fornecer uma definição precisa dos dados com base na descrição fornecida no arquivo de esquema DTD ou XML. Quando você importa uma definição de origem baseada em um arquivo XML sem um esquema XML ou DTD associado, o Assistente de XML determina os tipos e as ocorrências dos dados com base nos dados representados no arquivo XML. Ao criar a definição XML, você pode obter resultados inesperados. Por exemplo, o Designer pode definir um atributo de escala impreciso para colunas de string. Se você exportar a definição de origem XML e importá-la com os atributos de escala imprecisos, ocorrerão erros.

Depois de criar uma definição de origem XML, você não poderá alterar a definição de origem para nenhum outro tipo de origem. Entretanto, você não poderá alterar outros tipos de definição de origem para definições XML.

O Assistente de XML usa chaves para relacionar as exibições XML e reconstruir a hierarquia XML. Você pode optar por gerar exibições e chaves primárias ou pode criar exibições e especificar chaves. Ao criar exibições personalizadas, você pode selecionar raízes e escolher como tratar a expansão de metadados.

O Assistente de XML salva a hierarquia XML e as informações de exibição como um esquema XML no repositório. Quando você importa uma definição XML, a capacidade de alterar a cardinalidade e o tipo de dados dos elementos na hierarquia depende do tipo de arquivo que você está importando. Por exemplo, arquivos DTD e XML não armazenam informações de tipo de dados. Ao importar esses arquivos para criar uma definição XML, você pode configurar o tipo de dados, a precisão e a escala no Designer. Se importar um esquema XML, você poderá alterar a precisão e a escala.

Não é possível criar uma definição de origem XML com base em um arquivo XML de objetos exportados do repositório. Quando você importa uma definição de origem, o Designer aplica uma página de código padrão à definição XML no repositório. A página de código é baseada na página de código do Cliente do PowerCenter. Você não pode alterar a página de código de uma definição de origem XML, mas pode alterar a página de código de uma definição de destino XML depois de criá-la.

Use o Assistente de XML para importar as definições de origem XML.

Para importar um arquivo XML:

1. Clique em Origens > Importar Definição XML.

A caixa de diálogo Importar Definição XML é exibida.

2. Clique em Opções Avançadas.

A caixa de diálogo Alterar Opções de Criação e Nomenclatura das Exibições XML é exibida. Selecione as opções para especificar como o Designer cria e atribui nomes às exibições XML.

A seguinte tabela descreve as opções de exibição XML:

Opção	Descrição
Substituir todos os comprimentos infinitos	Você pode especificar um comprimento padrão para componentes com comprimentos indefinidos, como strings. Se você não definir um comprimento padrão, a precisão desses componentes será definida como infinita. A precisão infinita poderá provocar erros de tamanho de buffer do DTM quando você executar uma sessão com arquivos grandes.
Analisar elementos/atributos no XML autônomo como declarações globais	Escolha esta opção para criar declarações globais de elementos ou atributos XML autônomos. Você pode reutilizar elementos globais fazendo referência a eles em outras partes do esquema. Quando você desmarca essa opção, o XML autônomo é uma declaração local.
Criar uma exibição XML para um elemento com compartimentos	Você poderá criar uma exibição separada de um elemento com compartimentos se o elemento e os elementos filhos puderem ocorrer mais de uma vez. Um elemento com compartimentos é aquele que não tem conteúdo de texto ou atributos, mas tem elementos filhos.
Criar elementos dinâmicos em colunas	Você poderá criar elementos folha dinâmicos se eles tiverem limite de ocorrência. Os elementos dinâmicos poderão ser criados somente em definições de origem.
Ignorar elemento fixo e valores de atributo	Você pode ignorar valores fixos em um esquema e permitir outros valores de elementos nos dados.
Ignorar atributos proibidos	Você pode declarar um atributo como proibido em um esquema XML. Os atributos proibidos restringem tipos complexos. Ao importar o esquema ou o arquivo, você pode optar por ignorar os atributos proibidos.
Gerar nomes para colunas XML	<p>Você pode optar por nomear colunas XML com uma sequência de números ou com o nome do elemento ou do atributo do esquema. Se você usar nomes, escolha entre as seguintes opções:</p> <ul style="list-style-type: none">- Quando a XMLColumn se referir a um atributo, use o nome do elemento como prefixo. O PowerCenter usa o seguinte formato para o nome da coluna XML: <code>NameOfElement_NameOfAttribute</code>- Use o nome da exibição XML como prefixo para cada coluna XML. O PowerCenter usa o seguinte formato para o nome da coluna XML: <code>NameOfView_NameOfElement</code>- Use o nome da exibição XML como prefixo para cada coluna de chave estrangeira. O PowerCenter usa o seguinte formato para o nome da uma coluna de chave estrangeira gerada: <code>FK_NameOfView_NameOfParentView_NameOfPKColumn</code> <p>O comprimento máximo de um nome de coluna é de 80 caracteres. O PowerCenter trunca nomes de coluna com mais de 80 caracteres. Se um nome de coluna não for exclusivo, o PowerCenter adiciona um sufixo numérico para manter o nome exclusivo.</p>

3. Clique em OK para aplicar as alterações.

4. Escolha o tipo de arquivo a ser importado. Escolha uma das seguintes opções:
 - **Importar a definição de um arquivo XML local ou de uma URL.** Crie uma definição de origem a partir de um arquivo XML, DTD ou de esquema XML. Se você importar um arquivo XML com um DTD ou um esquema associado, o Assistente de XML usará o DTD ou o esquema para gerar o documento XML.
 - **Importar a definição de uma origem ou um destino que não é XML.** Use esta opção para criar uma definição de origem a partir de arquivo simples ou definições relacionais. A nova definição de origem contém um grupo para cada definição de entrada mais um grupo de elemento raiz.
5. Clique em Avançar para concluir o Assistente de XML.

Valores de atributos de várias linhas

O Assistente de XML não permite valores de atributos que contenham caracteres de nova linha e que se expandam por mais de uma linha. Quando você importa uma definição de origem ou de destino de um arquivo XML que contém um atributo com caracteres de nova linha, o Assistente de XML exibe um erro e não importa o arquivo.

Trabalhando com exibições XML

O Designer mostra exibições como grupos na definição de origem.

O Assistente de XML fornece opções para a criação de exibições na definição. Você também pode criar as exibições manualmente no Editor de XML.

Você pode escolher uma das seguintes opções para criar exibições XML:

- **Gerar relacionamentos de entidade.** Se você criar relacionamentos de entidade, o Assistente de XML gerará exibições para elementos referenciados ou com várias ocorrências e tipos complexos.
- **Gerar relacionamentos de hierarquia.** Quando você cria relacionamentos hierárquicos, cada referência a um componente se expande no respectivo elemento pai. É possível gerar exibições XML normalizadas ou desordenadas em um relacionamento de hierarquia.
 - **Exibições XML normalizadas.** Quando você gera uma exibição XML normalizada, os elementos e os atributos aparecem uma vez. Elementos com várias ocorrências ou elementos em relacionamentos de um para muitos aparecem em exibições diferentes relacionadas por chaves.
 - **Exibições XML desordenadas.** Quando você gera uma exibição XML desordenada, todos os elementos e atributos aparecem em uma exibição. O Designer não modela relacionamentos de muitos para muitos entre elementos e atributos em uma definição XML.
- **Criar exibições XML personalizadas.** Você pode especificar qualquer elemento global como raiz ao criar uma exibição XML personalizada. Você pode optar por reduzir a explosão de metadados para elementos, tipos complexos e tipos complexos herdados.
- **Sincronizar definições XML.** Você poderá atualizar uma ou mais definições XML quando seus esquemas subjacentes forem alterados.
- **Ignorar criação de exibições XML.** Se optar por ignorar a criação de exibições XML, poderá defini-las mais tarde no Editor de XML. Ao definir exibições no Editor de XML, você pode definir que as exibições correspondam aos destinos e simplifiquem o mapeamento.

- **Criar exibições XML para os elementos e atributos no arquivo XML.** Ao importar um arquivo XML com um esquema associado, você só pode criar exibições XML para os elementos e os atributos do arquivo XML, não pode criá-las para todos os componentes no esquema.

Se você optar por gerar relacionamentos de entidades ou hierarquias, o Designer escolherá uma raiz padrão e criará as exibições XML. Se a definição XML exigir mais de 400 exibições, uma mensagem será exibida informando que a definição é grande demais. Você pode criar exibições manualmente no Editor de XML. Importe a definição XML e opte por criar exibições personalizadas ou ignore a geração de exibições XML.

Quando você importa uma definição de um esquema XML que não tem elementos globais, o Designer não pode criar uma exibição raiz na definição XML. O Designer exibe uma mensagem informando que não há elemento global.

Depois de criar uma exibição XML, você não pode alterar as opções de configuração que definiu para a exibição. Por exemplo, se você criar uma exibição XML normalizada, não poderá alterar a exibição para desordenada. Você deverá importar uma nova definição de origem XML e selecionar a opção desordenada.

Para obter informações sobre dimensionamento de XML no PowerCenter, consulte [“Visão Geral do Uso do XML com o PowerCenter” na página 33](#). Para obter mais informações sobre as limitações que se aplicam ao tratamento de XML no PowerCenter, consulte [“Limitações” na página 33](#).

Para criar uma transformação com outros tipos de elementos e para transformar arquivos de entrada XML maiores, use uma transformação de Processador de Dados. Para obter mais informações sobre como criar transformações de Processador de Dados, consulte o *Guia do Usuário do Informatica Data Transformation* e o *Guia de Introdução do Informatica Data Transformation*.

Importando Parte de um Esquema XML

Quando você importa um esquema XML, o Designer cria uma definição XML que representa o esquema inteiro. Ao importar um arquivo XML com um esquema associado, você só poderá criar exibições XML para os elementos e os atributos do arquivo XML, e não para todos os componentes no esquema. O Designer cria uma definição limitada aos componentes no arquivo XML.

Para importar parte de um esquema, importe um arquivo XML que faz referência ao esquema. Selecione a opção para criar exibições XML somente para os elementos e atributos do arquivo XML.

Regras e Diretrizes para Importar Parte de um Esquema XML

Considere as seguintes regras e diretrizes para importar uma parte do esquema:

- O Designer limita os metadados ao que está no arquivo XML. Se você alterar o arquivo XML e importar o esquema XML novamente, a definição XML será alterada.
- Se um elemento ou atributo ocorrer uma vez na hierarquia de arquivo XML, mas ocorrer em mais de uma parte da hierarquia de esquema, o Designer incluirá todas as ocorrências do elemento ou atributo na definição XML.

Por exemplo, um esquema pode ter dois elementos de endereço, Store/Address e Employee/Address. Se você importar um arquivo XML com o esquema e o arquivo XML tiver apenas Store/Address, o Designer criará todos os elementos Address, incluindo Store/Address e Employee/Address.

- Se o arquivo XML contiver um tipo complexo derivado, o Designer incluirá todos os tipos bases na definição XML como exibições separadas. O Designer não expande o tipo derivado para incluir o tipo base na mesma exibição.
- O Designer não expande referências circulares na definição XML quando o arquivo XML tem vários níveis de referências circulares.

Gerando Relacionamentos de Entidade

Você pode gerar uma hierarquia XML como um modelo de relacionamento de entidade. Quando você gera relacionamentos de entidades de exibição XML, o Designer conclui as seguintes ações:

- Gera exibições para elementos com várias ocorrências ou com referência e tipos complexos.
- Cria relacionamentos entre as exibições.

Quando o Designer gera relacionamentos de entidade, ele gera entidades diferentes para tipos complexos, elementos globais e elementos com várias ocorrências com base nos relacionamentos do esquema.

Se desejar criar grupos diferentes dos grupos padrão ou se desejar combinar elementos de tipos complexos diferentes, você poderá criar exibições XML personalizadas.

Ao exibir uma definição de origem XML no Editor de XML, você pode ver os relacionamentos entre cada elemento na hierarquia XML. Para cada relacionamento entre as exibições, o Editor de XML gera links com base no tipo de relacionamento entre as exibições.

Para gerar relacionamentos de entidade:

1. No Source Analyzer, clique em Origens > Importar Definição XML.
O Assistente de XML é aberto.
2. Navegue até a origem a ser importada e clique em Abrir.
3. Insira um nome para o arquivo e clique em Avançar.
4. Selecione Relacionamentos de Entidade e clique em Concluir.
O Assistente de XML gera uma definição XML que usa relacionamentos de entidade.

Gerando relacionamentos de hierarquia

Quando você cria relacionamentos de hierarquia, cada referência a um componente é expandida sob seu elemento pai. O Assistente de XML seleciona a raiz padrão e usa as configurações padrão para criar grupos XML.

Para gerar relacionamentos de hierarquia:

1. No Source Analyzer, clique em Origens > Importar Definição XML.
O Assistente de XML é aberto.
2. Navegue até a origem a ser importada e clique em Abrir.
3. Insira um nome para o arquivo e clique em Avançar.
4. Selecione Relacionamentos de Hierarquia.

5. Selecione Exibições XML Normalizadas ou Exibições XML Desordenadas e clique em Concluir.
O Assistente de XML gera exibições XML com base em relacionamentos de hierarquia.

Criando exibições XML personalizadas

Você pode criar exibições XML personalizadas usando o Assistente de XML. Ao criar exibições personalizadas, você pode escolher raízes e especificar como gerar metadados. É possível optar por incluir ou excluir elementos globais com base nas informações raízes que são aplicadas ou não aos dados que você pretende processar. Por exemplo, se o esquema contiver informações sobre lojas e clientes, você poderá criar uma definição XML que processe apenas os clientes.

Você pode especificar como deseja gerar os metadados associados à exibição. Você pode reduzir a explosão de metadados para elementos, tipos complexos e tipos complexos herdados gerando relacionamentos de entidades. Se você não reduzir as referências de metadados, o Designer gerará relacionamentos de hierarquia e expandirá todos os elementos filhos sob seus elementos pais.

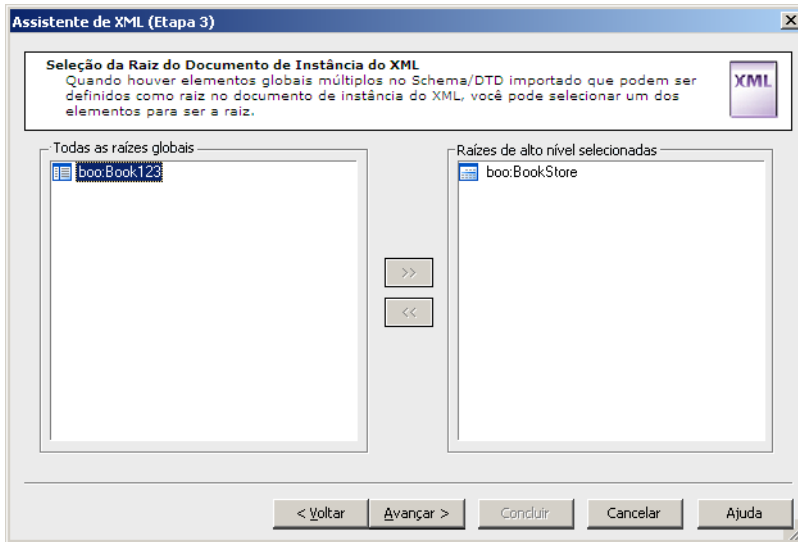
Para criar uma exibição personalizada usando o Assistente de XML:

1. No Source Analyzer, clique em Origens > Importar Definição XML.
O Assistente de XML é aberto.
2. Navegue até a origem a ser importada e clique em Abrir.
3. Insira um nome para o arquivo e clique em Avançar.
4. Selecione Criar Exibições XML Personalizadas e clique em Avançar.
Nota: Para criar manualmente todas as exibições XML no Editor de XML, selecione Ignorar Criação de Exibições XML. O Assistente de XML cria o esquema no repositório, mas não cria as exibições XML.
5. Selecione elementos raiz na lista de elementos raiz globais e clique em Avançar.
6. Opte por reduzir os metadados de elementos, tipos complexos ou tipos complexos herdados e clique em Concluir.

Selecionando Elementos Raiz

Ao criar uma exibição personalizada, você pode escolher entre os elementos globais no esquema importado para definir a raiz do documento de instância XML. Elementos globais são elementos em uma hierarquia de esquema XML que cai diretamente abaixo do elemento raiz superior.

A seguinte figura mostra a página Seleção de Raiz:



Neste exemplo, o elemento Bookstore é selecionado como raiz e o elemento Book123 é eliminado como o elemento raiz.

Reduzindo a Explosão de Metadados

Quando o Designer cria uma definição XML com base em um esquema XML que usa herança, ele pode expandir os metadados de cada elemento ou grupo com referência na exibição que faz referência aos metadados. Ou o Designer pode criar uma exibição separada de um objeto com referência e criar relacionamentos entre o objeto e outras exibições.

Se você usar referências em um esquema XML, convém reduzir o número de vezes que o Designer inclui os metadados associados a uma referência. O Assistente de XML fornece as seguintes opções para reduzir referências de metadados:

- **Reduzir explosão de elementos.** O Designer cria uma exibição de qualquer elemento com várias ocorrências ou qualquer elemento ao qual seja feita referência por mais de um elemento. Cada exibição pode ter vários relacionamentos hierárquicos com outras exibições na definição.
- **Reduzir explosão de tipo complexo.** O Designer cria uma exibição XML para cada tipo complexo com referência ou elemento com várias ocorrências. A exibição XML pode ter vários relacionamentos de tipo com outras exibições. Se o esquema usar tipos complexos herdados, você também poderá reduzir a explosão de tipos complexos herdados.
- **Reduzir explosão de tipo complexo.** Para qualquer tipo herdado, o Assistente de XML cria um relacionamento de tipo.

Quando você reduz a explosão de metadados, o Designer cria relacionamentos de entidade entre as exibições XML que ele gera.

Sincronizando definições XML

Quando você trabalha com definições XML, as origens ou os arquivos usados para criar a definição XML podem ser alterados. Por exemplo, você pode adicionar um novo elemento ou tipo complexo a um

arquivo XSD. Você pode sincronizar uma definição XML com qualquer das seguintes definições de repositório ou arquivos usados para criar a definição XML:

- Definições de origem relacionais
- Definições de destinos relacionais
- Arquivos simples
- URLs
- Arquivos XML
- DTDs
- Arquivos de esquema

Quando você sincroniza uma definição XML, o Designer atualiza o esquema XML no Navegador de Esquema, mas ele não atualiza as exibições na definição XML. Você pode atualizar manualmente as colunas de exibição no Editor de XML depois de sincronizar a definição XML.

Sugestão: Use arquivos de esquema para sincronizar definições XML.

Para sincronizar definições de origem XML:

1. No Source Analyzer, clique em Origens > Importar Definição XML.
O Assistente de XML é aberto.
2. Navegue até a definição do repositório de arquivo usado para criar a definição XML e clique em Abrir.
3. Na Etapa 1 do assistente, clique em Avançar. O assistente ignora qualquer alteração que você tenha feito no nome.
4. Na Etapa 2 do Assistente de XML, opte por sincronizar a definição XML e clique em Avançar.
O Assistente de XML pula para a Etapa 5.
5. Na Etapa 5 do Assistente de XML, escolha a definição XML que você deseja sincronizar.
O Assistente de XML sincroniza a origem com a definição selecionada.

Use esse método para sincronizar definições de destino XML. Se você modificar uma definição de origem XML, talvez também seja necessário sincronizar a definição de destino.

Nota: Verifique se você sincronizou a definição XML com a origem que usou para criar a definição. Se você sincronizar uma definição XML com uma origem que não usou para criar a definição, o Designer não poderá sincronizar as definições e perderá metadados. Clique em Editar > Reverter para Salvo para restaurar a definição XML.

Editando Propriedades de Definição de Origem XML

Depois de importar uma definição de origem XML, você pode editar as propriedades da definição de origem, como o nome da definição. Se configurar a sessão para ler uma lista de arquivos, você poderá configurar o mapeamento para gravar o nome do arquivo de origem em cada linha de destino. Também é possível adicionar extensões de metadados.

Para editar propriedades de definição da origem XML:

1. Clique com o botão direito na parte superior da definição no espaço de trabalho do Source Analyzer. Selecione Editar.
2. Na guia Tabela, edite as seguintes configurações:

Configurações da Tabela	Descrição
Selecionar Tabela	Exibe a definição de origem que você está editando.
Nome Comercial	Nome descritivo da definição de origem. Você pode editar o Nome Comercial clicando no botão Renomear.
Nome do Proprietário	Não aplicável para arquivos XML.
Descrição	Descrição da origem. O limite de caractere é 2.000 bytes/K, em que K é o número máximo de bytes para cada caractere na página de código do repositório. Insira links para a documentação comercial.
Tipo de Banco de Dados	Tipo de origem ou de banco de dados.
Página de Código	Somente Leitura. Não aplicável para arquivos de origem XML. O Serviço de Integração converte todos os arquivos de origem XML em Unicode.

3. Clique na guia Colunas.

Na guia Colunas, é possível exibir informações sobre as colunas da definição. Para alterar nomes ou valores de coluna, use o Editor de XML.

Você pode exibir as seguintes informações:

Configurações de Colunas	Descrição
Selecionar Tabela	A definição de origem que você está editando.
Nome da Coluna	Nome da coluna.
Tipo de dados	Tipos de dados do PowerCenter.
Precisão	Comprimento da coluna.
Escala	Número de posições decimais em dados numéricos.
Não Nulo	Indica se a coluna pode aceitar nulos.
Tipo de Chave	Chave primária, chave externa ou não é uma chave.

Configurações de Colunas	Descrição
XPath	Caminho do elemento ao qual a coluna atual faz referência na hierarquia XML. O XPath não é exibido para chaves primárias ou externas geradas.
Nome Comercial	Nome descritivo definido pelo usuário para a coluna. Se Nome Comercial não estiver visível na janela, role para a direita a fim de exibir ou modificar a coluna.

- Se você configurar a sessão para ler uma lista de arquivos e desejar gravar o nome do arquivo de origem em cada linha de destino, clique na guia Propriedades e selecione Adicionar Porta de Nome de Arquivo Simples Processado Atualmente.

O Designer adiciona a porta `CurrentlyProcessedFileName` à guia Colunas. É a última coluna no primeiro grupo. O Serviço de Integração usa essa porta para retornar o nome do arquivo de origem. A porta `CurrentlyProcessedFileName` é uma porta de string com precisão padrão de 256 caracteres.

Para remover a porta `CurrentlyProcessedFileName`, clique na guia Propriedades e desmarque Adicionar Porta de Nome de Arquivo Simples Processado Atualmente.

- Clique na guia Extensões de Metadados para criar, editar e excluir extensões de metadados definidas pelo usuário.
- Clique em OK.
- Clique em Repositório > Salvar para salvar as alterações no repositório.

Criando definições XML a partir de definições de repositório

Você pode importar uma definição de origem ou de destino XML a partir de definições de arquivos simples ou relacionais no repositório. Quando você importa uma definição XML de uma definição de repositório, o Assistente de XML cria uma hierarquia XML com base nos relacionamentos entre os objetos selecionados. O Assistente de XML cria um elemento raiz para a hierarquia. Você poderá escolher uma raiz nos grupos que criar. Ou poderá criar uma raiz separada e relacionar os grupos a ela.

Quando você criar uma definição de destino XML, o Assistente de XML gerará as chaves para relacionar cada grupo à raiz.

Para criar uma definição XML a partir de origens ou destinos do repositório:

- No Source Analyzer, clique em Origens > Importar Definição XML.
- ou -
No Warehouse Designer, clique em Destinos > Importar Definição XML.
- Na caixa de diálogo Importação de XML, clique em Origens Não XML e Destinos Não XML.
- Selecione uma definição na lista de origens ou de destinos. Clique no botão de Seta para adicionar a definição à lista de origens selecionadas.

