Advanced Technical Skills (ATS) North America

Using IMS to Manage XML Documents

Rosemary Shay, DBA, American Express

William Li IBM Software Group, Information Management
IMS XML DB, XQuery
wili@us.ibm.com

Kenny Blackman, zSeries Software IMS Advanced Technical Skills, IBM
kblackm@us.ibm.com

Important Disclaimer

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. THE INFORMATION ON NEW PRODUCTS IS FOR INFORMATIONAL PURPOSES ONLY AND MAY NOT BE INCORPORATED INTO ANY CONTRACT. THE INFORMATION ON ANY NEW PRODUCTS IN NOT A COMMITMENT, PROMISE, OR LEGAL OBLIGATION TO DELIVER ANY MATERIAL, CODE OR FUNCTIONALITY. THE DEVELOPMENT, RELEASE, AND TIMING OF ANY FEATURES OR FUNCTIONALITY DESCRIBED FOR OUR PRODUCTS REMAINS AT THE SOLE DISCRETION OF IBM. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS AND/OR SOFTWARE.

Why IMS?

- IMS has been used in industries for at least 4 decades giving IMS a reputation of a reliable, stable, proven technology
- IMS contains large amounts of core business data
- IMS makes use of direct pointers which attributes to high performance
- New and wow is out - we have to be smarter to support our companies in this economy
- IMS can now grow past 40 terabytes (HALDB)
- IMS can now be accessed via X-Query-SQL (XML)
- IMS can be accessed from Java on the web

Some Access Models

- LU2 - 3270 Green Screen
  - Left behind in favor of distributed clients accessing IMS services.
- LU6.2 - APPC/SNA
  - Left behind since TCP/IP is the de-facto standard network protocol.
- IMS-trigger monitor - BMP schedules IMS tran
  - there are Sysplex issues, same RACF ID used for all transactions triggered by the BMP, programs must code explicit MQ put/get calls.
- MQ-IMS Bridge - MQ to OTMA
  - MQ-IMS Bridge allows distributed and mainframe clients to access application services in IMS. Requires special MQ IIH header but **no coding changes to existing programs** – communication is handled via the I/O PCB
- IMS TM Resource Adapter - IMS Connect to OTMA
  - IMS TM Resource Adpater allows distributed and mainframe Java clients to access application services in IMS. Supports “Local Host” no TCP/IP stack if the Java client is running on the same LPAR.
- Stored Proc - ODBA (Open Database Access)
  - Calling to an IMS database from a stored proc

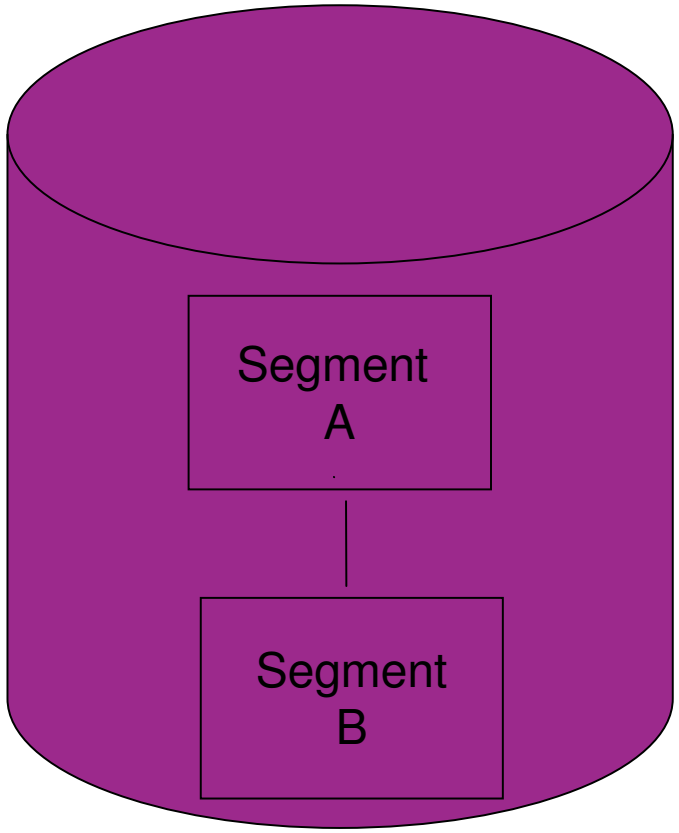
Access Model using Java - XML - SQL

- We now have JBP (Java Batch Processing) and JMP (Java Message Processing) programs.
- We use the IMS/TM to schedule and run JBP and JMP programs.
- The coding language is Java.
- The query language to access the IMS data is X-Query (SQL).
- Data can be stored as an XML document.
- Results can be returned as an XML document or data fields.