Você pode selecionar mais de uma entrada e mais de um tipo de entrada. Se as definições forem relacionadas pelas chaves primárias e externas, o Assistente de XML usará as chaves para relacionar grupos quando ele gerar a hierarquia.
- Para criar um grupo separado para o elemento raiz, insira o Nome Raiz Opcional do XML.

O nome raiz usado como padrão é XRoot. O grupo raiz inclui todos os outros grupos. Remova o nome raiz se desejar usar um dos outros grupos como o raiz. O Assistente de XML cria um grupo para cada definição de origem ou de destino de entrada que você selecionar e gera uma chave primária em cada grupo. O Assistente de XML cria uma chave externa em cada grupo. A chave externa aponta para a chave de link do grupo raiz.

5. Clique em Abrir.

O Assistente de XML é exibido.

6. Use o Assistente de XML para gerar os grupos de origem ou de destino.

Solucionando Problemas de Origens XML

Como posso colocar dois elementos com várias ocorrências se ambos têm o mesmo elemento pai em uma exibição? Por exemplo, preciso colocar todos os elementos de EMPLOYEE em uma exibição:

```
<!ELEMENT EMPLOYEE (EID, EMAIL+, PHONE+)>
```

EMAIL e PHONE pertencem ao mesmo elemento pai, mas não pertencem à mesma cadeia pai. Não é possível colocá-los na mesma exibição desordenada. Para colocar todos os elementos de funcionário em uma exibição, você pode criar dinamicamente um dos elementos com várias ocorrências.

Siga estas etapas para adicionar dois elementos com várias ocorrências na mesma exibição:

1. Crie uma exibição EMPLOYEE.
2. Adicione os elementos EID e EMAIL à exibição EMPLOYEE.
3. Crie dinamicamente o número de ocorrências de EMAIL que você deseja incluir na exibição. Cada ocorrência de EMAIL torna-se um elemento de uma única ocorrência na exibição.
4. Adicione o elemento PHONE.

Tenho a seguinte definição de elemento em meu DTD:

```
<!ELEMENT EMPLOYEE (EMPNO, SALARY)+>
```

Como posso corresponder EMPNO e SALARY na mesma exibição?

O exemplo de DTD é ambíguo. A definição é equivalente a seguinte:

```
<!ELEMENT EMPLOYEE (EMPNO+, SALARY+)>
```

No exemplo de DTD, EMPLOYEE tem os elementos com várias ocorrências EMPNO e SALARY. Não é possível ter dois elementos com várias ocorrências na mesma exibição.

Use uma das seguintes soluções:

- **Reescreva a definição do elemento para torná-la clara.**

Você pode definir o elemento EMPLOYEE da seguinte forma:

```
<!ELEMENT EMPLOYEES (EMPLOYEE+)>
<!ELEMENT EMPLOYEE (EMPNO, SALARY)>
```

Ao usar esta sintaxe, você define um EMPNO e um SALARY para cada EMPLOYEE. A exibição EMPLOYEE contém dois elementos. Inclua EMPLOYEE como um elemento com várias ocorrências em EMPLOYEES.

- **Deixe os elementos em exibições separadas e use a definição de origem duas vezes em um mapeamento.**

Quando EMPNO e SALARY estiverem em exibições diferentes, você ainda poderá combinar os dados em um mapeamento. Use duas instâncias da mesma definição de origem e use uma transformação de Joiner.

Importei um arquivo XML com a seguinte estrutura:

```
<Bookstore>
    <Book>Book Name</Book>
    <Book>Book Name</Book>
    <ISBN>051022906630</ISBN>
</Bookstore>
```

Quando importo este arquivo XML, o Designer elimina o elemento ISBN. Por que isso ocorre? Como posso fazer com que o Designer inclua o elemento ISBN?

- **Use o esquema para importar a definição XML.** Quando você usa um arquivo XML para importar uma definição XML, o Designer lê o primeiro elemento como conteúdo simples, pois o elemento não tem elementos filhos. O Designer descarta o elemento filho ISBN da segunda instância de Book. Se você usar um esquema para importar a definição, o Designer usará a definição do esquema para determinar como ler os dados XML.
- **Verifique se o arquivo XML representa com precisão o esquema associado.** Se você usar um arquivo XML para importar uma definição de origem, verifique se o arquivo XML é uma representação precisa da estrutura no esquema XML correspondente.

Para obter informações sobre dimensionamento de XML no PowerCenter, consulte [“Visão Geral do Uso do XML com o PowerCenter” na página 33](#). Para obter informações sobre as limitações que se aplicam ao tratamento de XML no PowerCenter, consulte [“Limitações” na página 33](#).

Para criar uma transformação com outros tipos de elementos e para transformar arquivos de entrada XML maiores, use uma transformação de Processador de Dados. Para obter mais informações sobre como criar transformações de Processador de Dados, consulte o *Guia do Usuário do Informatica Data Transformation* e o *Guia de Introdução do Informatica Data Transformation*.

Usando o Editor de XML

Usando a visão geral do Editor de XML

Ao importar uma definição XML no Designer, você cria uma definição XML com exibições padrão, exibições personalizadas ou nenhuma exibição. Depois de criar uma definição XML, você usa o Editor de XML para alterá-la.

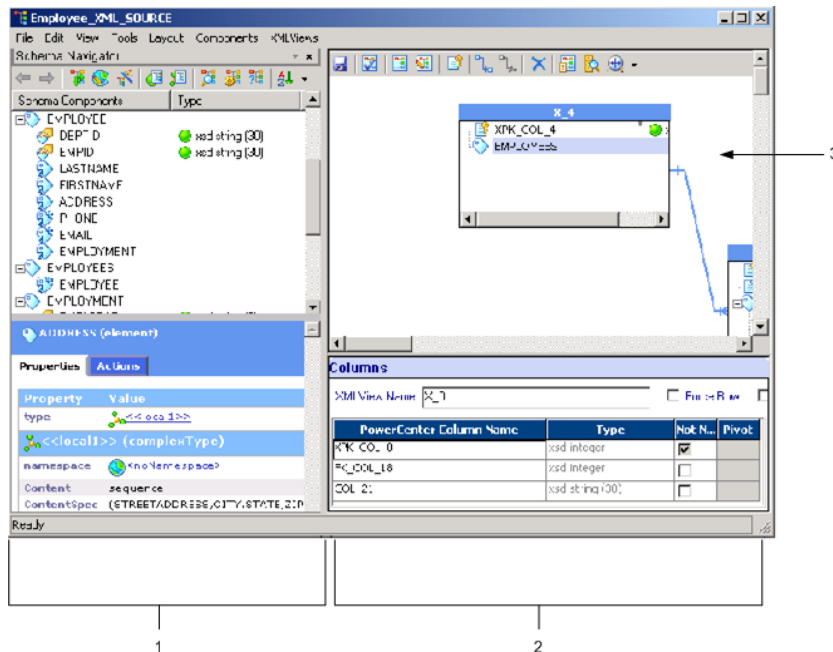
Use o Editor de XML para criar exibições, modificar componentes, adicionar colunas e manter relacionamentos de exibição no espaço de trabalho. Quando você atualiza uma definição XML, o Designer propaga as alterações em qualquer mapeamento que inclui a origem. Algumas alterações nas definições XML podem invalidar mapeamentos.

Nota: Se fizer alterações significativas na origem usada para criar uma definição XML, você poderá sincronizar a definição com a nova origem, em vez de editar a definição manualmente.

O Editor de XML contém as seguintes janelas:

- Navegador
- Espaço de Trabalho do XML
- Janela Colunas

A seguinte figura mostra o Editor de XML:



1. Navegador
2. Janela Colunas
3. Espaço de Trabalho do XML

O Editor de XML usa ícones para representar tipos de componentes XML. Para exibir uma legenda que descreva os ícones no Editor de XML, clique em Exibir > Legenda.

Navegador XML

O Navegador exibe o esquema em um formato hierárquico e fornece informações sobre os componentes selecionados. Você pode classificar componentes no Navegador por tipo, hierarquia ou namespace. Você também pode expandir um componente para ver os componentes abaixo dele na hierarquia.

A barra de ferramentas do Navegador fornece atalhos para a maioria das funções do Navegador. A barra de ferramentas também fornece setas de navegação em que você clica para localizar de forma rápida os componentes exibidos anteriormente na hierarquia.

Estas são as guias do Navegador:

- **Guia Propriedades.** Exibe informações sobre um componente selecionado, como o tipo, o comprimento e a ocorrência do componente.
- **Guia Ações.** Fornece uma lista de opções para exibir mais informações sobre o componente selecionado.

Guia Propriedades

A guia Propriedades exibe informações sobre um componente selecionado no Navegador. Se o componente for um elemento complexo, você poderá exibir as propriedades do elemento no esquema, como namespace, tipo e conteúdo. Quando você exibe um elemento ou atributo simples, a guia Propriedades mostra o tipo e o comprimento do elemento. A guia Propriedades também exibe anotações.

Se importar a definição de um arquivo XML, você poderá editar o tipo de dados e a cardinalidade na guia Propriedades. Se criar a definição a partir de um arquivo DTD, você poderá editar o tipo de componente.

Se um esquema usar um namespace, você poderá alterar o namespace, o prefixo e a localização do esquema do namespace. O prefixo identifica os elementos ou os atributos que pertencem a um namespace. Os elementos e os atributos em um namespace padrão não têm prefixo. Você pode selecionar um namespace como o padrão de um destino XML.

Guia Ações

A guia Ações lista opções que você usa para ver mais informações sobre um componente selecionado. Você também pode reverter alterações feitas em componentes da guia Ações.

As seguintes opções são exibidas na guia Ações, dependendo das propriedades do componente selecionadas:

- **Referências ComplexType.** Exibe os componentes do esquema que são deste tipo.
- **Hierarquia ComplexType.** Exibe os tipos complexos derivados do componente selecionado.
- **Referência SimpleType.** Exibe todos os componentes que são deste tipo.
- **Propagar valores SimpleType.** Propaga os valores de comprimento e escala para todos os componentes deste SimpleType.
- **Referências de elementos.** Exibe os componentes que fazem referência ao elemento selecionado.
- **Componentes filhos.** Exibe os componentes do esquema global que o componente selecionado usa.
- **Reverter simpleType.** Alterará os valores de tipo, comprimento e precisão para o valor original se você os tiver alterado.
- **Referências de exibição XML.** Mostra todas as exibições XML e colunas que fazem referência ao componente selecionado.

Espaço de Trabalho do XML

O espaço de trabalho XML mostra as exibições XML e os relacionamentos entre as exibições. Você pode criar exibições XML no espaço de trabalho e definir relacionamentos entre exibições.

A barra de ferramentas do espaço de trabalho XML fornece atalhos para a maioria das funções que você pode utilizar no espaço de trabalho.

Você pode modificar o tamanho do espaço de trabalho XML das seguintes maneiras:

- **Oculte a janela Colunas.** Clique em Exibir > Coluna Propriedades.
- **Oculte o Navegador.** Clique em Exibir > Navegador.
- **Reduza o espaço de trabalho.** Clique no botão Zoom na barra de tarefas Espaço de Trabalho.

Janela Colunas

A janela Colunas exibe as colunas de uma exibição no espaço de trabalho. Use a janela Colunas para denominar as colunas adicionadas. Se usar colunas dinâmicas, você usará a janela Colunas para selecionar e renomear ocorrências de elementos com várias ocorrências. Você também pode especificar opções, como Não Nulo, Forçar Linha, Linha de Relacionamento de Tipo ou de Hierarquia e Linha Não Recursiva. Essas opções afetam como o Serviço de Integração grava dados em destinos XML.

Criando e editando exibições

Use o Editor de XML para criar exibições XML personalizadas ou criadas com o Assistente de XML. Para criar uma exibição, você a define e especifica as colunas na exibição. Se o esquema tiver elementos com várias ocorrências, você poderá especificar quais ocorrências do elemento incluir na exibição. Também é possível criar portas especiais para os nomes de arquivo de destino XML e portas de passagem para transformações de Analisador de XML e Gerador de XML.

Conclua as seguintes tarefas para criar e editar exibições XML:

- **Criar uma exibição XML.** Adicione uma exibição ao espaço de trabalho.
- **Adicionar colunas a uma exibição.** Crie novas colunas em uma exibição.
- **Exclua as colunas de uma exibição.** Exclua as colunas de uma exibição.
- **Expandir um tipo complexo.** Escolha um tipo complexo derivado para adicionar a uma exibição.
- **Importar um elemento anyType.** Importe elementos anyType.
- **Aplicar conteúdo a um elemento anyAttribute.** Defina conteúdo para elementos anyAttribute.
- **Usar um elemento anySimpleType.** Use um elemento anySimpleType em uma definição XML.
- **Adicione uma porta de passagem.** Adicione uma porta para transmitir dados que não são XML por meio de uma transformação XML.
- **Adicione uma coluna FileName.** Adicione uma coluna para gerar um novo nome de arquivo para cada arquivo de destino XML.

Criando uma exibição XML

Você pode criar exibições no espaço de trabalho XML. Quando você cria uma definição XML sem exibições, o Editor de XML exibe um espaço de trabalho vazio. É possível criar uma exibição e adicionar a ela as colunas e a linha de exibição.

Para criar uma nova exibição XML no espaço de trabalho:

1. Abra a definição XML no Editor de XML.
2. Clique em Exibições XML > Criar Exibição XML.
O Editor de XML cria uma exibição em branco no espaço de trabalho e exibe colunas vazias na janela Colunas.
3. Insira um nome para a exibição na janela Colunas.
O nome aparece na exibição XML no espaço de trabalho.
4. Destaque o nó no Navegador de Esquema a partir do qual deseja criar uma exibição.

5. Clique em Componentes > Navegador do XPath.
O Navegador do XPath é exibido na janela Navegador.
6. Defina o Modo de Coluna no Navegador do XPath como Modo de Linha de Exibição para adicionar a linha de exibição.
7. Selecione o elemento no Navegador e arraste o elemento para a exibição no espaço de trabalho.
O Editor de XML destaca a linha de exibição em azul.
Na primeira vez que você adiciona uma coluna à exibição, o Designer verifica se a coluna pode ser uma linha de exibição. Isso ocorrerá mesmo se você não especificar para adicionar uma linha de exibição.
8. Para alterar a linha de exibição para outra coluna, clique com o botão direito do mouse na linha apropriada e escolha Definir Como Linha de Exibição.

Adicionando colunas a uma exibição

Você pode adicionar colunas a uma exibição XML dentro do espaço de trabalho de XML. Para adicionar uma coluna, selecione-a no Navegador do XPath.

Você poderá adicionar uma coluna à exibição XML quando as seguintes condições forem verdadeiras:

- O caminho do componente começar no elemento do esquema que é a raiz da exibição.
- O componente não for um elemento com compartimentos.
- O componente não violar as regras de normalização ou criação dinâmica. Por exemplo, você não pode adicionar mais de um elemento com várias ocorrências a uma exibição.
- Você pode adicionar elementos de conteúdo misturados como tipos simples ou complexos.
- Duas exibições podem compartilhar a mesma coluna e uma exibição pode conter várias colunas idênticas.

Para adicionar colunas a uma exibição:

1. Selecione uma exibição XML no espaço de trabalho.
2. Destaque um elemento pai no Navegador que contém os componentes a serem adicionados.
3. Clique em Componentes > Navegador do XPath.
O Navegador do XPath é exibido na janela Navegador.
4. Clique no botão Modo e opte por adicionar uma coluna ou uma linha de exibição.
5. Escolha Avançado para adicionar uma coluna dinâmica à exibição.
Uma coluna dinâmica é uma ocorrência de um elemento com várias ocorrências. Você pode adicionar colunas com uma única ocorrência em Modo Avançado.
Nota: Não é possível criar colunas dinâmicas em definições de destino XML.
6. Arraste uma coluna do Navegador do XPath para a exibição apropriada no espaço de trabalho de XML. Você pode selecionar várias colunas simultaneamente.
O Editor de XML valida a coluna adicionada. Se a coluna for inválida para a exibição, uma mensagem será exibida na barra de status ao arrastar a coluna. Novas colunas são exibidas conforme você as adiciona às exibições.

Adicionando uma Coluna Dinâmica

Coluna dinâmica em uma definição XML é um elemento com várias ocorrências que forma colunas separadas para as ocorrências do elemento em uma exibição. Você pode criar colunas dinâmicas a partir de qualquer elemento com várias ocorrências em uma definição XML. Você também pode criar elementos dinâmicos em uma linha de exibição.

Se você adicionar uma coluna dinâmica a uma exibição, o número da ocorrência padrão será exibido na janela Colunas. Esse número indica a ocorrência do elemento que deve ser usada na coluna. É possível alterar o número da ocorrência ou adicionar mais ocorrências do elemento como novas colunas. Se você não renomear as colunas, o Editor de XML adicionará um número de sequência a cada nome de coluna dinâmica.

Nota: Você não poderá alterar um valor dinâmico se ele fizer parte de uma linha de exibição.

Para adicionar uma coluna dinâmica à exibição:

1. Selecione a exibição XML no espaço de trabalho.
2. Realce o elemento no Navegador que deseja tornar dinâmico.
Ou destaque qualquer elemento na cadeia pai do elemento que deseja tornar dinâmico.
3. Clique em Componentes > XPath Navigator.
4. Selecionar Modo Avançado.
5. Arraste a coluna a ser tornada dinâmica do XPath Navigator para a exibição no espaço de trabalho do XML.

O Designer adiciona um número de ocorrência padrão na janela Colunas. Esse número indica a ocorrência do elemento que deve ser usada na coluna.

Na figura a seguir, a exibição contém duas ocorrências de Vendas para a terceira ocorrência de Produto:

Columns			
Nome de exibição em XML		Product_Sales_By_Store	Opções de Exibição em XML ▼
Exibir XPath da Linha		STORE/SNAME	
Nome da Coluna PowerCenter	Tipo	Não Nulo	XPath
FIRST_HALF_SALES	xsd:string (30)	<input type="checkbox"/>	../PRODUCT[3]/SALES[1]/@REGION
SECOND_HALF_SALES	xsd:string (30)	<input type="checkbox"/>	../PRODUCT[3]/SALES[2]/@REGION

A primeira ocorrência de Vendas está na coluna chamada First_Half_Sales. A segunda ocorrência de Vendas é Second_Half_Sales. Região é um atributo.

6. Clique no link do XPath para alterar o número de ocorrência de um elemento.
A janela Especificar Predicado de Consulta para XPath é exibida.
7. Selecione o elemento com várias ocorrências para edição.
8. Altere o número da ocorrência na caixa Editar Dinâmico e clique em OK.

Excluindo colunas de uma exibição

É possível excluir colunas em uma exibição.

Para excluir uma coluna da exibição:

1. Clique com o botão direito na coluna da exibição no espaço de trabalho.

2. Selecione Excluir Esta Coluna.

O Editor de XML solicita a sua confirmação para excluir a coluna.

3. Clique em Sim para confirmar.

O Editor de XML remove a coluna da exibição. No entanto, a coluna permanece na hierarquia do Navegador do XPath.

Excluindo uma Coluna Dinâmica

Para excluir uma ocorrência de uma coluna dinâmica, selecione e exclua a coluna da janela Colunas.

Para excluir uma coluna dinâmica:

1. Clique com o botão direito do mouse na coluna que deseja excluir da janela Colunas.
2. Clique em Excluir > Dinâmica.
3. Clique em Sim para confirmar a exclusão.

Expandindo um tipo complexo

Um esquema pode definir um tipo complexo que é um tipo base para mais de um tipo. Por exemplo, uma Publicação pode ser uma Revista ou um Jornal. Ao criar a exibição XML, você pode optar por usar Publicação como um tipo Revista ou Jornal.

Ao exibir um tipo complexo no Navegador do XPath, você pode exibir os tipos derivados.

Para expandir um tipo complexo:

1. Destaque o tipo complexo no Navegador do XPath.
A lista Expandir Tipos Complexos mostra os tipos derivados.
2. Selecione o tipo que deseja usar.
Se você adicionar o componente a uma definição XML, ela conterá o tipo selecionado.

Importando Elementos anyType

Você pode importar um esquema XML que contém um elemento anyType. Um elemento do tipo anyType pode conter qualquer tipo de dados que ocorra em um documento XML. Por exemplo, a seguinte seção de um esquema XML inclui um elemento Document que é anyType:

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="Publication" type="xsd:string"/>
  <xsd:element name="Date" type="xsd:string"/>
  <xsd:element name="Document" type="xsd:anyType"/>
</xsd:schema>
```

Quando você importa um esquema com um elemento anyType, este último é exibido no Schema Navigator como anyType. O Designer não cria uma porta para um elemento do tipo anyType.

Você deve alterar o elemento anyType para um tipo complexo global no Designer para usar o elemento anyType no PowerCenter.

Para alterar um elemento anyType para outro tipo complexo global:

1. Realce o elemento anyType no Schema Navigator.
A propriedade anyType é exibida na guia Propriedades do Navegador do Esquema.
2. Clique na propriedade anyType.
Se o esquema não contiver um tipo complexo global, o Designer exibirá um erro do qual não é possível escolher um tipo complexo global.
3. Selecione um tipo complexo e clique em OK.

Aplicando conteúdo aos elementos anyAttribute ou ANY

Você pode importar um esquema XML que contenha o anyAttribute ou o elemento de conteúdo ANY. Para usar o elemento em uma exibição, você deve definir o conteúdo do elemento no Editor de XML. Quando você importa esquema com um elemento de conteúdo anyAttribute ou ANY, o elemento é exibido no Navegador de Esquema sem propriedades. O elemento não aparece em uma exibição.

Por exemplo, o seguinte elemento de esquema inclui o conteúdo ANY:

```
<xs:element name="person">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="firstname" type="xs:string"/>
      <xs:element name="lastname" type="xs:string"/>
      <xs:any minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Para aplicar conteúdo a anyAttribute ou ANY:

1. Clique no link de conteúdo do elemento no Navegador de Esquema.
A caixa de diálogo Editar Conteúdo do Tipo Any ou anyAttribute é exibida.
2. Clique em Adicionar Tipo.
Uma nova linha aparece.
3. Escolha um elemento na lista de elementos válidos no esquema para a definição XML.
4. Selecione uma cardinalidade e clique em OK.
O tipo substituído aparece no Navegador de Esquema.

Usando anySimpleType no Editor de XML

Um elemento anySimpleType pode conter qualquer tipo de dados atômico, como strings, inteiro, decimal ou data. Use anySimpleType em um esquema quando não souber o tipo de dados do elemento em um documento de instância XML.

Quando você define um elemento como anySimpleType, o Designer cria uma coluna anySimpleType para o elemento quando você importa o esquema. Quando você usa a coluna em um mapeamento, o Qualificador de Origem XML mapeia esse tipo para uma strings.

Adicionando uma porta de passagem

Você pode adicionar uma porta de passagem a uma transformação Analisador de XML ou Gerador de XML para passar dados não XML pela transformação. Quando define a porta na transformação, você adiciona a porta ao grupo DataInput na transformação Analisador de XML ou ao grupo DataOutput na transformação Gerador de XML.

Depois de gerar a porta de passagem, você adiciona outra porta para passar os dados pela transformação. Essa porta é a de referência. Em uma transformação Analisador de XML, a porta de passagem passa os dados para a transformação e a porta de referência passa os dados para fora da transformação. Em um Gerador de XML, a porta de passagem passa os dados para fora da transformação e a porta de referência passa os dados para a transformação.

Se possuir portas de passagem em uma definição XML, você poderá determinar as portas de referência correspondentes.

Para determinar a porta de referência para uma porta de passagem:

1. Clique com o botão direito na porta de passagem.
2. Selecione Navegar até > Coluna Referenciada.

O Editor de XML realça a coluna referenciada no espaço de trabalho.

Adicionando uma Coluna FileName

Quando você executa uma sessão, o Serviço de Integração gera um novo arquivo XML de destino sempre que um novo valor raiz ocorre nos dados. Você pode adicionar uma coluna FileName a uma exibição XML para gerar um nome de arquivo exclusivo para cada arquivo XML. O nome do arquivo substitui o nome do arquivo de saída padrão nas propriedades da sessão.

Ao usar a coluna FileName, você configura uma transformação de Expressão ou outra transformação no mapeamento para gerar os nomes de arquivos exclusivos a serem transmitidos para a coluna FileName.

Para criar uma coluna FileName em uma exibição XML:

1. Clique com o botão direito do mouse na exibição no Editor de XML.
2. Selecione Criar Coluna FileName.

O Editor de XML cria uma nova coluna de strings na exibição.

3. Altere o nome da coluna na janela Colunas.
4. Saia do Editor de XML.

A coluna é exibida na definição XML. O XPath é \$Filename.

Nota: Ao usar portas FileName de XML, você precisa especificar o nome do diretório do arquivo.

Criando um Predicado de Consulta do XPath

Você pode filtrar dados XML em uma sessão criando um predicado de consulta do XPath da linha de exibição ou na coluna abaixo da linha de exibição de uma hierarquia XML. Quando você cria um predicado de consulta do XPath, o Editor de XML adiciona o predicado ao XPath da linha de exibição.

Quando você cria um predicado de consulta do XPath no Editor de XML, o Editor de XML fornece elementos, atributos, operadores e funções para criar a consulta. Você pode selecionar os componentes, inserir componentes ou copiar componentes em uma consulta. O Editor de XML valida cada consulta criada.

Você pode consultar o valor de um elemento ou atributo ou pode verificar se um elemento ou atributo existe.

Consultando o valor de um elemento de atributo

Você pode criar um predicado de consulta do XPath para filtrar valores de elemento ou de atributo em uma exibição. Por exemplo, para extrair funcionários no departamento 100, crie o seguinte predicado de consulta do XPath:

```
EMPLOYEE [./DEPT = '100']
```

A expressão da consulta está entre colchetes. O XPath de Dept é abreviado por “./” para indicar que o caminho é uma continuação de Employee.

O seguinte predicado de consulta do XPath extrairá funcionários se o sobrenome for Smith:

```
EMPLOYEE [./NAME/LASTNAME='SMITH']
```

Name é um elemento filho de Employee e é o pai de Lastname.

Use operadores Booleanos ou numéricos em um predicado de consulta do XPath. Também é possível usar as funções strings, numérico e booleano em uma consulta.

Consultando conteúdo misto

Um elemento XML tem conteúdo misto quando contém um valor e elementos filhos. Você pode criar um predicado de consulta do XPath para filtrar valores de elementos divididos por elementos filhos. No entanto, o Serviço de Integração não avalia predicados que ocorrem depois do primeiro elemento filho no conteúdo misto.

Por exemplo, um arquivo XML pode conter um elemento NAME com conteúdo misto:

```
<NAME>
  Kathy
  <MIDDLE> Mary </MIDDLE>
  Russell
</NAME>
```

O elemento NAME tem o valor “Kathy”, um conteúdo filho “MIDDLE” e um segundo valor “Russell”. O valor da coluna NAME é “KathyRussell”. No entanto, o Serviço de Integração avalia o NAME “Kathy”.

A seguinte expressão é falsa:

```
EMPLOYEE [./NAME = 'KathyRussell']
```

A seguinte consulta é verdadeira:

```
EMPLOYEE [./NAME = 'Kathy']
```

Operadores booleanos

Use os seguintes operadores Booleanos em um predicado de consulta do XPath:

```
and or < <= > >= !=
```

Use o seguinte predicado de consulta do XPath para extrair funcionários do departamento 100 com o sobrenome Jones:

```
EMPLOYEE [./DEPT = '100' and ./ENAME/LASTNAME = 'JONES']
```

Operadores numéricos

Use os seguintes operadores numéricos em um predicado de consulta do XPath:

```
+ - * div mod
```

Use o seguinte predicado de consulta do XPath para extrair produtos quando o preço for maior que o custo mais impostos:

```
PRODUCT[./PRICE > ./COST + ./TAX]]
```

Funções

Use os seguintes tipos de função em um predicado de consulta do XPath:

- **String.** Use as funções string para testar valores de substring, concatenar strings ou converter strings em outras strings. O seguinte predicado de consulta do XPath determina se o nome completo de um funcionário é igual à concatenação de sobrenome e nome:

```
EMPLOYEE[./FULLNAME=concat(./ENAME/LASTNAME,./ENAME/FIRSTNAME) ]
```

- **Númérico** Use funções numéricas com valores de elemento e de atributo. As funções numéricas operam em números e retornam números inteiros. Por exemplo, o seguinte predicado de consulta do XPath arredonda desconto e testa se o resultado é maior que 15:

```
ORDER_ITEMS[round(./DISCOUNT > 15]
```

- **Booleano.** As funções Boolean retornam true ou false. Use-as para testar elementos, verificar o atributo de linguagem ou forçar um resultado true ou false. Por exemplo, uma string será true se o seu valor for maior que zero:

```
boolean(string)
```

Testando elementos ou atributos

Você poderá determinar se um elemento ou atributo ocorrerá nos dados XML. O seguinte predicado de consulta do XPath determinará se um departamento tem um atributo de nome de departamento:

```
COMPANY/DEPT[@DEPTNAME]/EMPLOYEE
```

Deptname é um atributo. Os atributos são precedidos por “@” em uma expressão de predicado de consulta XML.

Quando você executar uma sessão, o Serviço de Integração extrairá dados de funcionários da origem XML se o departamento do funcionário tiver um nome de departamento. Caso contrário, o Serviço de Integração não extrairá os dados do funcionário.

Regras e diretrizes do predicado de consulta do XPath

Use as seguintes regras e diretrizes quando você criar um predicado de consulta do XPath:

- Você pode configurar um predicado de consulta do XPath para qualquer elemento em uma linha de exibição.

Por exemplo, se uma linha de exibição for Company/Dept, você poderá criar o seguinte predicado de consulta do XPath:

```
COMPANY[./DEPT=100]
```

- Você pode corresponder o conteúdo.
- Você poderá adicionar um predicado de consulta do XPath a uma coluna se ela for colocada abaixo da linha de exibição na hierarquia XML de exibições e o XPath de coluna incluir a linha de exibição.

Por exemplo, se a linha de exibição for Product/Toys[1], você poderá criar o seguinte predicado de consulta do XPath:

```
Product/Toys[1][./Sales > 100]
```

O seguinte exemplo mostra um predicado de consulta do XPath inválido para a linha de exibição Product/Toys[1]:

```
Product/Toys[2][./Sales > 100]
```

Product/Toys[1] é a linha de exibição. Você não pode usar Product/Toys[2].

- Use um elemento ou atributo com uma única ocorrência. Você não pode criar um predicado de consulta do XPath em um elemento com várias ocorrências.
- Você não pode criar uma consulta de predicado do XPath em um elemento com compartimentos, pois um elemento com compartimentos não contém valores.

Etapas para criar um predicado de consulta do XPath

Você pode criar um predicado de consulta do XPath para uma exibição em uma definição de origem XML.

Para criar um predicado de consulta do XPath:

1. Abra uma definição de origem XML no Editor de XML.
2. Selecione uma exibição no espaço de trabalho do Editor de XML.
As colunas de exibição aparecem na janela Colunas.
3. Clique em XPath de Linha de Exibição.
A janela Especificar Predicado de Consulta para XPath é exibida. Você pode inserir um predicado de consulta do XPath na janela Predicado do XPath ou pode escolher elementos, operadores e funções nas guias da caixa de diálogo.
4. Clique na guia Elementos e Atributos Filhos para exibir os elementos e atributos que podem ser adicionados a um predicado de consulta do XPath.
5. Clique duas vezes em um elemento ou atributo para adicioná-lo ao predicado de consulta do XPath.

O componente é exibido no painel.

6. Clique na guia Operadores.

Use os operadores para criar expressões. Você pode comparar elementos com valores ou outros elementos ou pode criar expressões matemáticas.

A seguinte tabela descreve os operadores que podem ser adicionados a um predicado da consulta do XPath:

Operador	Descrição
+	Adicionar
-	Subtrair
*	Multiplicar
div	Dividir
mod	Módulo
and	Booleano e
or	Booleano ou
<	Menor que
<=	Menor que ou igual a
>	Maior que
>=	Maior que ou igual a
=	Igual
!=	Diferente de

7. Clique duas vezes em um operador para adicioná-lo ao predicado de consulta do XPath.
8. Para adicionar as funções do predicado de consulta do XPath do PowerCenter, clique na guia Funções. As funções aceitam argumentos e valores retornados.
9. Selecione outro elemento ou atributo na guia Elementos e Atributos Filhos ou insira um valor na janela Predicado do XPath para completar uma expressão.
10. Clique em Validar para validar o predicado de consulta do XPath.

Mantendo relacionamentos de exibição

Você pode concluir as seguintes tarefas para manter relacionamentos de exibição no Editor de XML:

- **Criar um relacionamento entre exibições.** Defina relacionamentos entre exibições no espaço de trabalho.
- **Criar um relacionamento de tipo.** Defina um relacionamento de tipo entre uma coluna em uma exibição e uma exibição de tipo no espaço de trabalho.
- **Recrie relacionamentos de entidade.** Gere exibições e relacionamentos usando as mesmas opções que no Assistente de XML.

Criando um relacionamento entre exibições

Use o Editor de XML para criar relacionamentos hierárquicos ou de herança entre exibições. Você não pode criar um relacionamento entre uma origem XML e não XML.

Para criar um relacionamento entre exibições XML:

1. Clique com o botão direito do mouse na parte superior da exibição XML filha no espaço de trabalho.
2. Selecione Criar Relacionamento.
3. Mova o ponteiro para a exibição pai para estabelecer um relacionamento.
À medida que você move o ponteiro, um link aparece entre as exibições e o Editor de XML verifica se o relacionamento é válido. Se um relacionamento não for válido, uma mensagem de erro aparecerá na barra de status.
4. Se nenhuma mensagem de erro aparecer, clique na exibição pai para estabelecer o relacionamento.
O Editor de XML cria o relacionamento e adiciona a chave externa adequada.
5. Para exibir detalhes sobre o relacionamento, coloque o cursor sobre o link entre as exibições.
O Editor exibe o tipo de relacionamento e as chaves primárias e externas.

Criando um relacionamento de tipo

Você pode criar um relacionamento de tipo entre uma coluna na exibição e uma exibição de tipo. Se a coluna for dinâmica, você poderá escolher ocorrências para incluir no relacionamento.

Para criar um relacionamento de tipo:

1. Clique com o botão direito na coluna da exibição que você deseja usar.
2. Selecione Criar Relacionamento.
3. Se a coluna for dinâmica, selecione uma ocorrência a ser usada.
4. Mova o ponteiro para a exibição de tipo para estabelecer um relacionamento.
À medida que você move o ponteiro, um link aparece entre as exibições e o Editor de XML verifica se o relacionamento é válido. Se um relacionamento não for válido, uma mensagem de erro aparecerá na barra de status.
5. Se nenhuma mensagem de erro aparecer, clique na exibição pai para estabelecer o relacionamento.
O Editor de XML cria o relacionamento.

Recriando Relacionamentos de Entidade

Você pode recriar relacionamentos de entidade para uma definição XML. Use a caixa de diálogo Recriar Relacionamentos de Entidade para gerar novas exibições XML usando as mesmas opções que no Assistente de XML. Ao regenerar as exibições, você pode optar por manter as exibições existentes. A definição XML contém todas as exibições.

Para recriar relacionamentos de entidade para uma definição XML:

1. Abra a definição XML no Editor de XML.
2. Destaque a raiz de XML no Navegador.
Se você destacar outro componente, o Editor de XML usará esse componente como a raiz.
3. Clique em Exibições XML > Criar Relacionamento de Entidade.
4. Escolha entre as seguintes opções para reduzir a explosão de metadados:
 - **Reduzir explosão de elementos.** Para qualquer elemento de referência ou com várias ocorrências, o Assistente de XML cria uma exibição XML com vários relacionamentos hierárquicos.
 - **Reduzir explosão de tipo complexo.** Para qualquer tipo complexo de referência ou com várias ocorrências, o Assistente de XML cria uma exibição XML com vários relacionamentos de tipo. Se o esquema usar tipos complexos herdados, você também poderá reduzir a explosão de tipos complexos herdados.
 - **Reduzir explosão de tipo complexo.** Para qualquer tipo herdado, o Assistente de XML cria uma exibição XML usando vários relacionamentos de tipo.
 - **Compartilhar exibições XML existentes.** Não remova as exibições XML existentes.
 - **Atualizar exibições XML compartilhadas.** Salve as exibições existentes, mas atualize-as.
5. Clique em Avançar.
A caixa de diálogo Recriar Relacionamentos de Entidade é exibida.
6. Para exibir um componente filho, selecione um elemento compartilhado ou tipo complexo e clique no nome.
7. Para excluir um componente filho, desmarque o elemento no painel Excluir Componentes Filhos.
Para gerar uma nova exibição, selecione o elemento ou o tipo complexo. Ao criar os novos relacionamentos de entidade, você gera uma exibição com esse elemento como uma raiz de exibição.

Exibindo componentes do esquema

Conclua as seguintes tarefas para exibir componentes no Navegador e no espaço de trabalho:

- **Atualizar o namespace.** Altere a localização de um esquema ou o namespace padrão em um destino XML.
- **Navegar para componentes** Localize componentes navegando de um componente para outro ou para outra área da janela Editor de XML.
- **Organizar exibições no espaço de trabalho.** Organizar exibições no espaço de trabalho hierarquicamente. Você pode organizar as exibições em uma disposição hierárquica no espaço de

trabalho. Para organizar as exibições no espaço de trabalho, clique em Layout > Organizar ou clique com o botão direito do mouse no espaço de trabalho e selecione Organizar.

- **Procurar componentes.** Localize componentes no Navegador ou no espaço de trabalho.
- **Exiba a hierarquia de tipos simples ou complexos.** Exiba uma hierarquia dos tipos simples ou complexos no esquema XML.
- **Exibir metadados XML.** Exiba um arquivo XML, esquema ou DTD que o Editor de XML cria a partir da definição XML.
- **Visualizar dados XML.** Mostre uma exibição XML usando exemplo de dados de um arquivo XML externo.
- **Validar a definição XML.** Valide a definição XML e exiba erros.

Atualizando um namespace

Ao criar uma definição XML, você pode alterar o prefixo do namespace e o local do esquema no Editor de XML. Também é possível adicionar um esquema ao namespace.

Se criar uma definição XML de destino que tenha um ou mais namespaces, você poderá escolher um namespace padrão. Quando você executa uma sessão, o Serviço de Integração grava os elementos e os atributos do namespace padrão sem um prefixo de namespace.

Não use “xml” ou “xmlns” como prefixo de namespace. Um prefixo “xml” aponta para o namespace <http://www.w3.org/XML> por padrão. “Xmlns” vincula elementos para namespaces em um esquema XML.

Nota: Você não pode adicionar um namespace usando o Editor de XML.

Para atualizar um namespace:

1. Selecione um elemento no Navegador.
2. Clique na guia Propriedades.
3. Clique no link de namespace.

A caixa de diálogo Editar Prefixo do Namespace e Local do Esquema é exibida.

4. Para alterar o prefixo ou local do esquema, selecione o texto que você deseja alterar e insira o novo valor.
5. Para adicionar mais de um esquema a um namespace, selecione um local de esquema e clique em Adicionar.

Um esquema em branco aparece no namespace Local do Esquema.

6. Para excluir um esquema, realce o local do esquema e clique em Excluir.
7. Para criar um namespace padrão no destino XML, selecione um namespace.

Todos os componentes do namespace que você selecionar pertencem ao namespace padrão no destino XML. Quando você executa uma sessão, o Serviço de Integração não grava o prefixo do namespace padrão no arquivo de destino XML.

Navegando para componentes

Para localizar componentes rapidamente, em definições XML grandes, selecione um componente do espaço de trabalho a partir do qual navegar e selecione uma opção de navegação. Por exemplo, se clicar em uma chave externa em uma exibição, você poderá navegar para a chave primária associada ou para a coluna na janela Colunas. Você pode navegar entre componentes no espaço de trabalho, na janela Colunas e no Navegador.

Para navegar para componentes:

1. Clique com o botão direito do mouse em um componente no espaço de trabalho ou na janela Colunas.
2. Selecione Navegar para.
3. Selecione uma opção disponível.

Você pode selecionar entre as seguintes opções, dependendo do componente entre os quais seleciona:

- **Componente de esquema.** Destaca o componente no Navegador.
- **Coluna PowerCenter.** Destaca a coluna na janela Colunas.
- **Chave primária** Destaca a chave primária associada a uma chave externa selecionada.
- **Coluna mencionada.** Destaca a coluna mencionada associada a uma porta de passagem em uma transformação de Analisador ou Gerador de XML.
- **Navegador do XPath.** Exibe o caminho para o componente selecionado.
- **Exibição XML.** Destaca uma exibição no espaço de trabalho que contém a coluna selecionada na janela Colunas.

Procurando componentes

Você pode procurar componentes em uma definição XML. O Editor de XML exibe todas as ocorrências dos componentes. Você pode clicar em um resultado de pesquisa para que o Editor de XML destaque o componente no esquema ou na exibição. O esquema XML ou as exibições XML podem ser pesquisados no espaço de trabalho.

Pesquisando o esquema XML

Você pode pesquisar componentes por nome, tipo e namespace. Você pode especificar as propriedades a serem pesquisadas, como uma anotação, um valor fixo ou padrão ou um comprimento. Você pode pesquisar valores legais usando a propriedade de enumeração.

Para pesquisar componentes no esquema:

1. Clique em Editar > Pesquisar no Esquema.
A caixa de diálogo Pesquisar Componentes é exibida.
2. Para pesquisar por propriedades de componentes, clique no link Opções Avançadas para exibir as Propriedades que podem ser pesquisadas.
3. Insira um nome, um tipo ou uma propriedade a ser pesquisada.
4. Se você deseja pesquisar em um namespace específico, clique em Tudo e selecione um namespace na lista.

5. Clique em Pesquisar.

Os resultados da pesquisa aparecem na caixa de diálogo.

6. Clique em um resultado da pesquisa para exibir o componente na janela Propriedades.

Pesquisando em Exibições XML

Você pode pesquisar componentes e colunas nas exibições XML. Se pesquisar por nome de componente, você poderá localizar todas as colunas associadas na definição. Por exemplo, se você pesquisar um componente "Number", os resultados conterão as exibições e as colunas que possuem o componente "Number".

Para pesquisar usando uma chave parcial, insira os primeiros caracteres do nome da coluna ou do componente.

Para pesquisar componentes em exibições XML:

1. Clique em Editar > Pesquisar Exibições XML.

A caixa de diálogo Pesquisar Exibições XML e Colunas é exibida.

2. Insira os critérios de pesquisa.

Você pode pesquisar todos os tipos de colunas, tipos de colunas regulares, chaves geradas ou outros tipos. Outros tipos incluem colunas FileName, portas de referência e campos de passagem.

3. Clique em Pesquisar.

Os resultados da pesquisa aparecem na caixa de diálogo.

4. Para limpar os campos de pesquisa, clique em Nova Pesquisa.

Exibindo uma hierarquia de tipo simples ou complexo

Você pode exibir uma hierarquia dos tipos simples de XML na definição de esquema. Para exibir uma hierarquia de tipos simples, clique em Exibir > Hierarquia SimpleType. Uma janela exibe a hierarquia dos tipos simples.

Você pode exibir uma hierarquia dos tipos complexos na definição de esquema. Para exibir uma hierarquia de tipos complexos, clique em Exibir > Hierarquia ComplexType. Uma janela exibe a hierarquia dos tipos complexos no esquema. Selecione um componente na janela Hierarquia ComplexType para navegar até o componente no esquema.

Exibindo metadados XML

Você pode exibir uma definição XML como um arquivo XML, DTD ou de esquema XML.

Para exibir metadados XML:

1. Para exibir os metadados como um documento XML de exemplo, escolha um componente global no Navegador.
2. Clique em Exibir > Metadados XML.
A caixa de diálogo Exibir Metadados XML é exibida.
3. Escolha exibir a definição XML como um arquivo XML, um arquivo DTD ou um esquema XML.

Se você usar vários namespaces, escolha um deles para ser usado.

Um aplicativo ou editor de texto padrão exibe os metadados.

4. Para salvar uma cópia do arquivo XML, DTD ou de esquema XML, clique em Salvar Como.
5. Insira um novo nome de arquivo.

Se o aplicativo de exibição padrão for um editor de texto, você precisará incluir o sufixo correto do arquivo com o nome do arquivo. O sufixo é .xml, .dtd ou .xsd, dependendo do tipo de arquivo que você está trabalhando.

Validando definições XML

Você pode validar uma definição XML no Editor de XML.

Para validar uma definição XML:

1. Abra a definição no Editor de XML.
 2. Clique dentro do espaço de trabalho.
 3. Clique em Exibições XML > Validar Definição XML.
- A janela Validar exibe os resultados.

Definindo opções de exibição XML

Por padrão, o Serviço de Integração gera linhas para cada exibição que tenha dados na linha de exibição. Para alterar como o Serviço de Integração gera linhas para algumas exibições de definição de origem XML, selecione Opções de Exibição em XML na janela Colunas XML.

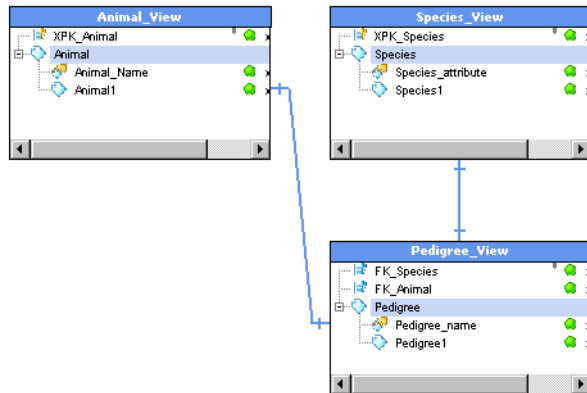
Use os métodos a seguir para determinar quando o Serviço de Integração gera linhas de uma origem XML:

- **Gerar todas as chaves externas em uma exibição.** Por padrão, o Serviço de Integração gera valores para uma chave externa em uma exibição. Se uma exibição tiver mais de uma chave externa, as outras chaves externas terão valores nulos. Você pode gerar valores para todas as chaves externas.
- **Interromper leituras recorrentes em um relacionamento circular.** Por padrão, o Serviço de Integração gera linhas para todas as ocorrências de dados em um relacionamento circular. Você pode gerar uma linha apenas para a primeira ocorrência de dados recorrentes.
- **Gerar uma linha para uma exibição filho se um pai existir.** Por padrão, o Serviço de Integração cria linhas para todas as exibições com dados na linha de exibição. Você só pode gerar uma linha para uma exibição filho quando uma exibição pai não possuir dados.
- **Gerar uma linha para uma exibição sem dados da linha de exibição.** Por padrão, o Serviço de Integração gera dados para uma exibição quando a linha de exibição possui dados. Você pode gerar uma linha para uma exibição que não tenha dados na linha de exibição.
- **Gerar linhas para tipos complexos específicos em um relacionamento de tipo.** Por padrão, o Serviço de Integração gera linhas para todas as exibições na definição XML. Você pode gerar linhas para exibições específicas nos relacionamentos de tipo.

Gerando todas as chaves externas da hierarquia

Por padrão, o Serviço de Integração gera valores para uma chave externa em uma exibição. Se uma exibição tiver mais de uma chave externa, as outras chaves externas terão valores nulos. Você pode optar por gerar valores de chave para todas as chaves externas em uma exibição. Selecione a opção Todas as Chaves Externas da Hierarquia.

A figura a seguir mostra a Pedigree_View com duas chaves externas, FK_Species e FK_Animal:



Se você selecionar a opção Todas as Chaves Externas da Hierarquia da Pedigree_View, o Serviço de Integração gera valores de chave para FK_Species e FK_Animal.

A seguinte figura mostra exemplo de dados para Pedigree_View com a opção Todas as Chaves Externas da Hierarquia:

Pedigree_...	FK_Species	FK_Animal	Pedigree1
Plains Cheeta	1	1	100
African Lion	2	2	200

Se você desmarcar a opção Todas as Chaves Externas da Hierarquia, o Serviço de Integração gerará valores de chave para uma coluna de chave externa. Neste exemplo, o Serviço de Integração gera valores para FK_Species, pois Species_View é o pai mais próximo de Pedigree_View na hierarquia XML. A chave externa FK_Animal tem valores nulos.

A seguinte figura mostra exemplo de dados para Pedigree_View se você desmarca a opção. Todas as Chaves Externas da Hierarquia:

Pedigree_...	FK_Species	FK_Animal	Pedigree1
Plains Cheeta	1	NULL	100
African Lion	2	NULL	200

O Serviço de Integração gera chaves externas para o pai mais próximo.

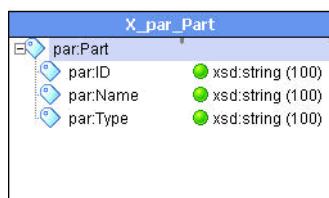
Gerando linhas em relacionamentos circulares

Por padrão, o Serviço de Integração gera linhas para todas as ocorrências de dados em um relacionamento circular. Você pode optar por criar uma linha apenas para a primeira ocorrência. Selecione a opção de linha Não Recursiva.

O seguinte arquivo XML contém um elemento Part com uma referência circular:

```
<?xml version="1.0" encoding="utf-8"?>
<Vehicle xmlns="http://www.PartInvoice.org"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.PartInvoice.org part.xsd">
  <VehInfo>
    <make>Honda</make>
    <model>Civic</model>
    <type>Compact</type>
  </VehInfo>
  <Part>
    <ID>123</ID>
    <Name>Body</Name>
    <Type>Exterior</Type>
    <Part>
      <ID>111</ID>
      <Name>DoorFL</Name>
      <Type>Exterior</Type>
      <Part>
        <ID>1112</ID>
        <Name>KnobL</Name>
        <Type>Exterior</Type>
      </Part>
      <Part>
        <ID>1113</ID>
        <Name>Window</Name>
        <Type>Exterior</Type>
      </Part>
    </Part>
  </Part>
</Vehicle>
```

A figura a seguir mostra que o elemento Parte é a linha de exibição X_par_Part na definição de XML:



A figura a seguir mostra que os Serviços de Integração geram linhas para Parte 123 e todas as partes do componente quando você executa uma sessão:

☐ NonRecursive Row

par_ID	par_Name	par_Type
123	Body	Exterior
111	DoorFL	Exterior
112	DoorFR	Exterior
113	DoorRL	Exterior

A figura a seguir mostra que o Serviço de Integração lê a primeira ocorrência do elemento Parte e gera uma linha de dados para Parte 123 quando você seleciona Linha NonRecursive:

☒ NonRecursive Row

par_ID	par_Name	par_Type
123	Body	Exterior

Gerando linhas de relacionamento de hierarquia

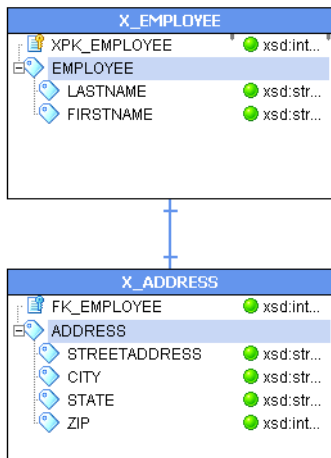
Por padrão, o Serviço de Integração cria linhas para todas as exibições com dados na linha de exibição. Selecione Linha de Relacionamento de Hierarquia para gerar uma linha para uma exibição filho se a exibição filho tiver dados correspondentes em um relacionamento de hierarquia. A exibição pai deve ter dados para gerar uma linha para a exibição filho.

Por exemplo, uma definição XML pode ter uma hierarquia que consiste em uma exibição Employee e uma exibição Address. Employee é a exibição pai. Os dados de endereço podem incluir Employee \Addresses ou Store\Addresses. Você pode optar por obter a saída de Employee\Address.

O arquivo XML a seguir tem um endereço dentro do elemento Store e um endereço dentro do elemento Employee:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE STORE >
<STORE SID="BE1752">
  <SNAME>Mud and Sawdust Furniture Store</SNAME>
  <ADDRESS>
    <STREETADDRESS>335 Westshore Road</STREETADDRESS>
    <CITY>Fausta City</CITY>
    <STATE>CA</STATE>
    <ZIP>97584</ZIP>
  </ADDRESS>
  <EMPLOYEE DEPID="34">
    <ENAME>
      <LASTNAME>Bacon</LASTNAME>
      <FIRSTNAME>Allyn</FIRSTNAME>
    </ENAME>
    <ADDRESS>
      <STREETADDRESS>1000 Seaport Blvd</STREETADDRESS>
      <CITY>Redwood City</CITY>
      <STATE>CA</STATE>
      <ZIP>94063</ZIP>
    </ADDRESS>
    <EPHONE>(408) 226-7415</EPHONE>
    <EPHONE>(650) 687-6831</EPHONE>
  </EMPLOYEE>
</STORE>
```

A figura a seguir mostra um relacionamento hierárquico entre a exibição Funcionário e a exibição Endereço:



A exibição Funcionário está conectada à exibição Endereço com uma linha azul que define um relacionamento um a um entre o pai e a exibição filho. A exibição Funcionário tem uma chave primária XPK_Employee e um elemento Funcionário que consiste em LastName e FirstName. A exibição Endereço tem uma chave externa, FK_Employee e um elemento Endereço que consiste em StreetAddress, Cidade, Estado e Cep.

Por padrão, o Serviço de Integração gera uma linha para cada ocorrência do elemento Address. O Serviço de Integração gera um linha para Store\Address e outra para Employee\Address.

A imagem a seguir mostra os dados XML de Address se você desmarcar a opção Linha de Relacionamento de Hierarquia:

☐ Relacionamento de Hierarquia

STREETADDRESS	FK_EMPL...	CITY	STATE	ZIP
335 Westshore Road	NULL	Fausta City	CA	97584
1000 Seaport Blvd	1	Redwood City	CA	94063

Quando você seleciona a opção Linha de Relacionamento de Hierarquia, o Serviço de Integração gera linhas em uma sessão da seguinte maneira:

- O Serviço de Integração gera uma linha para a exibição Address quando a exibição Employee tem dados correspondentes em uma sessão.
- O Serviço de Integração gera uma linha que representa o relacionamento da hierarquia Employee \Address.
- O Serviço de Integração não gera uma linha para Store\Address.

A linha seguinte mostra os dados de Address se você tiver selecionado a opção Linha de Relacionamento de Hierarquia:

☒ Relacionamento de Hierarquia

STREETADDRESS	FK_EMPL...	CITY	STATE	ZIP
1000 Seaport Blvd	1	Redwood City	CA	94063

Configurando a opção Forçar Linha

Por padrão, o Serviço de Integração gera dados para uma exibição quando a linha de exibição possui dados. Selecione Forçar Linha para gerar uma linha para a exibição XML quer o elemento de linha de exibição contenha dados ou não. Por exemplo, se uma linha de exibição for endereço\CEP, você poderá optar por gerar o endereço, mesmo que a linha de exibição não possua dados de CEP.

A figura a seguir mostra o elemento CEP como linha de exibição:



O elemento cep é realçado no final da lista, com as linhas de elementos de rua, cidade e estado listadas acima dele.

Por padrão, o Serviço de Integração gera uma linha para cada ocorrência do elemento cep no elemento endereço.

Por exemplo, você pode processar o arquivo XML a seguir em uma sessão:

```
<?xml version="1.0" ?>
<company xmlns="http://www.example.org"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.example.org forcerow.xsd" >
  <name>company1</name>
  <address>
    <street>stree1</street>
    <city>city1</city>
    <zip>1001</zip>
  </address>
  <employee>
    <name>emp1</name>
    <address>
      <street>emp1_street</street>
      <city>emp1_city</city>
    </address>
  </employee>
  <employee>
    <name>emp2</name>
    <address>
      <street>emp2_street</street>
      <city>emp2_city</city>
      <zip>2001</zip>
    </address>
  </employee>
</company>
```

Por padrão, o Serviço de Integração gera as linhas stree1 e emp2_street para a exibição Endereço:

A figura a seguir mostra as linhas stree1 e emp2_street para a exibição Endereço:

☐ Force Row

STREET	CITY	ZIP
stree1	city1	1001
emp2_street	emp2_city	2001

O Serviço de Integração não gera uma linha para emp1, porque a linha de exibição é CEP e emp1 não possui dados para o elemento CEP.

Se você ativar Forçar Linha, poderá obter a saída dos elementos rua e cidade com ou sem o CEP. A sessão gera uma linha para emp1, embora emp1 não possua dados para o elemento CEP.

A figura a seguir mostra as linhas que o Serviço de Integração gera quando você ativa Forçar Linha:

☒ Force Row

STREET	CITY	ZIP
street	city1	1001
emp1_street	empl_city	NULL
emp2_street	emp2_city	2001

Gerando Linhas para exibições em Relacionamentos de Tipo

Por padrão, o Serviço de Integração gera linhas para todas as exibições em um relacionamento de tipo. Escolha a opção Linha de Relacionamento de Tipo para gerar linhas de um tipo complexo específico em um relacionamento de tipo. Defina a Linha de Relacionamento de Tipo para cada exibição que você deseja gerar.

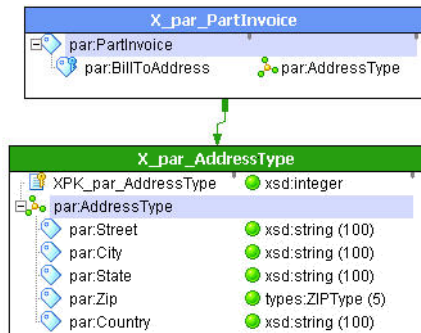
Por exemplo, uma definição pode ter uma hierarquia que inclui BillToAddress e ShipToAddress. Se você quiser gerar linhas para BillToAddress, use a opção Linha de Relacionamento de Tipo.

Definindo um relacionamento de tipo

O esquema a seguir define BillToAddress e ShipToAddress. BillToAddress é um AddressType e ShipToAddress é um USAddressType. USAddressType estende AddressType.

```
<xsd:sequence>
  <xsd:element name="Street" type="xsd:string" />
  <xsd:element name="City" type="xsd:string" />
  <xsd:element name="State" type="xsd:string" />
  <xsd:element name="Zip" type="types:ZIPTYPE" />
  <xsd:element name="Country" type="xsd:string" />
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="USAddressType">
  <xsd:complexContent>
    <xsd:extension base="AddressType">
      <xsd:sequence>
        <xsd:element name="PostalCode" type="xsd:string" minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<xsd:element name="BillToAddress" type="AddressType" />
<xsd:element name="ShipToAddress" type="USAddressType" />
```

A figura a seguir mostra um relacionamento de tipo na definição de XML com a exibição PartInvoice mostrando o elemento BillToAddress na exibição XML X_par_PartInvoice e a exibição XML relacionada X-par_AddressType abaixo:



A exibição PartInvoice contém dados de fatura. A exibição inclui um BillToAddress. O relacionamento de tipo na definição XML é entre BillToAddress e AddressType.

Para limitar os dados de AddressType a BillToAddress, selecione a exibição X_par_PartInvoice no espaço de trabalho do Editor de XML. Escolha a opção Linha de Relacionamento de Tipo. Quando você executa uma sessão, o Serviço de Integração gera as linhas Address para BillToAddress, mas não ShipToAddress. ShipToAddress não está no relacionamento de tipo.

Exemplo

O exemplo a seguir mostra como limitar dados a tipos específicos em um relacionamento de tipos. O exemplo usa as exibições PartInvoice e AddressType.

O seguinte arquivo XML contém dados de fatura que incluem BillToAddress e ShipToAddress:

```

<xsd:complexType name="AddressType">
<?xml version="1.0" encoding="utf-8"?>
<Invoices xmlns="http://www.PartInvoice.org"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.PartInvoice.org Part.xsd">
<PartInvoice InvoiceNum="185" DateShipped="2005-01-01">
  <PartOrder>
    <PartID>HLG100</PartID>
    <PartName>Halogen Bulb</PartName>
    <Quantity>2</Quantity>
    <UnitPrice>35</UnitPrice>
  </PartOrder>
  <BillToAddress>
    <Street>2100 Seaport Blvd</Street>
    <City>Redwood City</City>
    <State>CA</State>
    <Zip>94063</Zip>
    <Country>USA</Country>
  </BillToAddress>
  <ShipToAddress xsi:type="USAddressType">
    <Street>3350 W Bayshore Rd</Street>
    <City>Palo Alto</City>
    <State>CA</State>
    <Zip>97890</Zip>
    <Country>USA</Country>
    <PostalCode>97890</PostalCode>
  </ShipToAddress>

```

```
</PartInvoice>
</Invoices>
```

Quando você executa uma sessão, o Serviço de Integração gera uma linha X_par_AddressType para cada AddressType.

A figura a seguir mostra as linhas BillToAddress e ShipToAddress que o Serviço de Integração gera por padrão:

☐ Relacionamento de Tipo Row

XPK_par_...	par_Street	par_City	par_State	par_Zip	par_Country
1	2100 Seaport Blvd	Redwood ...	CA	94063	USA
2	3350 W Bayshore Rd	Palo Alto	CA	97890	USA

Para gerar as linhas AddressType relacionadas à exibição PartInvoice, defina a Opção de Linha de Relacionamento de Tipo para a exibição PartInvoice.

A figura a seguir mostra a linha BillToAddress que a Opção de Linha de Relacionamento de Tipo gera:

☒ Relacionamento de Tipo Row

XPK_par_...	par_Street	par_City	par_State	par_Zip	par_Country
1	2100 Seap...	Redwood ...	CA	94063	USA

Ela não gera uma linha para ShipToAddress, pois ShipToAddress não está no relacionamento de tipo.

Solucionando problemas de trabalho com o Editor de XML

Quando valido uma definição XML, é exibido um erro informando que a definição XML é muito grande. Por que isso ocorre?

Se importar um arquivo XML que tenha componentes definidos com comprimentos infinitos, você poderá exceder facilmente o limite de 500 MB para o comprimento total da coluna. É possível alterar os comprimentos de coluna no Editor de XML ou definir uma opção para substituir todos os comprimentos infinitos e reimportar o arquivo.

Não consigo localizar o arquivo DTD ou de esquema XML que criei ao exibir os metadados XML.

O arquivo DTD ou de esquema XML que você pode exibir é um arquivo temporário que o Designer cria para exibição. Se você quiser usar o arquivo para outras finalidades, salve-o com outro nome e diretório quando o exibir.

Quando adiciono colunas às exibições de origem XML, a hierarquia no arquivo XML de origem permanece a mesma.

Ao adicionar colunas às exibições de origem XML, você não adiciona elementos à hierarquia subjacente. A hierarquia XML que você importa permanece a mesma, independentemente de como você cria as exibições ou mapeia as colunas em uma exibição para os elementos na hierarquia. Você pode modificar os tipos de dados e a cardinalidade dos elementos, mas não pode modificar a estrutura da hierarquia.

Para obter informações sobre dimensionamento de XML no PowerCenter, consulte [“Visão Geral do Uso do XML com o PowerCenter” na página 33](#). Para obter mais informações sobre as limitações que se aplicam ao tratamento de XML no PowerCenter, consulte [“Limitações” na página 33](#).

Para criar uma transformação com outros tipos de elementos e para transformar arquivos de entrada XML maiores, use uma transformação de Processador de Dados. Para obter mais informações sobre como criar transformações de Processador de Dados, consulte o *Guia do Usuário do Informatica Data Transformation* e o *Guia de Introdução do Informatica Data Transformation*.

Trabalhando com Destinos XML

Visão geral do trabalho com destinos XML

Você pode criar definições de destino XML das seguintes maneiras:

- **Importe a definição de um esquema ou arquivo XML.** É possível criar uma definição de destino a partir de um arquivo XML, DTD ou de esquema XML. É possível importar definições de arquivo XML de uma URL ou de um nó local. Se você importar um arquivo XML com um DTD associado, o Assistente de XML usará o DTD para gerar o arquivo.
- **Crie uma definição de destino XML com base em uma definição de fonte XML.** Você pode arrastar uma definição de fonte XML para o Target Designer. Se você criar uma definição de destino XML, o Designer criará uma definição de destino com base na hierarquia da definição XML.
- **Crie um destino XML com base em uma definição de arquivo relacional.** Você pode importar uma definição de destino XML de uma definição de repositório de arquivo simples ou relacional.

Além de criar definições de destino XML, você pode realizar as seguintes tarefas com destinos XML no Target Designer:

- **Editar propriedades de destino.** Editar uma definição de destino XML para adicionar comentários documentando alterações em arquivos de esquema XML, DTD ou XML de destino.
- **Sincronizar definições de destino.** Você pode sincronizar a definição XML de destino em um esquema se precisar fazer alterações. Quando sincroniza a definição, você atualiza a definição XML em vez de importar o esquema se ele for alterado.

Importando uma definição de destino XML de um arquivo XML

Você pode importar definições XML de um esquema XML ou de um arquivo DTD ou XML. Você pode importar arquivos locais ou arquivos aos quais faz referência com uma URL. Para garantir que o Designer forneça uma definição precisa dos dados, importe uma definição de destino de um esquema XML.

Você pode escolher uma das seguintes opções para criar exibições XML:

- **Criar relacionamentos de entidade.** Use esta opção para criar exibições de elementos com várias ocorrências ou com referência e tipos complexos. Você cria relacionamentos entre exibições, em vez de criar uma hierarquia grande.

- **Criar relacionamentos hierárquicos.** Use esta opção para criar uma raiz e expandir os componentes XML sob a raiz. Você pode optar por criar exibições normalizadas ou desordenadas. Se você escolher normalizado, todos os elementos ou atributos aparecerão apenas uma vez. Relações um para muitos se tornam exibições XML separadas com chaves para relacionar as exibições. Se você criar exibições XML desordenadas, todos os elementos e atributos serão exibidos em um grupo hierárquico.
- **Criar exibições XML personalizadas.** Use esta opção para selecionar vários elementos globais como raízes para exibições XML e selecionar opções para reduzir a explosão de metadados.
- **Ignorar criação de exibições XML.** Use esta opção para importar a definição sem criar nenhuma exibição. Se você escolher esta opção, use o Editor de XML para criar exibições XML no espaço de trabalho posteriormente.
- **Sincronizar definições XML.** Use esta opção para atualizar uma ou mais definições XML quando seus esquemas adjacentes forem alterados.

Sugestão: Importe arquivos de esquema DTD ou XML, em vez de arquivos XML. Se você importar um arquivo XML com um DTD associado, o Assistente de XML usará o DTD.

Para importar definições de destino XML:

1. No Target Designer, clique em Destinos > Importar Definição XML.
A janela Importar Definições XML é aberta. Os arquivos de esquema da pasta local são exibidos por padrão.
2. Clique em Arquivo Local ou URL para procurar arquivos XML.
3. Para procurar arquivos DTD ou XML, selecione a extensão de arquivo adequada na lista Arquivos de Tipo.

Criando um Destino a partir de uma Definição de Origem XML

Quando quiser criar uma definição de destino que seja muito semelhante a uma definição de origem existente, use a definição de origem ou um atalho para a definição de origem a fim de criar a definição de destino. Arraste a definição de origem XML para o Target Designer de modo a criar uma definição de destino XML ou uma definição de destino relacional.

Use as seguintes diretrizes para criar definições de destino XML:

- Quando você cria uma definição de destino XML a partir de uma definição de origem XML, uma duplicata da definição de origem XML é criada.
- Uma definição de origem XML válida não cria necessariamente uma definição de destino XML válida. Para garantir que uma definição de destino válida seja criada, valide o destino.

Nota: Definições de destino XML não podem conter colunas dinâmicas.

Use as seguintes diretrizes para criar definições de destinos relacionais:

- Se você criar uma definição de destino relacional, o Designer criará as definições de destino relacional com base nos grupos da definição de origem XML. Cada grupo na definição de origem XML se torna uma definição de destino.
- O Designer cria o mesmo relacionamento entre as definições de destino existente entre os grupos na definição de origem.

Para criar uma definição de destino a partir de uma definição de origem XML:

1. Arraste uma definição de origem XML do Navegador até o espaço de trabalho Target Designer.

A caixa de diálogo Exportação de XML é exibida.

2. Selecione para criar um destino relacional ou XML. Clique em OK.

A definição de destino é exibida no espaço de trabalho do Target Designer. Se você selecionar destinos relacionais, mais de uma definição de destino poderá aparecer no espaço de trabalho, dependendo da origem.

Editando Propriedades da Definição de Destino XML

Depois de criar uma definição de destino XML, você poderá editar as propriedades para que reflitam as alterações feitas nos dados de destino, adicionar nomes e comentários comerciais ou alterar a página de código.

Para editar propriedades de definição do destino XML:

1. Abra o destino XML no Target Designer.
2. Clique com o botão direito do mouse e selecione Editar.
3. Na guia Tabela, edite as configurações.

A tabela a seguir descreve as configurações de Tabela:

Configurações de Tabela	Descrição
Selecionar Tabela	O nome da definição de destino. Para alterar o nome, clique no botão Renomear.
Nome Comercial	Nome descritivo da tabela de destino. Edite o Nome Comercial usando o botão Renomear.
Restrições	Não aplicável a destinos XML. Qualquer entrada é ignorada.
Opções de Criação	Não aplicável a destinos XML. Qualquer entrada é ignorada.
Descrição	Descrição da tabela de destino. O limite de caracteres é de 2.000 bytes/K, em que K é o número máximo de bytes para cada caractere na página de código do repositório. Insira os links para a documentação comercial.
Página de Código	Selecione a página de código a ser usada na definição de destino.
Tipo de Banco de Dados	Indica que a definição de destino é um destino XML.
Palavras-chave	Use palavras-chave para organizar e identificar destinos. As palavras-chave podem incluir nomes de desenvolvedor, mapeamentos ou nomes de esquema XML. Use palavras-chave para executar pesquisas no Repository Manager.

4. Na guia Colunas, você pode exibir definições de coluna XML.

A tabela a seguir descreve as configurações de Colunas:

Configurações de Colunas	Descrição
Selecionar Tabela	Exibe a definição de destino que você está editando. Para escolher uma definição diferente a ser editada, selecione uma na lista de definições disponível no espaço de trabalho.
Nome da Coluna	Nome da coluna.
Tipo de Dados	O tipo de dados do PowerCenter para a coluna.
Precisão	Tamanho da coluna. Você pode alterar a precisão para alguns tipos de dados, como string.
Escala	Número máximo de dígitos após o ponto decimal de valores numéricos.
Não Nulo	Indica se a coluna pode conter valores nulos.
Tipo de Chave	Tipo de chave que o Assistente de XML gera para vincular as exibições.
XPath	Caminho até a hierarquia de arquivo XML que localiza um item.

5. Na guia Propriedades, você pode modificar os atributos de transformação da definição de destino. Se usar uma sessão de confirmação baseada na fonte ou uma transformação Controle de Transação com o destino XML, você poderá definir como desejará liberar os dados para o destino. A tabela a seguir descreve os atributos que você pode editar:

Configurações de Colunas	Descrição
Selecionar Tabela	Exibe a definição de origem que você está editando. Para escolher uma definição de origem diferente para edição, selecione-a na lista.
Manipulação de Linha de Grupo Duplicada	Escolha uma destas opções para manipular o processamento de linhas duplicadas no destino: <ul style="list-style-type: none"> - Primeira Linha. O Serviço de Integração transfere a primeira linha duplicada ao destino. As linhas a seguir com a mesma chave primária são rejeitadas. - Última Linha. O Serviço de Integração transfere a primeira linha duplicada para o destino. - Erro. O Serviço de Integração transfere a primeira linha para o destino. As linhas com chaves primárias duplicadas incrementam a contagem de erros. A sessão falha quando a contagem de erros atinge o limite de erros.
Referência de DTD	O nome do arquivo DTD ou de esquema XML do arquivo XML de destino. O Serviço de Integração adiciona a declaração de tipo de documento ao arquivo XML quando você cria o arquivo XML.

Configurações de Colunas	Descrição
Na Confirmação	<p>O Serviço de Integração pode gerar vários arquivos XML ou anexar a um arquivo XML após uma confirmação. Use uma das seguintes opções:</p> <ul style="list-style-type: none"> - Ignorar Confirmação. O Serviço de Integração cria um arquivo XML e grava nele no fim do arquivo. - Criar Novo Documento. Cria um novo arquivo XML em cada confirmação. - Anexar ao Documento. Grava no mesmo arquivo XML após cada confirmação.
Diretório de Cache	Diretório para os arquivos de cache do destino XML. O padrão é a variável de processo do serviço \$PMCacheDir.
Tamanho do Cache	O tamanho total em bytes do cache de destino XML. O padrão é auto.

- Na guia Extensões de Metadados, é possível criar, modificar, excluir e promover extensões de metadados não reutilizáveis e atualizar seus valores. Você também pode atualizar os valores de extensões de metadados reutilizáveis.
- Clique em OK.
- Clique em Repositório > Salvar.

Validando Destinos XML

Você pode criar exibições XML personalizadas que descrevam como extrair dados para um arquivo XML. No entanto, nem todas as estruturas de exibição ou relacionamentos entre exibições são válidos em uma definição XML. Algumas estruturas de exibição podem ser válidas para uma origem XML, mas não para um destino XML. O Designer impede a criação de definições ambíguas.

O PowerCenter valida as exibições XML de destino quando você executa as seguintes tarefas:

- O Designer executa a validação limitada quando você salva ou busca um destino XML no repositório.
- O Editor de XML valida cada etapa quando você edita o XML no espaço de trabalho do XML.
- Você pode optar por validar uma definição de destino que está editando no Editor de XML.
- O Designer valida conexões de destino XML quando valida mapeamentos.

O Designer usa regras para validar relacionamentos de hierarquia, de tipo e de herança.

Nota: O Serviço de Integração não valida a instância de destino XML em um esquema em uma sessão. Você pode definir a propriedade da sessão Validar Destino para validar os tipos simples nos dados.

Validação de Relacionamento de Hierarquia

O Designer usa as seguintes regras para validar relacionamentos de hierarquia:

- Uma exibição que tem raiz em um tipo não pode ser uma exibição autônoma. A exibição deve ser um filho em um relacionamento de herança ou a exibição deve ter um relacionamento de tipo com outra exibição. Um destino XML será inválido se ele não tiver exibições com raiz em um elemento.
- Você deve conectar uma exibição com uma linha de exibição com várias ocorrências a outra exibição.
- Duas exibições não podem ter a mesma linha de exibição efetiva.

- Um destino XML torna-se inválido quando não tem a raiz de exibição em um elemento.
- Você pode separar as exibições pai e filha por outros elementos, mas se você tiver uma opção de dois pais para uma exibição, deverá usar a mais próxima. Determine o pai mais próximo pelo caminho da linha de exibição efetiva. Um pai vem antes do outro no caminho. Escolha a exibições que vem em segundo lugar no caminho.
- Você deve conectar todas as exibições à mesma raiz de exibição na mesma hierarquia. A definição não pode conter várias árvores para a mesma raiz de exibição.
- Uma exibição XML poderá ter um relacionamento hierárquico com ela mesma se a linha de exibição e a raiz da exibição forem idênticas para a exibição.

Validação de Relacionamento de Tipo

Um relacionamento de tipo é um relacionamento entre uma coluna e uma exibição. Um relacionamento de tipo não é um relacionamento entre duas exibições. As seguintes regras se aplicam a relacionamentos de tipo:

- Uma coluna na exibição V1 poderá ter um relacionamento de tipo com uma exibição V2 se as raízes da exibição forem do mesmo tipo ou se o tipo de raiz da exibição V2 for derivado da raiz de exibição V1. As duas raízes de exibição devem ser tipos complexos globais.
- Se uma coluna em uma exibição tiver um relacionamento de tipo com outra exibição, você não poderá expandir a coluna.

Validação de herança

Você pode criar dois tipos de relacionamento de herança com exibições XML:

- **Herança de exibição para exibição.** Uma exibição é um tipo derivado de outra exibição. As duas exibições devem ter raízes de exibição complexas globais.
Uma exibição pode ter um relacionamento de herança com outra exibição se a raiz da exibição for um tipo complexo derivado do tipo de raiz de exibição da outra exibição.
Uma exibição pode ser um pai em vários relacionamentos de herança, mas uma exibição pode ser um filho em apenas um relacionamento de herança.
- **Herança de coluna para exibição.** A coluna é um elemento de um tipo complexo local, Type1, e a raiz da exibição está em um tipo complexo global, Type2. Type1 é derivado de Type2.
Uma coluna em uma exibição poderá ter um relacionamento de herança com outra exibição se a coluna for um tipo complexo local e o tipo for derivado do tipo de raiz de exibição da outra exibição.
Se a coluna em uma exibição, V1, tiver um relacionamento de herança com uma exibição, V2, você não poderá colocar o conteúdo de V2 na exibição V1.

Usando um Destino XML em um Mapeamento

Ao adicionar um destino XML a um mapeamento, você precisa seguir as diretrizes de mapeamento para transformações de vários grupos.

Os seguintes componentes afetam o modo de mapear um destino XML em um mapeamento:

- Fontes ativas

- Elementos raiz
- Conexões de porta de destino
- Elementos abstratos
- Pontos de controle de transação
- Colunas FileName

Fontes ativas

Uma fonte ativa é uma transformação que pode retornar um número diferente de linhas para cada linha de entrada. O Serviço de Integração pode carregar dados de diferentes fontes ativas para um destino XML. No entanto, todas as portas de um único grupo do destino XML devem receber dados da mesma fonte ativa.

As seguintes transformações são fontes ativas:

- Agregador
- Qualificador de Origem de Aplicativo
- Personalizada, configurada como uma transformação ativa
- Java, configurada como uma transformação ativa
- Joiner
- Qualificador de Origem MQ
- Normalizador (VSAM ou pipeline)
- Classificação
- Classificador
- Qualificador de Origem
- SQL
- Qualificador de Origem XML
- Mapplet, se o mapplet contiver uma das transformações mencionadas acima

Selecionando um Elemento Raiz

Se uma definição XML tiver mais de uma raiz possível, você poderá especificar um elemento raiz para uma instância de destino.

Para especificar o elemento raiz:

1. Clique com o botão direito do mouse na definição de destino no Mapping Designer e selecione Editar.
2. Clique na guia Propriedades.
3. Clique na seta na coluna de valor Elemento Raiz.
A caixa de diálogo Selecionar Raiz é exibida.
4. Selecione um elemento na lista.

Conectando Portas de Destino

Você precisa conectar corretamente as portas de destino XML em um mapeamento para que o Serviço de Integração possa criar um arquivo XML válido durante uma sessão. Quando você salva ou valida um mapeamento com um destino XML, o Designer valida as conexões de porta de destino.

Use as seguintes diretrizes quando você conectar portas em um mapeamento:

- Se você conectar uma porta em um grupo, conecte as portas das chaves primária e externa do grupo.
- Se você conectar uma porta de chave externa em um grupo, conecte a porta de chave primária associada no outro grupo. Se você não conectar a porta de chave primária do grupo raiz, não será necessário conectar as portas de chave externa associadas nos outros grupos.
- Se você usar um esquema XML com um valor de atributo padrão, conecte a porta de atributo para criar o atributo padrão no destino. Se você transmitir um valor nulo pela porta conectada, o Serviço de Integração gravará o valor padrão no destino.

Conectando Elementos Abstratos

Um elemento abstrato não pode ocorrer diretamente em um arquivo de instância XML. Em vez disso, você deve usar um elemento derivado do elemento abstrato. Por padrão, o Designer cria uma exibição para qualquer elemento complexo abstrato. Para reduzir metadados, os elementos do tipo abstrato não se repetem em nenhum tipo derivado. Ao mapear dados para o tipo abstrato, você também precisa mapear dados para pelo menos um tipo derivado.

Durante uma sessão, se o Serviço de Integração carregar dados para um tipo abstrato, o Serviço de Integração também deverá ter dados para um tipo derivado não abstrato associado ao tipo abstrato. Se o tipo derivado não tiver dados, o Serviço de Integração não gravará o elemento abstrato no arquivo XML de destino.

Libertando Dados XML para Destinos

Você pode liberar dados para um destino XML em cada ponto de confirmação de uma sessão. No entanto, cada grupo de entrada deve receber dados do mesmo ponto de controle de transação no mapeamento. Ao criar uma sessão com base nesse mapeamento, você pode anexar dados ao destino do arquivo XML em cada confirmação ou criar um novo arquivo em cada confirmação. É possível especificar qualquer uma das opções na propriedade de destino Na Confirmação.

Quando você conecta os grupos de entrada de destino XML a vários pontos de controle de transação, o Serviço de Integração grava os dados no destino do arquivo XML depois que processa todas as linhas de origem.

Atribuindo Nomes a Arquivos XML Dinamicamente

Você pode adicionar uma coluna FileName a uma definição de destino XML a fim de criar dinamicamente nomes para arquivos XML. Quando o Serviço de Integração transmite dados à coluna FileName, ele substitui o nome do arquivo de saída nas propriedades de destino. Por exemplo, se você transmitir a string "Harry" para a coluna FileName, o Serviço de Integração atribuirá o nome de Harry ao arquivo XML.

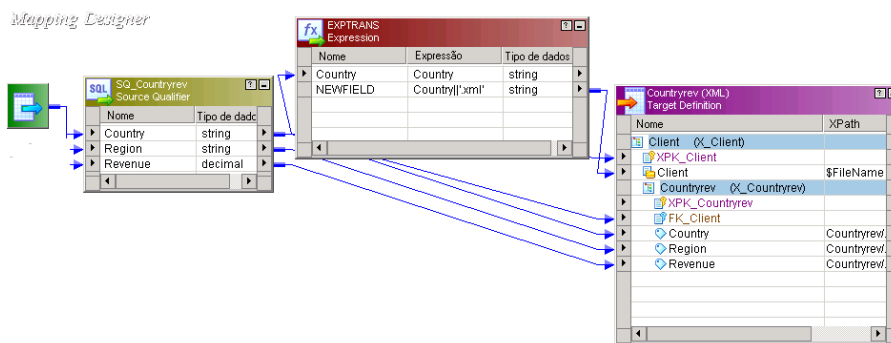
Nota: Se estiver criando um novo arquivo XML em cada confirmação, você precisará atribuir nomes dinamicamente a cada arquivo XML criado. Se você não atribuir nome dinamicamente a cada arquivo XML, o Serviço de Integração substituirá o arquivo XML da confirmação anterior.

O Serviço de Integração gera um novo arquivo XML para cada valor de chave primária distinto no grupo raiz do destino. Você adiciona uma coluna FileName a fim de definir nomes diferentes para cada arquivo. Cada nome substitui o nome do arquivo de saída nas propriedades da sessão.

Exemplo

A transformação de Expressão gera um nome de arquivo a partir do elemento XML Country e transmite o valor para a coluna FileName. O mapeamento transmite um país para a raiz de destino, que é chamada Client. Sempre que o valor Client é alterado, o Serviço de Integração cria um novo arquivo XML. O Serviço de Integração cria um arquivo de lista que contém cada nome de arquivo de destino XML. O Serviço de Integração lista o caminho absoluto para cada arquivo na lista.

A figura a seguir mostra um mapeamento que contém um destino XML com uma coluna FileName:



O Serviço de Integração passa as seguintes linhas para o destino:

```
Country,Region,Revenue
USA,region1,1000
France,region1,10
Canada,region1,100
USA,region2,200
USA,region3,300
USA,region4,400
France,region2,20
France,region3,30
France,region4,40
```

A sessão produz os seguintes arquivos por nome de país:

```
Canada.xml
France.xml
USA.xml
```

O nome do arquivo de lista é o nome do arquivo de saída das propriedades da sessão:

```
revenue_file.xml.lst
```

Solucionando Problemas dos Destinos XML

Importei uma definição de origem de um arquivo XML. Em seguida, importei uma definição de destino do mesmo arquivo XML. Os grupos padrão das definições de origem e de destino não são iguais.

O Assistente de XML nem sempre criará a mesma estrutura de grupos para uma definição de origem e uma definição de destino se você alterar alguma das opções ao importar o destino.

Por exemplo, o elemento ContactInfo no DTD a seguir é um elemento enclosure. O elemento enclosure não tem conteúdo de texto, mas tem maxOccurs > 1. Os elementos filhos também têm maxOccurs > 1.

```
<!ELEMENT HR (EMPLOYEE+)>
<!ELEMENT EMPLOYEE (LASTNAME, FIRSTNAME, ADDRESS+, CONTACTINFO+)>
<!ATTLIST EMPLOYEE EMPID CDATA #REQUIRED>
<!ELEMENT LASTNAME (#PCDATA)>
<!ELEMENT FIRSTNAME (#PCDATA)>
<!ELEMENT ADDRESS (STREETADDRESS, CITY, STATE, ZIP)>
<!ELEMENT STREETADDRESS (#PCDATA)>
<!ELEMENT CITY (#PCDATA)>
<!ELEMENT STATE (#PCDATA)>
<!ELEMENT ZIP (#PCDATA)>
<!ELEMENT CONTACTINFO (PHONE+, EMERGCONTACT+)>
<!ELEMENT PHONE (#PCDATA)>
<!ELEMENT EMERGCONTACT (#PCDATA)>
```

Se você não criar uma exibição XML para os elementos enclosure na definição de origem, não criará o elemento Contactinfo na origem.

A figura a seguir mostra as definições de origem e destino que o Assistente de XML cria:

Nome	Tipo de dados
EMPLOYEE (X_EMPLOYEE)	
XPK_EMPLOYEE	xsd:integer
EMPID	xsd:string
LASTNAME	xsd:string
FIRSTNAME	xsd:string
ADDRESS (X_ADDRESS)	
XPK_ADDRESS	xsd:integer
FK_ADDRESS	xsd:integer
STREETADDRESS	xsd:string
CITY	xsd:string
STATE	xsd:string
ZIP	xsd:string
PHONE (X_PHONE)	
XPK_PHONE	xsd:integer
FK_PHONE	xsd:integer
PHONE	xsd:string
EMERGCONTACT (X_EMERGCONTACT)	
XPK_EMERGCONTACT	xsd:integer
FK_EMERGCONTACT	xsd:integer
EMERGCONTACT	xsd:string

Nome	Tipo de dados
EMPLOYEE (X_EMPLOYEE)	
XPK_EMPLOYEE	xsd:integer
EMPID	xsd:string
LASTNAME	xsd:string
FIRSTNAME	xsd:string
ADDRESS (X_ADDRESS)	
XPK_ADDRESS	xsd:integer
FK_ADDRESS	xsd:integer
STREETADDRESS	xsd:string
CITY	xsd:string
STATE	xsd:string
ZIP	xsd:string
CONTACTINFO (X_CONTACTINFO)	
XPK_CONTACTINFO	xsd:integer
FK_CONTACTINFO	xsd:integer
PHONE (X_PHONE)	
XPK_PHONE	xsd:integer
FK_PHONE	xsd:integer
PHONE	xsd:string
EMERGCONTACT (X_EMERGCONTACT)	
XPK_EMERGCONTACT	xsd:integer
FK_EMERGCONTACT	xsd:integer
EMERGCONTACT	xsd:string

A definição de origem não inclui o elemento ContactInfo. A definição de destino inclui o elemento ContactInfo. O assistente não inclui o elemento ContactInfo na definição de origem porque você optou por não criar exibições para elementos enclosure quando criou a origem. No entanto, o assistente inclui o elemento ContactInfo na definição de destino.

A definição de destino XML que criei com base em minhas origens relacionais contém todos os elementos, mas não atributos. Como posso modificar a hierarquia de destino de forma que seja possível marcar certos dados como atributos?

Você não pode modificar os tipos de componentes que o assistente cria a partir das tabelas relacionais. No entanto, pode exibir um arquivo DTD ou de esquema XML da hierarquia XML de destino. Salve o arquivo DTD ou de esquema XML com um novo nome de arquivo. Abra esse novo arquivo e modifique a hierarquia, definindo os atributos e os elementos. Em seguida, use o arquivo para importar uma definição de destino com uma nova hierarquia.

Transformação de Qualificador de Origem XML

Visão Geral da Transformação de Qualificador de Origem XML

Ao adicionar uma definição de origem XML a um mapeamento, você precisa conectar a definição de origem a uma transformação de Qualificador de Origem XML. A transformação de Qualificador de Origem XML define os elementos de dados que o Serviço de Integração lê durante uma sessão. O qualificador de origem determina como o PowerCenter lê os dados de origem. A transformação de Qualificador de Origem XML é uma transformação ativa.

Você pode adicionar manualmente uma transformação de qualificador de origem ou pode criá-la por padrão ao adicionar uma definição de origem a um mapeamento.

Você pode editar algumas das propriedades e adicionar extensões de metadados a uma transformação de Qualificador de Origem XML.

Ao conectar uma transformação de Qualificador de Origem XML em um mapeamento, você deve seguir as regras para criar um mapeamento válido.

Adicionando um Qualificador de Origem XML a um Mapeamento

Uma transformação de Qualificador de Origem XML tem uma porta de entrada/saída para cada coluna na origem XML. Quando você cria uma transformação de Qualificador de Origem XML para uma definição de origem, o Designer vincula cada porta na definição de origem XML a uma porta na transformação de Qualificador de Origem XML. Não é possível remover ou editar nenhum dos links. Se você remover uma definição de origem XML de um mapeamento, o Designer também removerá a transformação de Qualificador de Origem XML correspondente. Você pode vincular uma definição de origem XML a uma transformação de Qualificador de Origem XML.

Você pode vincular portas de um grupo de Qualificador de Origem XML a portas de diferentes transformações para formar fluxos de dados separados. No entanto, não é possível vincular portas de mais de um grupo em uma transformação de Qualificador de Origem XML a portas na mesma transformação de destino.

Se você arrastar colunas de mais de um grupo para uma transformação, o Designer copiará as colunas de todos os grupos para a transformação. No entanto, o Designer vincula apenas as portas do primeiro grupo às portas correspondentes das novas colunas na transformação.

Você pode adicionar uma transformação de Qualificador de Origem XML a um mapeamento arrastando uma definição de origem XML para o espaço de trabalho do Mapping Designer ou criando manualmente o qualificador de origem.

Criando uma transformação de qualificador de origem XML por padrão

Quando você arrasta uma definição de origem XML para o espaço de trabalho do Mapping Designer, o Designer cria uma transformação de Qualificador de Origem XML por padrão.

Para criar uma transformação de Qualificador de Origem XML por padrão:

1. No Mapping Designer, crie um novo mapeamento ou abra um existente.

2. Arraste uma definição de origem XML para o mapeamento.

O Designer cria uma transformação de Qualificador de Origem XML e vincula cada porta na definição de origem XML a uma porta dessa transformação.

Criando uma Transformação de Qualificador de Origem XML Manualmente

Você poderá criar uma transformação de Qualificador de Origem XML em um mapeamento se tiver um mapeamento que contém definições de origem XML sem Qualificadores de Origem ou se excluir a transformação de Qualificador de Origem XML de um mapeamento.

Para criar uma transformação de Qualificador de Origem XML manualmente:

1. No Mapping Designer, crie um novo mapeamento ou abra um existente.
Nota: Deve haver, pelo menos, uma definição de origem XML sem um qualificador de origem no mapeamento.
2. Clique em Transformação > Criar.
A caixa de diálogo Criar Transformação é exibida.
3. Selecione a transformação de Qualificador de Origem XML e digite um nome para a transformação.
A convenção de nomenclatura para as transformações de Qualificador de Origem XML é *XSQ_TransformationName*.
4. Clique em Criar.
O Designer lista todas as definições de origens XML no mapeamento sem transformações de Qualificador de Origem XML correspondentes.
5. Selecione uma definição de origem e clique em OK.
O Designer cria a transformação de Qualificador de Origem XML no mapeamento e vincula cada porta na definição de origem XML a uma porta dessa transformação.

Editando uma Transformação de Qualificador de Origem XML

Você pode editar propriedades de transformação de Qualificador de Origem XML, como nome e descrição de transformação.

Para editar uma transformação de Qualificador de Origem XML:

1. No Mapping Designer, abra a transformação de Qualificador de Origem XML.
2. Na guia Transformação, edite as propriedades.

A tabela a seguir descreve as propriedades da transformação:

Configuração de Transformação	Descrição
Selecionar Transformação	Exibe a transformação que você está editando. Para escolher outra transformação para edição, selecione a transformação na lista.
Renomear	Edita o nome da transformação.
Descrição	Descreve a transformação.

- Clique na guia Portas para exibir as portas da transformação de Qualificador de Origem XML.
Use a coluna Sequência para definir valores iniciais para chaves geradas em grupos XML. Você pode inserir um valor diferente para cada chave gerada. As chaves de sequência são do tipo de dados bigint. Sempre que você alterar esses valores, os números de sequência reiniciarão a próxima vez que você executar uma sessão.
- Clique na guia Propriedades para configurar as propriedades que afetam como o Serviço de Integração executa o mapeamento durante uma sessão.

A seguinte tabela descreve as propriedades do Qualificador de Origem XML:

Configuração de Propriedades	Descrição
Selecionar Transformação	Exibe a transformação que você está editando. Para escolher outra transformação para edição, selecione a transformação na lista.
Nível de Rastreamento	Determina a quantidade de informações sobre esta transformação que o Serviço de Integração grava no log da sessão quando executa o fluxo de trabalho. Você pode substituir esse nível de rastreamento quando configura uma sessão.
Redefinir	No fim de uma sessão, o Serviço de Integração redefine os valores iniciais para os valores iniciais da sessão atual.
Reiniciar	No início de uma sessão, o Serviço de Integração inicia a sequência de chaves geradas de todos os grupos ao mesmo tempo.

- Clique na guia Extensões de Metadados para criar, editar e excluir extensões de metadados definidas pelo usuário.
Você pode criar, modificar, excluir e promover extensões de metadados não reutilizáveis e atualizar seus valores. Você também pode atualizar os valores de extensões de metadados reutilizáveis.
- Clique em OK.

Configurando Números de Sequência para Chaves Geradas

Cada exibição na definição de Qualificador de Origem XML tem uma chave primária e um valor de sequência para a chave. Durante uma sessão, o Serviço de Integração gera chaves de valores de sequência e incrementa os valores.

No fim da sessão, o Serviço de Integração atualiza cada valor de sequência no repositório para o valor atual mais 1. Esses valores se tornam os valores iniciais a próxima vez que o Serviço de Integração processa a transformação de Gerador de Sequências.

O repositório mantém os seguintes valores de sequências:

- **Valor padrão** O valor de sequência para uma chave que aparece no Qualificador de Origem XML quando você cria o qualificador de origem pela primeira vez. O padrão é 1 para cada chave.
- **Valor inicial.** Um valor de número de sequência para uma chave no início de uma sessão. Você pode exibir os valores iniciais na transformação de Qualificador de Origem XML antes de executar um fluxo de trabalho.
- **Valor atual.** Um valor de sequência de chave durante uma sessão.

Os valores iniciais das chaves geradas são exibidos na coluna Sequência no Qualificador de Origem XML.

Nota: Se editar os valores iniciais da sequência na guia Portas, você deverá salvar as alterações e sair do Designer antes de executar um fluxo de trabalho.

Alterando Valores Iniciais da Sequência

Você pode alterar os valores iniciais da sequência antes ou após uma sessão usando as seguintes opções na guia Propriedades da transformação de Qualificador de Origem XML:

- **Redefinir** No fim de uma sessão, o Serviço de Integração redefine os valores iniciais para os valores iniciais da sessão atual. Por exemplo, no início de uma sessão, o valor inicial de uma chave é 2000. No fim de uma sessão, o valor atual é 2500. Quando a sessão é concluída, o valor inicial no repositório permanece em 2000. Você poderá usar esta opção quando estiver testando e desejar gerar os mesmos números de chave a próxima vez que executar uma sessão.
- **Reiniciar.** No início de uma sessão, o Serviço de Integração reinicia os valores iniciais usando o valor padrão. Por exemplo, se o valor inicial de uma chave for 1005 e você selecionar Reiniciar, o Serviço de Integração alterará o valor inicial para 1. Você poderá usar esta opção se as chaves estiverem ficando grandes e você não terá conflitos de chaves duplicadas se reiniciar a numeração.

Usando o Qualificador de Origem XML em um Mapeamento

Cada grupo em uma definição de origem XML é análogo a uma tabela relacional e o Designer trata cada grupo na transformação de Qualificador de Origem XML como uma origem separada de dados.

O Designer impõe as regras de concatenação quando você conecta objetos em um mapeamento. Portanto, você precisa organizar os grupos na definição de origem XML para que cada grupo contenha todas as informações requeridas em uma ramificação de pipeline.

Considere as seguintes regras quando você conecta uma transformação de Qualificador de Origem XML em um mapeamento:

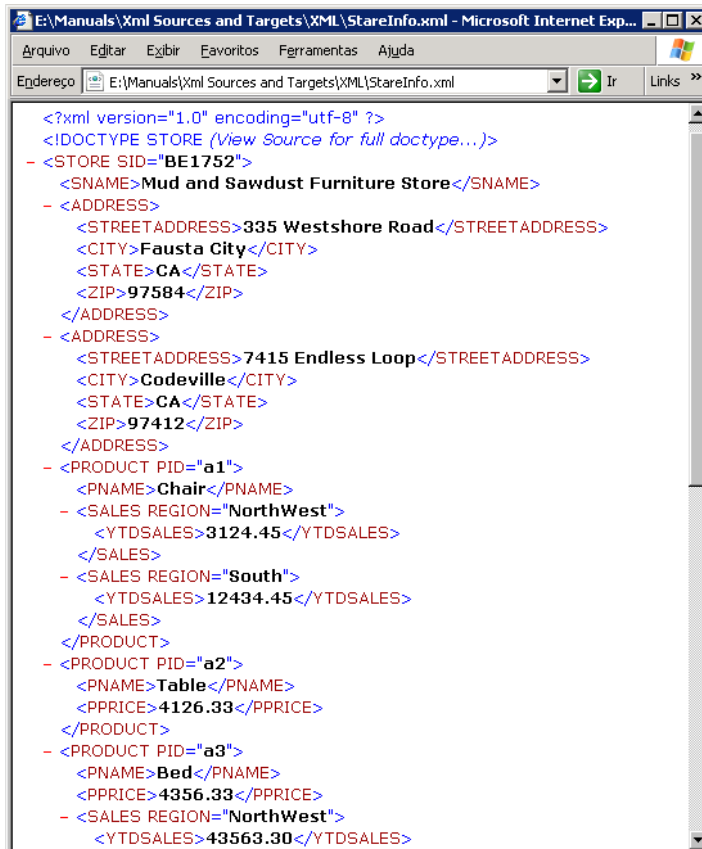
- **Você pode vincular portas de um grupo em uma transformação de Qualificador de Origem XML a portas em um grupo de entrada de outra transformação.** Você pode *copiar* as colunas de vários grupos para uma transformação, mas pode *vincular* as portas de apenas um grupo às portas correspondentes na transformação.

- **Você pode vincular portas de um grupo em uma transformação de Qualificador de Origem XML a portas em mais de uma transformação.** Cada grupo em uma transformação de Qualificador de Origem XML pode ser uma origem de dados para mais de uma ramificação de pipeline. Os dados podem ser transmitidos de um grupo para várias transformações diferentes.
- **Você pode vincular vários grupos de uma transformação de Qualificador de Origem XML a diferentes grupos de entrada em uma transformação.** Você pode vincular vários grupos de uma transformação de Qualificador de Origem XML a diferentes grupos de entrada na maioria das transformações de vários grupos de entrada, como Joiner ou Personalizada. No entanto, você poderá vincular vários grupos de uma transformação de Qualificador de Origem XML a uma transformação de Joiner se a Joiner tiver entrada classificada. Para conectar dois grupos de transformação de Qualificador de Origem XML a uma transformação de Joiner com entrada não classificada, você deve criar duas instâncias da mesma origem XML.

Exemplo de transformação de Qualificador de Origem XML

Esta seção mostra um exemplo de uma transformação de Qualificador de Origem XML em um mapeamento. O exemplo usa um arquivo XML que contém informações de lojas, produtos e vendas.

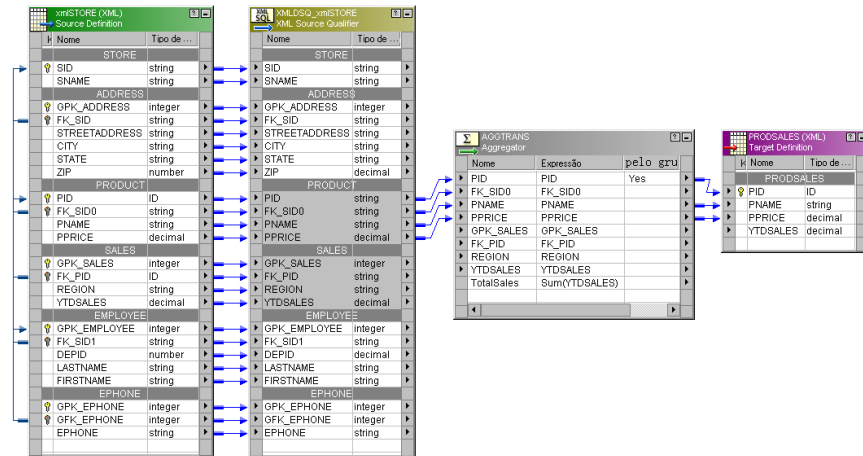
A seguinte figura mostra o arquivo StoreInfo.xml:



Você pode desejar calcular o total das vendas YTD de cada produto no arquivo XML, independentemente da região. Além das vendas, você também deseja obter os nomes e os preços de cada produto.

Para fazer isso, você precisa ter informações sobre vendas e produtos na mesma transformação. No entanto, quando você importa o arquivo StoreInfo.xml, o Designer cria grupos separados para produtos e vendas por padrão.

A seguinte figura mostra os grupos padrão para o arquivo StoreInfo com as informações sobre produtos e vendas em grupos separados:



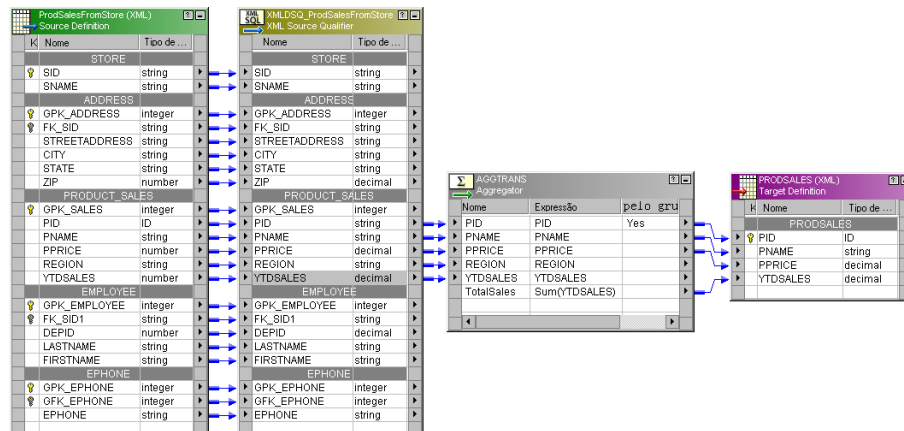
Como não é possível vincular os grupos Product e Sales à mesma transformação de único grupo de entrada, você pode criar o mapeamento de uma das seguintes maneiras:

- Use um grupo desordenado que contenha todas as informações necessárias.
- Una os dados dos dois grupos usando uma transformação de Joiner.

Usando Um Grupo Desordenado

Você pode organizar os grupos na definição de origem de forma que as informações venham do mesmo grupo. Por exemplo, você pode combinar os grupos Product e Sales em um grupo desordenado na definição de origem. Você pode processar todas as informações para a agregação de vendas do grupo desordenado por um fluxo de dados.

A imagem a seguir mostra um grupo desordenado Product_Sales contendo uma combinação de colunas dos grupos Product e Sales:



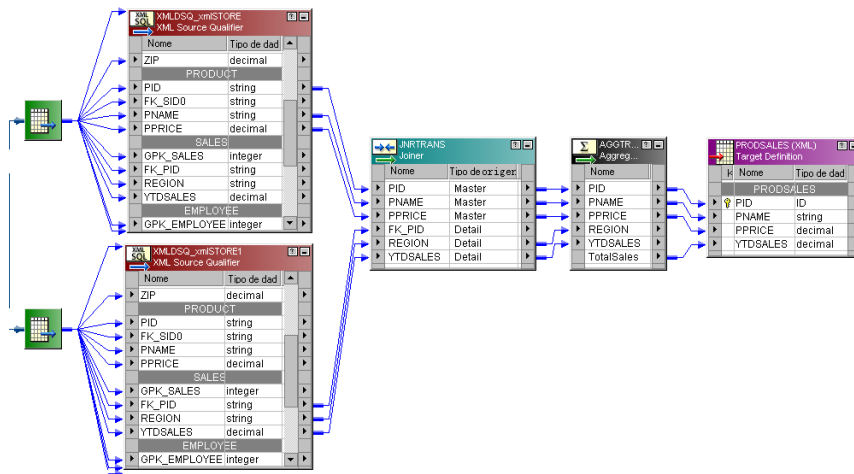
Para criar um grupo desordenado, edite a definição de origem no Source Analyzer. Você pode criar um novo grupo ou modificar um grupo existente. Adicione as colunas produto e vendas ao grupo para fazer o cálculo de vendas na transformação do Agregador. Use o Editor de XML para criar o grupo e validá-lo.

Unindo Dois Grupos de Transformação de Qualificador de Origem XML

Você pode unir dados de dois grupos de origem em um fluxo de dados. Una os dados dos grupos usando uma transformação de Joiner. Ao configurar a transformação de Joiner para a entrada classificada, você pode vincular dois grupos de uma instância da transformação de Qualificador de Origem XML à transformação de Joiner. Ao usar uma transformação de Joiner configurada para a entrada não classificada, você deve usar duas instâncias a mesma origem XML e vincular um grupo de cada instância da transformação de Qualificador de Origem XML à transformação de Joiner.

Em seguida, você pode enviar os dados da transformação de Joiner a uma transformação de Agregador para calcular o YTDSales de cada produto.

A seguinte figura mostra como é possível criar duas instâncias da mesma origem XML e unir dados de duas transformações de Qualificador de Origem XML:



Solucionando Problemas das Transformações de Qualificador de Origem XML

Quando arrasto dois grupos de uma transformação de Qualificador de Origem XML para uma transformação, o Designer copia as colunas, mas não vincula todas as portas.

Você pode vincular um grupo de uma transformação de Qualificador de Origem XML a uma transformação. Quando você arrasta mais de um grupo para uma transformação, o Designer copia todos os nomes de coluna para a transformação. No entanto, o Designer vincula apenas as colunas do primeiro grupo.

Não posso quebrar o vínculo entre a definição de origem XML e seu qualificador de origem.

As colunas da transformação de Qualificador de Origem XML coincidem com as colunas correspondentes da definição de origem XML. Não é possível remover ou modificar os vínculos entre uma definição de origem XML e sua transformação de Qualificador de Origem XML. Quando você

remove uma definição de origem XML, o Designer remove sua transformação de Qualificador de Origem XML.

Transformações XML de Midstream

Visão Geral das Transformações XML de Midstream

As definições XML leem ou criam dados XML. No entanto, às vezes você precisa extrair ou gerar XML dentro de um pipeline. Por exemplo, você pode enviar uma mensagem a um destino TIBCO que contém um documento XML como o campo de dados. Nesse caso, você precisa gerar um documento XML antes de enviar a mensagem ao TIBCO. Use uma transformação XML para gerar o XML.

Você pode criar os seguintes tipos de transformação XML de midstream:

- **Transformação de Analisador de XML.** A transformação de Analisador de XML lê o XML de uma porta de entrada e gera dados para um ou mais grupos.
- **Transformação de Gerador de XML.** A transformação de Gerador de XML lê dados de uma ou mais origens e gera XML. A transformação de Gerador de XML tem uma única porta de saída.

Use uma transformação de XML de midstream para extrair dados XML dos sistemas de mensagens, como TIBCO, WebSphere MQ, ou de outras origens, como arquivos ou bancos de dados. A funcionalidade da transformação XML é similar à funcionalidade de origem e destino XML, exceto que a transformação XML de midstream analisa o XML ou gera o documento no pipeline.

As transformações XML de midstream oferecem suporte aos mesmos componentes de esquema XML que o Assistente de XML e o Editor de XML têm suporte. Além disso, as transformações XML oferecem suporte à seguinte funcionalidade:

- **Portas de passagem.** Use as portas de passagem para transmitir dados que não são XML pela transformação de midstream. Esses campos não fazem parte da definição de esquema XML, mas você os usa para gerar grupos XML desordenados. Você usa esses campos da mesma maneira que os elementos XML de nível superior. Você também pode usar um campo de passagem como uma chave primária para o grupo de nível superior na definição XML.
- **Processamento em tempo real.** Use uma transformação XML de midstream para processar dados como BLOBs de sistemas de mensagens.
- **Suporte para várias partições.** Você pode gerar diferentes documentos XML para cada partição.

Transformação de Analisador de XML

Quando o Serviço de Integração processa uma transformação de Analisador de XML, ele lê uma linha de dados XML, analisa o XML e retorna dados por meio de grupos de saída. A transformação de Analisador de XML retorna dados que não são XML em portas de passagem. Você pode analisar mensagens de XML de origens, como JMS ou IBM WebSphere MQ. A transformação de Analisador de XML é uma transformação ativa.

A transformação de Analisador de XML tem um grupo de entrada e um ou mais grupos de saída. O grupo de entrada tem uma porta de entrada, DataInput, que aceita um documento XML em uma string.

Quando você criar uma transformação de Analisador de XML, use o Assistente de XML para importar um arquivo XML, DTD ou de esquema XML. Por exemplo, você pode importar o seguinte arquivo DTD Employee:

```
<!ELEMENT EMPLOYEES (EMPLOYEE+)>
<!ELEMENT EMPLOYEE (LASTNAME, FIRSTNAME, ADDRESS, PHONE+, EMAIL*, EMPLOYMENT)>
  <!ATTLIST EMPLOYEE EMPID CDATA #REQUIRED
    DEPTID CDATA #REQUIRED>
<!ELEMENT LASTNAME (#PCDATA)>
<!ELEMENT FIRSTNAME (#PCDATA)>
<!ELEMENT ADDRESS (STREETADDRESS, CITY, STATE, ZIP)>
<!ELEMENT STREETADDRESS (#PCDATA)>
<!ELEMENT CITY (#PCDATA)>
<!ELEMENT STATE (#PCDATA)>
<!ELEMENT ZIP (#PCDATA)>
<!ELEMENT PHONE (#PCDATA)>
<!ELEMENT EMAIL (#PCDATA)>
<!ELEMENT EMPLOYMENT (DATEOFHIRE, SALARY+)>
<!ATTLIST EMPLOYMENT EMPLSTAT (PF|PP|TF|TP|O) "PF">
<!ELEMENT DATEOFHIRE (#PCDATA)>
<!ELEMENT SALARY (#PCDATA)>
```

A transformação de Analisador de XML mostra a exibição raiz, X_Employees, com X_Employees mostrada como um pai de X_Employee. X_Employee é o pai de X_Salary, X_Phone e X_Email.

A seguinte figura mostra a transformação de Analisador de XML que o Designer criará se você optar por criar relacionamentos de entidade:

Nome	Tipo de dados	Comprim...
DataInput		
DataInput	string	64000
X_EMPLOYEES		
XPK_EMPLOYEES	integer	19
X_EMPLOYEE		
XPK_EMPLOYEE	integer	19
FK_EMPLOYEES	integer	19
EMPID	string	50
DEPTID	string	50
LASTNAME	string	50
FIRSTNAME	string	50
STREETADDRESS	string	50
CITY	string	50
STATE	string	50
ZIP	string	50
EMPLSTAT	string	2
DATEOFHIRE	string	50
X_SALARY		
XPK_SALARY	integer	19
FK_EMPLOYEE1	integer	19
SALARY	string	50
X_PHONE		
XPK_PHONE	integer	19
FK_EMPLOYEE	integer	19
PHONE	string	50
X_EMAIL		
XPK_EMAIL	integer	19
FK_EMPLOYEE0	integer	19
EMAIL	string	50

O Designer cria uma exibição raiz, X_Employees. X_Employees é o pai de X_Employee. X_Employee é o pai de X_Salary, X_Phone e X_Email.

Cada exibição na transformação de Analisador de XML tem, pelo menos, uma chave para estabelecer seu relacionamento com outra exibição. Se você não designar as chaves no Editor de XML, o Designer criará as chaves primária e externa de cada exibição. As chaves são do tipo de dados bigint. As chaves são chamadas chaves geradas, pois o Serviço de Integração cria os valores de chave sempre que retorna uma linha da transformação de Analisador de XML.

Quando o Designer cria uma coluna de chaves primárias ou externas, ele atribui um nome de coluna com um prefixo. Em uma definição XML, o prefixo são XPK_ para uma coluna de chaves primárias geradas e XFK_ para uma coluna de chaves externas geradas. Uma chave externa sempre se refere a uma chave primária em outro grupo. Uma coluna de chave externa gerada sempre se refere a uma coluna de chave primária gerada.

Por exemplo, o grupo X_Employee tem a chave primária XPK_Employee. O Designer cria colunas de chaves externas que conectam X_Phone, X_Email e X_Salary ao grupo X_Employee. Cada grupo tem a coluna de chave externa XFK_Employee.

O repositório armazena os valores de chaves. Você não pode alterar os valores no repositório, mas pode optar por redefinir ou reiniciar os números de sequência após uma sessão.

Validação de entrada de Analisador de XML

Você pode configurar a transformação de Analisador de XML para validar o XML antes de analisá-lo. A transformação de Analisador de XML valida o XML em relação a um esquema. Se o XML não for válido para o esquema, ocorrerá um erro de linha. A transformação de Analisador de XML retorna o XML e as mensagens de erro associadas a um grupo de saída separado. Você pode transmitir o XML inválido e a mensagem de erro para um destino.

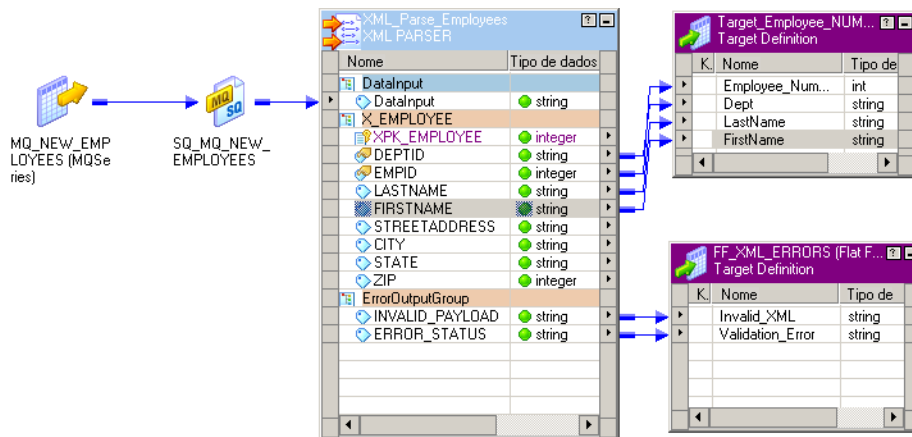
Por exemplo, uma sessão em tempo real do PowerCenter lê mensagens de XML de uma origem do WebSphere MQSeries. A sessão é executada com uma confirmação baseada na origem. Uma mensagem na transação de confirmação tem uma carga de XML inválida. Para evitar a falha na confirmação, você pode configurar a transformação de Analisador de XML para retornar o XML inválido para um grupo de saída separado dos dados válidos. A transformação de Analisador de XML processa as mensagens de XML válidas e conclui a transação.

O log de sessão contém uma mensagem que indica o momento em que a opção Rotear Carga Inválida por meio do Fluxo de Dados está ativada. Quando você define o nível de rastreamento da sessão como Normal, o Serviço de Integração grava uma mensagem no log de sessão que indica se a validação é bem-sucedida. A mensagem de log contém a localização do esquema acessado pelo Analisador de XML para validar o XML. Quando o streaming de XML está ativado e o XML é inválido, o Serviço de Integração trunca o XML e transmite-o para a porta Invalid_Payload. O Serviço de Integração registra o XML inválido no log de sessão.

Para configurar a transformação de Analisador de XML para validar o XML, ative a opção Rotear Carga Inválida por meio do Fluxo de Dados na guia Analisador de XML de Midstream. O Designer adiciona as seguintes portas à transformação de Analisador de XML:

- **Invalid_Payload.** Retorna mensagens de XML inválidas ao pipeline. Se a carga de XML for válida, a porta Invalid_Payload conterá um valor nulo. Essa porta tem a mesma precisão que a porta DataInput.
- **Error_Status.** Contém a strings de erro ou o status retornado da validação de XML. Se o XML for válido para a linha atual, Error_Status conterá um valor nulo. Essa porta tem a mesma precisão que a porta DataInput.

A figura a seguir mostra uma transformação de XML Parser que roteia as mensagens de XML inválidas para uma tabela de destino Erros:



O mapeamento contém os seguintes objetos:

- **Definição de origem MQSeries.** Contém dados XML do funcionário no campo de dados da mensagem.
- **Transformação do Qualificador de Fonte.** Lê dados do WebSphere MQ. Contém um conjunto de portas que representam os campos de cabeçalho da mensagem e o campo de dados da mensagem.
- **Transformação de Analisador de XML.** Recebe os dados da mensagem de XML na porta DataInput. Quando o XML é válido, a transformação de Analisador de XML retorna os dados do funcionário e transmite-os para um destino. Quando o XML não é válido, a transformação de Analisador de XML retorna o XML na porta Invalid_Payload. Ele retorna uma mensagem de erro na porta Error_Status.
- **Definição de destino de funcionários.** Recebe linhas de dados de funcionários válidos.
- **Definição do destino XML_Errors.** Recebe mensagens de XML e de erro inválidas.

Configure o atributo Localização de Esquema XML nas propriedades da sessão para a transformação. Insira o nome e a localização do esquema em relação ao qual validar o XML. Você pode configurar variáveis e parâmetros de fluxo de trabalho, sessão ou mapeamento para a definição de esquema XML. Você poderá configurar vários esquemas para validação se separá-los com ponto-e-vírgulas.

Você poderá usar um DTD para validação se incluí-lo na carga de XML de entrada. Não é possível configurar um DTD no atributo Localização de Esquema XML ou usá-lo para rotear dados XML inválidos para a porta Invalid Payload.

Se você ativar o streaming XML, verifique se a precisão para a porta Invalid_Payload corresponde ao tamanho máximo da mensagem. Se a precisão de porta for menor que o tamanho da mensagem, a transformação de Analisador de XML retornará o XML truncado na porta Invalid_Payload e gravará um erro no log de sessão.

Criar fluxo XML para a transformação de Analisador de XML

Você pode configurar uma sessão para criar fluxo do XML de uma transformação de Dados Não Estruturados, origem JMS ou origem WebSphere MQ para a transformação de Analisador de XML. Quando o Serviço de Integração do PowerCenter faz streaming de dados XML, ele divide esses dados em vários segmentos.

Você pode configurar uma porta de entrada menor na transformação de Analisador de XML e reduzir a quantidade de memória exigida por essa transformação para processar arquivos XML grandes. É possível analisar arquivos XML com mais de 100 MB.

Quando você ativa o streaming XML, a transformação de Analisador de XML recebe dados em segmentos que são menores ou iguais ao tamanho de porta. Quando o arquivo XML é maior que o tamanho da porta, o Serviço de Integração do PowerCenter passa mais de uma linha para a transformação de Analisador de XML. Cada linha XML tem um tipo de linha de streaming. A última linha tem um tipo de linha de inserção.

A precisão da porta de entrada deve ser maior ou igual à precisão da porta de saída do objeto que passa o XML para a transformação do Analisador de XML. Quando a maioria dos documentos XML são pequenos, mas algumas mensagens são grandes, defina a porta de transformação do Analisador de XML para o tamanho das mensagens menores para melhor desempenho.

Se você habilitar o streaming XML, também será necessário ativar o streaming XML para a origem ou transformação que está transmitindo os dados XML para a transformação do Analisador de XML. Se você não ativar o streaming, o Analisador de XML receberá o XML em uma linha, o que poderá tornar o desempenho lento.

Para ativar o streaming XML na transformação de Analisador de XML, selecione Ativar Streaming de Entrada de XML nas propriedades de sessão da transformação de Analisador de XML. Se você ativar o streaming XML na origem ou na transformação, mas não ativá-lo para a transformação de Analisador de XML, essa transformação não poderá processar o arquivo XML.

Quando você habilita o streaming de XML e ocorrer um erro no documento XML, o Serviço de Integração do PowerCenter grava o documento XML para o log de sessão por padrão. Quando ocorre um erro, é possível configurar a sessão para gravar o documento XML no arquivo de log de erros.

Habilite o log de dados de linha de origem nas propriedades da sessão. Quando você habilitar o log e ocorrer um erro no documento XML, o Serviço de Integração do PowerCenter gerará um erro de linha. O Serviço de Integração do PowerCenter grava o documento XML no arquivo de log de erros e incrementa a contagem de erros.

Para obter informações sobre dimensionamento de XML no PowerCenter, consulte [“Visão Geral do Uso do XML com o PowerCenter” na página 33](#). Para obter mais informações sobre as limitações que se aplicam ao tratamento de XML no PowerCenter, consulte [“Limitações” na página 33](#).

Para criar uma transformação com outros tipos de elementos e para transformar arquivos de entrada XML maiores, use uma transformação de Processador de Dados. Para obter mais informações sobre como criar transformações de Processador de Dados, consulte o *Guia do Usuário do Informatica Data Transformation* e o *Guia de Introdução do Informatica Data Transformation*.

Tipo de dados Decimal XML

Quando você define a precisão de um elemento decimal XML para que tenha mais de 34 dígitos, o Serviço de Integração chama uma função externa para converter o tipo de dados decimal XML em Double na transformação de Analisador de XML. A função retorna Double com uma precisão length que depende do nó que está sendo executado no Serviço de Integração. Em todas as plataformas, a precisão tem 17 dígitos antes de o número ser arredondado, mas a precisão pode ter mais em algumas plataformas.

Por exemplo, em Windows de 32 bits, o Serviço de Integração arredonda o número após 17 dígitos.

`1234.567890123456789 is converted to 1234.567890123460800.`

No HP-UX de 32 bits, o Serviço de Integração arredonda o número após 34 dígitos.

Transformação de gerador de XML

Use uma transformação de Gerador de XML para combinar entradas provenientes de várias origens a fim de criar um documento XML. Por exemplo, use a transformação para combinar os dados XML de duas origens TIBCO em um destino TIBCO. Uma origem pode conter informações de funcionários e salários e a outra informações de telefone e email. A transformação de Gerador de XML é uma transformação ativa e conectada.

A transformação de Gerador de XML é semelhante a uma definição de destino XML. Quando o Serviço de Integração processa uma transformação de Gerador de XML, ele grava linhas de dados XML. O Serviço de Integração também pode processar campos de passagem que contenham dados que não sejam XML na transformação.

A transformação de Gerador de XML tem um ou mais grupos de entrada e um grupo de saída. O grupo de saída tem uma porta, "DataOutput", que gera um documento XML BLOB de dados da strings. O grupo de saída contém a porta de passagem quando você cria campos de passagem.

Criando uma Transformação XML de Midstream

Ao criar uma transformação XML midstream, você usa o Assistente de XML e o Editor de XML para definir os grupos de XML. É possível criar a transformação no Transformation Developer e no Mapping Designer.

Para criar uma transformação XML de midstream:

1. Abra o Transformation Developer ou o Mapping Designer.
2. Clique em Transformação > Criar.
A caixa de diálogo Criar Transformação é exibida.
3. Selecione o tipo de transformação Analisador de XML ou Gerador de XML.
4. Insira um nome de transformação e clique em Criar.
A caixa de diálogo Importar Definição XML é exibida.
5. Escolha um arquivo a ser importado e clique em Abrir.
O Assistente de XML é exibido.
6. Crie as definições XML com o Assistente XML.
7. Clique em Concluir no Assistente de XML.
A transformação XML midstream aparece no espaço de trabalho.
8. Para editar as propriedades da transformação XML midstream, clique duas vezes na transformação no espaço de trabalho.

Sincronizando uma definição XML de midstream

Você pode sincronizar uma transformação XML de midstream usando uma versão diferente do esquema ou do arquivo de origem que você importou para criar a transformação. Por exemplo, você pode adicionar novos elementos ao arquivo de esquema importado para criar uma transformação de Analisador de XML. Você pode atualizar a transformação de Analisador de XML com o novo esquema, em vez de excluir a transformação e recriá-la.

Para sincronizar uma transformação XML de midstream, use o Transformation Developer ou o Mapping Designer.

Para sincronizar uma transformação XML de midstream:

1. Abra o Transformation Developer ou o Mapping Developer.
2. Arraste a transformação XML de midstream ou o mapeamento que deseja atualizar para o espaço de trabalho.
3. Clique com o botão direito na parte superior da transformação XML de midstream.
4. Selecione a transformação de Sincronizar XML.
A caixa de diálogo Importar Definição XML é exibida.
5. Navegue para a definição de repositório ou o arquivo que você usou para criar a transformação de Analisador de XML ou Gerador de XML.
6. Clique em Abrir para atualizar a transformação.

O Designer não pode sincronizar uma transformação com um arquivo que você não usou para criar a transformação.

Você usa o Source Analyzer e o Target Designer para sincronizar as definições XML de origem e de destino.

Editando propriedades de transformação XML midstream

Você pode editar algumas das propriedades de transformação XML de midstream. No entanto, como você usa o Assistente de XML e o Editor de XML para definir a transformação, deverá usar essas ferramentas para alterar a definição XML.

Quando você cria uma transformação XML de midstream no Mapping Designer, as seguintes regras se aplicam:

- Se tornar a transformação reutilizável, você poderá alterar algumas das propriedades de transformação no Mapping Designer. Não é possível adicionar portas de passagem ou extensões de metadados.
- Se criar uma transformação não reutilizável, você poderá editar a transformação no Mapping Designer.

Ao configurar uma transformação XML de midstream, você pode configurar componentes nas seguintes guias:

- **Guia Transformação.** Renomeie a transformação e adicione uma descrição na guia Transformação.
- **Guia Portas.** Exibe as portas e os atributos de transformação que você cria na guia Analisador de XML ou Gerador de XML.

- **Guia Propriedades.** Atualize o nível de rastreamento.
- **Guia Propriedades de Inicialização.** Cria propriedades em tempo de execução que um procedimento externo usa durante a inicialização.
- **Guia Extensões de Metadados.** Estende os metadados armazenados no repositório associando as informações aos objetos do repositório, como uma transformação XML.
- **Guia Definições de Atributo de Porta.** Define os atributos de porta que se aplicam a todas as portas na transformação.
- **Guia Analisador de XML ou Gerador de XML de Midstream.** Crie portas de passagem usando essa guia. As portas de passagem permitem transmitir dados que não são XML pela transformação. Para a transformação de Analisador de XML, você poderá optar por redefinir números da sequência se você usar a numeração de sequência para gerar nomes de coluna XML. Para a transformação de Gerador de XML, você pode optar por criar um novo documento XML nas confirmações.

Guia Propriedades

Configure as propriedades de transformação de XML de midstream na guia Propriedades.

A seguinte tabela descreve as opções que você pode alterar na guia Propriedades:

Transformação	Descrição
Local do Tempo de Execução	<p>Local que contém a DLL ou a biblioteca compartilhada. O padrão é \$PMExtProcDir. Insira um caminho relativo ao nó do Serviço de Integração que executa a sessão XML.</p> <p>Se esta propriedade estiver em branco, o Serviço de Integração usará a variável de ambiente definida no nó do Serviço de Integração para localizar a DLL ou a biblioteca compartilhada.</p> <p>Copie todas as DLLs ou bibliotecas compartilhadas para o local do tempo de execução ou para a variável de ambiente definida no nó do Serviço de Integração. O Serviço de Integração não carrega o procedimento quando não consegue localizar a DLL, a biblioteca compartilhada ou um arquivo especificado.</p>
Nível de rastreamento	Quantidade de detalhes exibidos no log de sessão dessa transformação. O padrão é Normal.
Escopo de Transformação	<p>Indica como o Serviço de Integração aplica a lógica de transformação aos dados de entrada. Você pode optar por um dos seguintes valores do escopo de transformação para a transformação de Analisador de XML:</p> <ul style="list-style-type: none"> - Linha. Aplica a lógica de transformação a um linha de dados de cada vez. Libera as linhas geradas para todos os grupos de saída antes de processar a próxima linha. - Transação. Aplica a lógica de transformação a todas as linhas em uma transação. Libera as linhas geradas nos limites da transação, quando os blocos de saída são completados e no fim do arquivo. - Todas as Entradas. Aplica a lógica de transformação em todos os dados de entrada. Libera as linhas geradas somente quando os blocos de saída são completados e no fim do arquivo. <p>Para a transformação do Gerador de XML, o Designer define o escopo de transformação para todas as entradas quando você define a configuração de Na Confirmação como Ignorar Confirmação. O Designer definirá o escopo de transformação como o nível de transação se você definir Na Confirmação para Criar Novo Documento.</p>

Transformação	Descrição
Saída é repetível	Indica se a ordem dos dados de saída é consistente entre as execuções de sessão. <ul style="list-style-type: none"> - Nunca. A ordem dos dados de saída é inconsistente entre as execuções de sessão. - Baseada na Ordem de Entrada. A ordem de saída é consistente entre as execuções de sessão quando a ordem dos dados de entrada é consistente entre elas. - Sempre. A ordem dos dados de saída será consistente entre as execuções de sessão mesmo se a ordem de entrada for inconsistente entre elas. O padrão é Baseada na Ordem de Entrada para a transformação de Analisador de XML. O padrão é Sempre para a transformação de Gerador de XML.
Requer Thread Único por Partição	Indica se o Serviço de Integração deve processar cada partição com um thread.
Saída é determinística	Indica se a transformação gera os mesmos dados de saída entre as execuções de sessão. Você deve ativar essa propriedade para executar a recuperação em sessões que usam essa transformação. O padrão é ativado.

Aviso: Se você configurar uma transformação como repetível e determinística, será sua responsabilidade garantir que os dados sejam repetíveis e determinísticos. Se você tentar recuperar uma sessão com transformações que não gerem os mesmos dados entre a sessão e a recuperação, o processo de recuperação poderá resultar em dados corrompidos.

Guia Analisador de XML Midstream

Use a guia Analisador de XML de Midstream para modificar o tamanho da porta DataInput. Também é possível adicionar portas de passagem a essa guia.

Você pode acessar o Editor de XML na guia Analisador de XML de Midstream. Clique no botão Editor de XML.

Nota: Depois que acessar o Editor de XML, você não poderá atualizar Editar Transformações enquanto não sair do Editor de XML.

A seguinte tabela descreve as opções que você pode alterar na guia Analisador de XML de Midstream:

Transformação	Descrição
Precisão	Comprimento da coluna. A precisão padrão da porta DataInput é 64K. A precisão padrão de uma porta de passagem é 20. É possível aumentar a precisão.
Reiniciar	Sempre inicie a sequência de chaves gerada em 1. Sempre que você executa uma sessão, os valores da sequência de chaves em todos os grupos da definição XML começam em 1.
Redefinir	No fim de uma sessão, redefina a sequência de valores de todas as chaves geradas em todos os grupos. Redefina os números da sequência para os valores originais antes da sessão.
Rotear Carga Inválida por meio do Fluxo de Dados	Valide o XML em relação a um esquema. Se o XML não for válido para o esquema, ocorrerá um erro de linha. A transformação de Analisador de XML retorna o XML e as mensagens de erro associadas a um grupo de saída separado.
Descrição	Descreve a transformação.

Nota: Se você não selecionar Redefinir ou Reiniciar, os números de sequência nas chaves geradas aumentarão de sessão para sessão. Se selecionar a opção Reiniciar ou Redefinir, você atualizará a propriedade Reiniciar ou Redefinir que aparece na guia Propriedades de Inicialização. No entanto, você não poderá alterar essas opções na guia Propriedades de Inicialização.

Guia Gerador XML de Midstream

Use a guia Gerador de XML para modificar o tamanho da porta DataOutput. Também é possível adicionar portas de passagem a essa guia.

É possível acessar o Editor XML da Guia Gerador XML de Midstream. Clique no botão Editor de XML. Depois de acessar o Editor de XML, você não poderá editar propriedades de transformação enquanto não sair do Editor de XML.

A seguinte tabela descreve as opções que você pode alterar na guia de transformação do Gerador de XML:

Configuração de transformação	Descrição
Precisão	Comprimento da coluna. A precisão padrão da porta DataOutput é 64K. A precisão padrão de uma porta de passagem é 20. É possível aumentar a precisão.
Na Confirmação	<p>O Serviço de Integração pode gerar vários documentos XML após uma confirmação. Use uma das seguintes opções:</p> <ul style="list-style-type: none">- Ignorar Confirmação. O Serviço de Integração cria o documento XML e grava dados no documento no fim do arquivo. Use essa opção se duas origens diferentes estiverem conectadas à transformação do Gerador de XML.- Criar Novo Documento. Cria um novo documento XML em cada confirmação. Use esta opção se você estiver executando uma sessão em tempo real. <p>Quando uma sessão usa várias partições, o Serviço de Integração gera um documento XML separado para cada partição, independentemente das configurações de Na Confirmação. Se você selecionar Criar Novo Documento, o Serviço de Integração criará novos documentos para cada partição.</p>
Descrição	Descreve a transformação.

Nota: O Designer define o escopo de transformação para todas as entradas quando você define a configuração de Na Confirmação como Ignorar Confirmação. O Designer definirá o escopo de transformação como o nível de transação se você definir Na Confirmação para Criar Novo Documento.

Gerando portas de passagem

Portas de passagem são colunas que transmitem dados que não são XML por meio de uma transformação XML de midstream. Por exemplo, você pode transmitir IDs de mensagens com XML para origens e destinos MQSeries. Use o ID de mensagem para correlacionar mensagens de entrada e saída referentes a solicitações e respostas.

Ao definir uma porta de passagem na transformação de midstream, você adiciona a porta de passagem ao grupo DataInput na transformação de Analisador de XML ou ao grupo DataOutput na transformação de Gerador de XML.

Depois de gerar a porta, você usa o Editor de XML para adicionar uma porta de referência correspondente à outra exibição na definição XML. Na transformação de Analisador de XML, a porta de passagem é uma porta de entrada e a porta de referência correspondente é uma porta de saída. Na transformação de Gerador de XML, a porta de passagem é uma porta de saída e a porta de referência associada é uma porta de entrada.

Para criar uma porta de passagem em uma transformação do gerador XML:

1. Abra a transformação no Transformation Developer ou no Mapping Designer.
2. Clique duas vezes na transformação para abrir Editar Transformações.
3. Clique na guia Gerador de XML de Midstream ou Analisador de XML de Midstream.
A porta DataInput ou DataOutput é exibida, dependendo do tipo de transformação.
4. Clique no botão Adicionar para adicionar uma porta de saída para a passagem.
Um campo padrão é exibido na coluna Nome de Campo.
5. Modifique o nome do campo. Você também pode modificar o tipo, a precisão e a escala, dependendo do arquivo usado para criar a definição.
6. Clique em Editor de XML para abrir a definição XML para a transformação.
As exibições XML da definição são exibidas no espaço de trabalho.
7. Clique com o botão direito do mouse na parte superior de uma exibição para adicionar a porta de referência.
8. Selecione Adicionar uma Porta de Referência.
A caixa de diálogo Porta de Referência é aberta.
A caixa de diálogo lista as portas de passagem adicionadas à transformação.
9. Selecione a porta de passagem que corresponderá à nova porta de referência na exibição e clique em OK.
A porta de referência de saída correspondente aparece na exibição. Você pode renomear a porta para um nome significativo na janela Colunas.
10. Clique em Aplicar Alterações e saia do Editor de XML.
11. Clique em OK na transformação.

Os dados que não são XML aparecem na porta de entrada chamada Pass_thru_field e são transmitidos pela porta de saída de referência COL_0 correspondente.

Solucionando Problemas das Transformações XML de Midstream

Preciso extrair arquivos XML de uma tabela de banco de dados que contém um CLOB de XML. Cada arquivo XML pode ter até 2 GB. Se eu criar uma transformação de Analisador de XML, precisarei definir um comprimento máximo fixo para a coluna CLOB. No entanto, o comprimento máximo para o tipo de dados CLOB é 104 MB.

Os dados XML são muito grandes para serem transmitidos diretamente da tabela para uma transformação de Analisador de XML. Você precisa preparar os dados da tabela CLOB para um arquivo simples e cria uma definição de origem XML a partir do arquivo.

Para obter informações sobre dimensionamento de XML no PowerCenter, consulte [“Visão Geral do Uso do XML com o PowerCenter” na página 33](#). Para obter mais informações sobre as limitações que se aplicam ao tratamento de XML no PowerCenter, consulte [“Limitações” na página 33](#).

Para criar uma transformação com outros tipos de elementos e para transformar arquivos de entrada XML maiores, use uma transformação de Processador de Dados. Para obter mais informações sobre como criar transformações de Processador de Dados, consulte o *Guia do Usuário do Informatica Data Transformation* e o *Guia de Introdução do Informatica Data Transformation*.

Referência de Tipo de Dados XML

Tipos de dados XML e de transformação

O PowerCenter oferece suporte a todos os tipos de dados XML especificados na Recomendação W3C de 2 de maio de 2001. A tabela a seguir lista os tipos de dados XML e os compara aos tipos de dados de transformação na transformação de Qualificador de Origem XML. Para obter mais informações sobre os tipos de dados XML, consulte as especificações W3C para tipos de dados XML em <http://www.w3.org/TR/xmlschema-2>.

Você poderá alterar os tipos de dados em definições XML e transformações XML de midstream se importar um arquivo XML para criar a definição. Você não poderá alterar tipos de dados XML quando importá-los de um esquema XML. Não é possível alterar os tipos de dados de transformação para origens XML em um mapeamento.

A seguinte tabela descreve os tipos de dados XML e os tipos de dados de transformação correspondentes:

Tipo de dados	Transformação	Intervalo
anySimpleType	String	1 a 104.857.600 caracteres
anyURI	String	1 a 104.857.600 caracteres
base64Binary	Binário	1 a 104.857.600 bytes
booleano	Número inteiro curto	Precisão 5; escala 0
byte	Número inteiro curto	Precisão 5; escala 0
data	Data/Hora	1 jan, 0001 D.C. a 31 dez, 9999 D.C. (precisão até o nanossegundo)
dateTime	Data/Hora	1 jan, 0001 D.C. a 31 dez, 9999 D.C. (precisão até o nanossegundo)
decimal	Decimal	Precisão 1 a 28; escala 0 a 28
duplo	Duplo	Precisão 15, escala 0
duração	String	1 a 104.857.600 caracteres

Tipo de dados	Transformação	Intervalo
ENTIDADES	String	1 a 104.857.600 caracteres
ENTIDADE	String	1 a 104.857.600 caracteres
float	Duplo	Precisão 15, escala 0
gDay	Número inteiro	-2.147.483.648 a 2.147.483.647 Precisão 10; escala 0
gMonth	Número inteiro	-2.147.483.648 a 2.147.483.647 Precisão 10; escala 0
gMonthDay	Data/Hora	1 jan, 0001 D.C. a 31 dez, 9999 D.C. (precisão até o nanossegundo)
gYear	Número inteiro	Precisão 10; escala 0
gYearMonth	Data/Hora	1 jan, 0001 D.C. a 31 dez, 9999 D.C. (precisão até o nanossegundo)
hexBinary	Binário	1 a 104.857.600 bytes
ID	String	1 a 104.857.600 caracteres
IDREF	String	1 a 104.857.600 caracteres
IDREFS	String	1 a 104.857.600 caracteres
int	Bigint	-9.223.372.036.854.775.808 a 9.223.372.036.854.775.807 Precisão 19; escala 0
inteiro	Bigint	-9.223.372.036.854.775.808 a 9.223.372.036.854.775.807 Precisão 19; escala 0
idioma	String	1 a 104.857.600 caracteres
longo	Bigint	-9.223.372.036.854.775.808 a 9.223.372.036.854.775.807 Precisão 19; escala 0
Nome	String	1 a 104.857.600 caracteres
Ncname	String	1 a 104.857.600 caracteres
negativeInteger	Bigint	-9.223.372.036.854.775.808 a 9.223.372.036.854.775.807 Precisão 19; escala 0
NMTOKEN	String	1 a 104.857.600 caracteres
NMTOKENS	String	1 a 104.857.600 caracteres
nonNegativeInteger	Bigint	-9.223.372.036.854.775.808 a 9.223.372.036.854.775.807 Precisão 19; escala 0

Tipo de dados	Transformação	Intervalo
nonPositiveInteger	Bigint	-9.223.372.036.854.775.808 a 9.223.372.036.854.775.807 Precisão 19; escala 0
normalizedString	String	1 a 104.857.600 caracteres
NOTAÇÃO	String	1 a 104.857.600 caracteres
positiveInteger	Bigint	-9.223.372.036.854.775.808 a 9.223.372.036.854.775.807 Precisão 19; escala 0
QName	String	1 a 104.857.600 caracteres
curto	Número inteiro curto	Precisão 5; escala 0
string	String	1 a 104.857.600 caracteres
hora	Data/Hora	1 jan, 0001 D.C. a 31 dez, 9999 D.C. (precisão até o nanossegundo)
token	String	1 a 104.857.600 caracteres
unsignedByte	Número inteiro curto	Precisão 5; escala 0
unsignedInt	Bigint	-9.223.372.036.854.775.808 a 9.223.372.036.854.775.807 Precisão 19; escala 0
unsignedLong	Bigint	-9.223.372.036.854.775.808 a 9.223.372.036.854.775.807 Precisão 19; escala 0
unsignedShort	Número inteiro	-2.147.483.648 a 2.147.483.647 Precisão 10; escala 0

Formato de data XML

O PowerCenter oferece suporte ao seguinte formato para os tipos de dados date, time e datetime:

CCYY-MM-DDThh:mm:ss:sss

Use este formato ou qualquer parte dele em um arquivo XML. O PowerCenter não oferece suporte a datas negativas para o formato de data e hora.

Use um elemento date, time ou datetime em qualquer um dos seguintes formatos em uma sessão:

CCYY-MM

- ou -

CCYY-MM-DD/Thh

O formato do primeiro elemento datetime em um arquivo XML determina o formato de todos os valores subsequentes do elemento. O Serviço de Integração rejeitará a linha se ele ler um valor para o mesmo elemento date, time ou datetime que tenha um formato diferente.

Por exemplo, se o primeiro valor de um elemento datetime estiver no seguinte formato:

CCYY-MM-DDThh:mm:ss

O Serviço de Integração rejeitará uma linha que contenha o elemento no seguinte formato:

CCYY-MM-DD

O analisador de XML converte o valor de data e hora na entrada XML para um valor no fuso horário local da máquina que hospeda o Serviço de Integração. Se você habilitar a opção para ajustar o relógio para alterações de horário de verão no Windows, o Analisador de XML adiciona uma hora para o valor de data e hora. Para conversões de valor de data e hora consistentes, não habilitar a opção no Windows para ajustar o relógio para alterações de horário de verão.

Referência de Funções de Consulta do XPath

Visão Geral das Funções de Consulta do XPath

XPath é uma linguagem que descreve uma maneira de localizar itens em um documento XML. O XPath usa uma sintaxe de endereçamento baseada no caminho de passagem pela hierarquia XML de um componente raiz. Você pode criar um predicado de consulta do XPath para elementos na linha de exibição ou uma coluna com um XPath que inclua a linha de exibição.

Use um predicado de consulta do XPath em uma exibição XML para filtrar dados de origem XML. Em uma sessão, o Serviço de Integração extrai dados de um arquivo XML de origem baseado na consulta. Se todas as consultas retornarem TRUE, o Serviço de Integração extrairá dados da exibição.

Um predicado de consulta do XPath inclui um elemento ou atributo a ser extraído e a consulta que determina os critérios. Você pode verificar o valor de um elemento ou atributo ou pode verificar se um elemento ou atributo existe nos dados XML de origem.

Este apêndice descreve cada função usada em um predicado de consulta do XPath. As funções aceitam argumentos e valores retornados. Ao criar uma função, você pode incluir componentes com base nos elementos e atributos contidos na exibição XML e pode adicionar valores literais. Quando você especificar um literal, coloque-o entre aspas simples ou duplas.

Referência Rápida de Função

Use os seguintes tipos de função em um predicado de consulta do XPath:

- **String.** Use as funções string para testar valores de substring, concatenar strings ou converter strings em outras strings. Por exemplo, o seguinte predicado de consulta do XPath determina se o nome completo de um funcionário é igual à concatenação de sobrenome e nome:

```
EMPLOYEE[./FULLNAME=concat(./ENAME/LASTNAME,./ENAME/FIRSTNAME) ]
```

- **Numérico** Use funções numéricas com valores de elemento e de atributo. As funções numéricas operam em números e retornam números inteiros. Por exemplo, o seguinte predicado de consulta do XPath arredonda desconto e testa se o resultado é maior que 15:

```
ORDER_ITEMS[round(./DISCOUNT) > 15]
```

- **Booleana.** Use as funções Booleanas para testar elementos, verificar o atributo de linguagem ou forçar um resultado true ou false. Por exemplo, o seguinte predicado de consulta do XPath retornará true se o valor for maior do que zero:

```
boolean(string)
```

A seguinte tabela descreve as funções string do predicado de consulta do XPath:

Função	Sintaxe	Descrição
concat	concat (string1, string2)	Concatena duas strings.
contains	contains (string, substring)	Determina se uma string contém outra string.
normalize-space	normalize-space (string)	Distribui espaço em branco à esquerda e à direita de uma string.
starts-with	starts-with (string, substring)	Determina se string1 começa com string2.
string	string (value)	Converte um número ou um Booleano em string.
string-length	string-length (string)	Retorna o número de caracteres em uma string, incluindo espaços em branco à direita.
subsequência	substring (string, start [,length])	Retorna uma parte de uma strings que inicia em uma posição especificada.
subsequência-depois	substring-after (string, substring)	Retorna uma parte de uma strings que inicia em uma posição especificada.
substring-before	substring-before (string, substring)	Retorna os caracteres em uma string que ocorre antes de uma substring.
translate	translate (string1, string2, string3)	Converte os caracteres de uma string em outros caracteres.

A tabela seguinte descreve as funções de número do predicado de consulta do XPath:

Função	Sintaxe	Descrição
ceiling	ceiling (number)	Arredonda um número para o menor número inteiro que seja maior ou igual ao número transmitido.
floor	floor (número)	Arredonda um número para o maior número inteiro que seja menor ou igual ao número transmitido.
number	number (value)	Converte uma string ou um valor Booleano em um número.
round	round (número)	Arredonda um número para o inteiro mais próximo.

A tabela seguinte descreve as funções Boolean do predicado de consulta do XPath:

Função	Sintaxe	Descrição
boolean	boolean (object)	Converte um objeto em Booleano.
false	false ()	Sempre retorna FALSE.
lang	lang (code)	Determina se um elemento tem o atributo xml:lang que corresponde ao argumento de código.

Função	Sintaxe	Descrição
not	not (condition)	Retornará TRUE se uma condição Boolean for FALSE e retornará FALSE se a condição Boolean for TRUE.
true	true ()	Sempre retorna TRUE.

boolean

Converte um valor em Booleano.

Sintaxe

```
boolean ( object )
```

A tabela a seguir descreve o argumento boolean:

Argumento	Descrição
<i>Objeto</i>	Tipo de dados numérico ou de string de caractere. Transmite um número ou string a ser testado.

Valor de Retorno

Boolean.

A função retorna um Booleano da seguinte forma:

- Uma string retornará TRUE se seu tamanho for diferente de zero. Do contrário, retornará FALSE.
- Um número retornará FALSE se for zero ou não for um número (NaN). Do contrário, retornará TRUE.

Exemplos

O seguinte exemplo verifica se um nome tem caracteres:

```
boolean ( NAME )
```

A tabela a seguir inclui exemplos de argumentos e valores de retorno:

NAME	RETURN VALUE
Lilah	TRUE
-	FALSE

O seguinte exemplo verifica se um CEP é numérico:

```
boolean ( ZIP_CODE )
```

A tabela a seguir inclui exemplos de argumentos e valores de retorno:

ZIP_CODE	RETURN VALUE
94061	TRUE
94005	TRUE

ZIP_CODE

9400g

RETURN VALUE

FALSE

ceiling

Arredonda um número para o menor número inteiro que seja maior ou igual ao número transmitido.

Sintaxe

```
ceiling ( number )
```

A tabela a seguir descreve o argumento para esta função:

Argumento	Descrição
<i>número</i>	Valor numérico. O número que você deseja arredondar.

Valor de Retorno

Número inteiro.

Exemplo

A seguinte expressão retorna o preço arredondado para o menor número inteiro:

```
ceiling ( PRICE )
```

A tabela a seguir contém exemplos de argumentos e valores de retorno:

PRICE	RETURN VALUE
39.79	40
125.12	126
74.24	75
NULL	NULL
-100.99	-100
100	100

concat

Concatena duas strings.

Sintaxe

```
concat ( string1, string2 )
```

A tabela a seguir descreve os argumentos para esta função:

Argumento	Descrição
<i>string1</i>	Tipo de dados de cadeia. Transmite a primeira cadeia a ser concatenada.
<i>string2</i>	Tipo de dados de cadeia. Transmite a segunda cadeia a ser concatenada.

Valor de Retorno

Cadeia.

Quando uma das cadeias de caractere é NULL, a função Concat ignora-a e retorna a outra cadeia.

Exemplo

A seguinte expressão concatena FIRSTNAME e LASTNAME:

```
concat( FIRSTNAME, LASTNAME )
```

A tabela a seguir inclui exemplos de argumentos e valores de retorno:

FIRSTNAME	LASTNAME	RETURN VALUE
John	Baer	JohnBaer
NULL	Campbell	Campbell
Greg	NULL	Greg
NULL	NULL	NULL

Dica

A função concat não adiciona espaços em cadeias. Para adicionar um espaço entre duas cadeias, você pode gravar uma expressão que contenha funções concat aninhadas. Por exemplo, a seguinte expressão adiciona um espaço ao final do nome e concatena o nome ao sobrenome:

```
concat ( concat ( FIRST_NAME, " " ), LAST_NAME )
```

A tabela a seguir mostra exemplos de argumentos e valores de retorno:

FIRST_NAME	LAST_NAME	RETURN VALUE
John	Baer	John Baer
NULL	Campbell	Campbell (includes leading space)
Greg	NULL	Greg

contains

Determina se uma string contém outra string.

Sintaxe

```
contains( string, substring )
```

A tabela a seguir descreve os argumentos para esta função:

Argumento	Descrição
<i>string</i>	Tipo de dados String. Transmite a string a ser examinada. O argumento faz distinção entre maiúsculas e minúsculas.
<i>substring</i>	Tipo de dados String. Transmite a string a ser pesquisada na string. O argumento faz distinção entre maiúsculas e minúsculas.

Valor de Retorno

Boolean.

Exemplo

As expressões a seguir retornam TRUE se NAME contiver SHORTNAME:

```
contains( NAME, SHORTNAME )
```

A tabela a seguir inclui exemplos de argumentos e valores de retorno:

NAME	SHORTNAME	RETURN VALUE
John	Baer	FALSE
SuzyQ	Suzy	TRUE
WorldPeace	World	TRUE
CASE_SENSITIVE	case	FALSE

false

Sempre retorna FALSE. Use essa função para definir um Booleano como FALSE.

Sintaxe

```
false ()
```

A função false não aceita argumentos.

Valor de Retorno

FALSE.

Exemplo

Combine a função false com outras funções para forçar um resultado FALSE.

A tabela a seguir inclui expressões que retornam FALSE:

EXPRESSION	RETURN VALUE
(salary) = false()	FALSE
A/B = false ()	FALSE

EXPRESSION

```
starts-with ( name, 'T' ) = false()
```

RETURN VALUE

FALSE

floor

Arredonda um número para o maior número inteiro que seja menor ou igual ao número transmitido.

Sintaxe

```
floor( number )
```

A tabela a seguir descreve argumentos para esta função:

Argumento	Descrição
<i>número</i>	Valor numérico. Use uma expressão numérica.

Valor de Retorno

Número inteiro.

NULL se um valor transmitido para a função for NULL.

Exemplo

A seguinte expressão retorna o maior número inteiro menor ou igual ao valor em BANK_BALANCE:

```
floor( BANK_BALANCE )
```

A tabela a seguir contém exemplos de argumentos e valores de retorno:

BANK_BALANCE	RETURN VALUE
39.79	39
NULL	NULL
-100.99	-101
5	5

lang

Retorna TRUE se o elemento tem um atributo xml:lang com a mesma linguagem que o argumento de código. Use a função lang para selecionar o XML por linguagem. O atributo xml:lang é um código que identifica a linguagem do conteúdo do elemento. Um elemento pode incluir texto em várias linguagens.

Sintaxe

```
lang ( code )
```

A tabela a seguir descreve argumentos para esta função:

Argumento	Descrição
<i>código</i>	Tipo de dados String. Transmite o código de linguagem de conteúdo do elemento.

Valor de Retorno

Boolean.

Exemplo

A expressão seguinte examina o código de linguagem de conteúdo do elemento:

```
lang ( âenâ )
```

A tabela a seguir contém exemplos de argumentos e valores de retorno:

XML	RETURN VALUE
<pre><Phrase xml:lang=âesâ> El perro esta en la casa. </Phrase></pre>	FALSE
<pre><Phrase xml:lang=âenâ> The dog is in the house. </Phrase></pre>	TRUE

normalize-space

Remove espaço em branco à esquerda e à direita de uma string. O espaço em branco contém caracteres que não são exibidos, como o caractere de espaço e o caractere de tabulação. Essa função substitui sequências de espaço em branco com um único espaço.

Sintaxe

```
normalize-space ( string )
```

A tabela a seguir descreve o argumento para esta função:

Argumento	Descrição
<i>string</i>	Tipo de dados strings. Transmite uma string que contém espaço em branco.

Valor de Retorno

String.

NULL se a string for NULL.

Exemplo

A seguinte expressão remove os espaços em branco em excesso de um nome:

```
normalize-space ( NAME )
```

A tabela a seguir contém exemplos de argumentos e valores de retorno:

NAME	RETURN VALUE
Jack Dog	Jack Dog
Harry Cat	Harry Cat

not

Retorna o inverso de uma condição Boolean. A função retornará TRUE se uma condição for falsa e retornará FALSE se uma condição for verdadeira.

Sintaxe

```
not ( condition )
```

A tabela a seguir descreve o argumento para esta função:

Argumento	Descrição
<i>condição</i>	Expressão ou valor Boolean.

Valor de Retorno

Boolean.

NULL se a condição for NULL.

Exemplo

A seguinte expressão retorna o inverso de uma condição Boolean:

```
not ( EMPLOYEE = concat ( FIRSTNAME, LASTNAME ) )
```

A tabela a seguir contém exemplos de argumentos e valores de retorno:

EMPLOYEE	FIRSTNAME	LASTNAME	RETURN
Fullname	Full	Name	FALSE
Lastname	Lastname	First	TRUE

number

Converte uma string ou um valor Booleano em um número.

Sintaxe

```
number ( value )
```

A tabela a seguir descreve o argumento para esta função:

Argumento	Descrição
<i>valor</i>	Use um valor Boolean ou string.

Valor de Retorno

A função retorna um número para os seguintes dados:

- Uma string será convertida em um número se a string contiver um caractere numérico. A string pode conter espaço em branco e incluir um sinal de menos seguido de um número. O espaço em branco pode vir depois do número na string. Qualquer outra string é NaN (Não É um Número).
- Um Booleano TRUE é convertido em 1. Um Booleano FALSE é convertido em 0.

Se um valor transmitido como um argumento para a função não for um número, a função retornará NaN.

Exemplo

A seguinte expressão converte pagamento em um número:

```
number ( PAYMENT )
```

A tabela a seguir contém exemplos de argumentos e valores de retorno:

PAYMENT	RETURN VALUE
850.00	850.00
TRUE	1
FALSE	0
AB	NaN

round

Arredonda um número para o inteiro mais próximo. Se o número estiver entre dois inteiros, a função round retornará o número mais alto.

Sintaxe

```
round ( number )
```

A tabela a seguir descreve o argumento para esta função:

Argumento	Descrição
<i>número</i>	Valor numérico. Transmite um tipo de dados numérico ou uma expressão que resulta em um número.

Valor de Retorno

Número inteiro.

Exemplo

A seguinte expressão arredonda BANK_BALANCE:

```
round( BANK_BALANCE )
```

A tabela a seguir contém exemplos de argumentos e valores de retorno:

BANK_BALANCE	RETURN VALUE
12.34	12
12.50	13
-18.99	-19
NULL	NULL

starts-with

Retorna TRUE se a primeira string começa com a segunda. Do contrário, retorna FALSE.

Sintaxe

```
starts-with ( string, substring )
```

A tabela a seguir descreve os argumentos para esta função:

Argumento	Descrição
<i>string</i>	Tipo de dados String. Transmite a string a ser pesquisada. A string faz distinção entre maiúsculas e minúsculas.
<i>substring</i>	Tipo de dados String. Transmite a substring a ser pesquisada na string. A substring faz distinção entre maiúsculas e minúsculas.

Valor de Retorno

Boolean.

Exemplo

A seguinte expressão determinará se NAME começará com FIRSTNAME:

```
starts-with ( NAME, FIRSTNAME )
```

A tabela a seguir contém exemplos de argumentos e valores de retorno:

NAME	FIRSTNAME	RETURN VALUE
Kathy Russell	Kathy	TRUE
Joe Abril	Mark	FALSE

string

Converte um número ou um Booleano em string.

Sintaxe

```
string ( value )
```

A tabela a seguir descreve o argumento para esta função:

Argumento	Descrição
<i>valor</i>	Valor numérico ou Booleano. Transmite um valor numérico ou Booleano.

Valor de Retorno

String.

Retorna uma string vazia se nenhum valor é transmitido. Retorna NULL se um valor nulo é transmitido.

A função string converte um número em uma string da seguinte forma:

- Se o número for um inteiro, a função retornará uma string em formato decimal, sem vírgula decimal e sem zeros à esquerda.
- Se o número não for um inteiro, a função retornará uma string, incluindo uma vírgula decimal com, pelo menos, um dígito antes dela e um dígito após.
- Se o número for negativo, a função retornará uma string com um sinal de menos (-).

A função string converte um Booleano para uma string da seguinte forma:

- Se o Boolean for FALSE, a função retornará a string "false".
- Se o Boolean for TRUE, a função retornará a string "true".

Exemplo

A seguinte expressão retorna uma string da velocidade de argumento numérico:

```
string( SPEED )
```

A tabela a seguir contém exemplos de argumentos e valores de retorno:

SPEED	RETURN VALUE
10.99	'10.99'
15.62567	'15.62567'
0	'0'
10	'10'
50	'50'
1.3	'1.3'

A seguinte expressão retorna uma string do argumento Booleano STATUS:

```
string( STATUS )
```

A tabela a seguir mostra exemplos de argumentos e valores de retorno:

STATUS	RETURN VALUE
TRUE	'true'
FALSE	'false'
NULL	NULL

string-length

Retorna o número de caracteres em uma string, incluindo espaços em branco à direita.

Sintaxe

```
string-length ( string )
```

A tabela a seguir descreve o argumento para esta função:

Argumento	Descrição
<i>string</i>	Tipo de dados String. A string que você deseja avaliar.

Valor de Retorno

Número inteiro.

NULL se um valor transmitido para a função for NULL.

Exemplo

A seguinte expressão retorna o tamanho do nome do cliente:

```
string-length ( CUSTOMER_NAME )
```

A tabela a seguir contém exemplos de argumentos e valores de retorno:

CUSTOMER_NAME	RETURN VALUE
Bernice Davis	13
NULL	NULL
John Baer	9

subsequência

Retorna uma parte de uma strings que inicia em uma posição especificada. A subsequência inclui espaços em branco como caracteres na strings.

Sintaxe

```
substring ( string, start [ ,length ] )
```

A tabela a seguir descreve argumentos para esta função:

Argumento	Descrição
<i>string</i>	Tipo de dados String. A string a ser pesquisada.
<i>início</i>	Tipo de dados Inteiro. Transmite a posição na string para iniciar a contagem. Se a posição inicial for um número positivo, a subsequência localizará a posição inicial contando a partir do início da string. O primeiro caractere é um. Se a posição inicial for um número negativo, a substring localizará a posição inicial contando a partir do final da string.
<i>tamanho</i>	Tipo de dados Inteiro. Deve ser maior do que zero. Transmite o número de caracteres a serem retornados em uma string. Se você omitir o argumento tamanho, a subsequência retornará todos os caracteres da posição inicial até o final da string.

Valor de Retorno

String.

Quando a strings contiver um valor numérico, a função retornará uma strings.

Se você transmitir um inteiro negativo ou zero, a função retornará uma strings vazia.

NULL se um valor transmitido para a função for NULL.

Exemplos

A seguinte expressão retorna o código de área em PHONE:

```
substring( PHONE, 1, 3 )
```

PHONE	RETURN VALUE
809-555-3915	809
NULL	NULL

A seguinte expressão retorna PHONE sem o código de área:

```
substring ( phone, 5, 8 )
```

A tabela a seguir inclui exemplos de argumentos e valores de retorno sem os códigos de área:

PHONE	RETURN VALUE
808-555-0269	555-0269
NULL	NULL

Você pode transmitir um valor inicial negativo para começar do lado direito da strings. A expressão lê a strings de origem da esquerda para a direita do argumento *length*:

```
substring ( PHONE, -8, 3 )
```

A tabela a seguir inclui exemplos de argumentos e valores de retorno quando a expressão lê a string de origem da esquerda para a direita:

PHONE	RETURN VALUE
808-555-0269	555
809-555-3915	555
357-687-6708	687
NULL	NULL

Quando o argumento *length* for maior que a strings, a subsequência retornará todos os caracteres da posição inicial até o fim da strings. Por exemplo:

```
substring ( 'abcd', 2, 8 )
```

retorna "bcd."

```
substring ( 'abcd', -2, 8 )  
returns âcd.â
```

subsequência-depois

Retorna os caracteres em uma strings que ocorre após uma subsequência.

Sintaxe

```
substring-after ( string, substring )
```

A tabela a seguir descreve os argumentos para esta função:

Argumento	Descrição
<i>string</i>	Tipo de dados String. Transmite a string a ser pesquisada.
<i>subsequência</i>	Tipo de dados String. Transmite uma subsequência a ser pesquisada na strings.

Valor de Retorno

String.

Strings vazia se a subsequência não for encontrada.

NULL se um valor transmitido para a função for NULL.

Exemplo

A seguinte expressão retorna a string em PHONE que ocorre após o código de área (415):

```
substring-after ( PHONE, (415) )
```

A tabela a seguir inclui exemplos de argumentos e valores de retorno:

PHONE	RETURN VALUE
(415) 555-1212	555-1212

PHONE	RETURN VALUE
(408) 368-4017	-
NULL	NULL
(415) 366-7621	366-7621

substring-before

Retorna a parte de uma string que ocorre antes de uma substring.

Sintaxe

```
substring-before ( string, substring )
```

A tabela a seguir descreve os argumentos para esta função:

Argumento	Descrição
<i>string</i>	Tipo de dados String. Transmite a string a ser pesquisada.
<i>substring</i>	Tipo de dados String. Transmite a substring a ser pesquisada na string.

Valor de Retorno

String.

Strings vazia se a subsequência não for encontrada.

NULL se um valor transmitido para a função for NULL.

Exemplo

A seguinte expressão retorna o número que ocorre em um endereço Third Street:

```
substring-before ( ADDRESS, Third Street )
```

A tabela a seguir contém exemplos de atributos e valores de retorno:

ADDRESS	RETURN VALUE
100 Third Street	100
250 Third Street	250
600 Third Street	600
NULL	NULL

translate

Converte os caracteres de uma string em outros caracteres. A função usa duas outras strings como pares de tradução.

Sintaxe

```
translate ( string1, string2, string3 )
```

A tabela a seguir descreve os argumentos para esta função:

Argumento	Descrição
<i>string1</i>	Tipo de dados String. Transmite a string a ser convertida.
<i>string2</i>	Tipo de dados String. Transmite a string que define quais caracteres converter. Translate substitui cada caractere em <i>string1</i> por um número que indica sua posição em <i>string2</i> .
<i>string3</i>	Tipo de dados String. Transmite a string que define para quais caracteres da <i>string1</i> codificada deve ser feita a conversão. Translate substitui cada caractere da <i>string1</i> criptografada por um caractere da <i>string3</i> no número de posição da <i>string2</i> .

Valor de Retorno

String.

Exemplo

A seguinte expressão traduz uma string usando duas outras strings:

```
translate ( EXPRESSION, STRING2, STRING3 )
```

A tabela a seguir contém exemplos de argumentos e valores de retorno:

EXPRESSION	STRING2	STRING3	RETURN VALUE
A Space Odissei	i	y	A Space Odyssey
rats	tras	TCAS	CATS
bar	abc	ABC	BAr

Translate não alterará um caractere em EXPRESSION se o caractere não ocorrer em string2. Se um caractere ocorrer em EXPRESSION e string2, mas não ocorrer em string3, o caractere não ocorrerá na string retornada.

true

Sempre retorna TRUE. Use essa função para definir um Booleano como TRUE.

Sintaxe

```
true ()
```

A função true não aceita argumentos.

Valor de Retorno

Booleano TRUE.

Exemplo

A tabela a seguir contém exemplos de expressões que retornam TRUE:

EXPRESSION	RETURN VALUE
(decision) = true ()	TRUE
A/B = true ()	TRUE
(starts-with (name, 'T'))= true	TRUE