IBM's XML Solution

- XML and IMS databases are both hierarchical, thus IMS is a natural DBMS for managing XML documents.

```
<A>
    <f1>xxxx</f1>
    <f2>xxxx</f2>

    <B>
        <f3>xxxx</f3>
        <f4>xxxx</f4>
    </B>
</A>
```

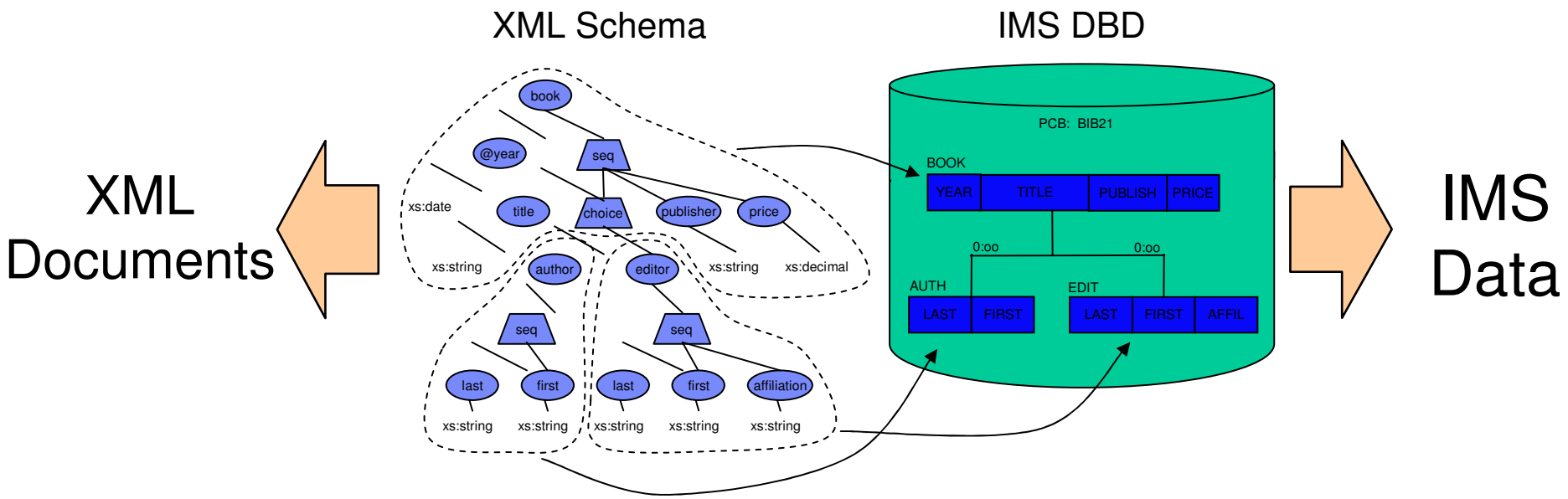


IBM's XML Solution in IMS V9

- Store incoming XML documents
  - Use IMS Java JDBC user defined function
    - StoreXML UDF
  - Can store in an IMS database as:
    - Decomposed – normal IMS fields
    - Intact – Tags and all as one field
    - Combination of the above
- Retrieve XML document or data
  - Use IMS Java JDBC user defined function
    - RetrieveXML UDF
  - Use DLI – regular DLI calls
- Compose XML documents from existing IMS databases

IMS v9 XML-DB

- Introduces a way to view/map IMS hierarchical data to *native* XML documents
- Aligns IMS Database (DBD) with XML Schema
- Allows the retrieval and storage of IMS Records as XML documents with ***no change*** to existing IMS databases



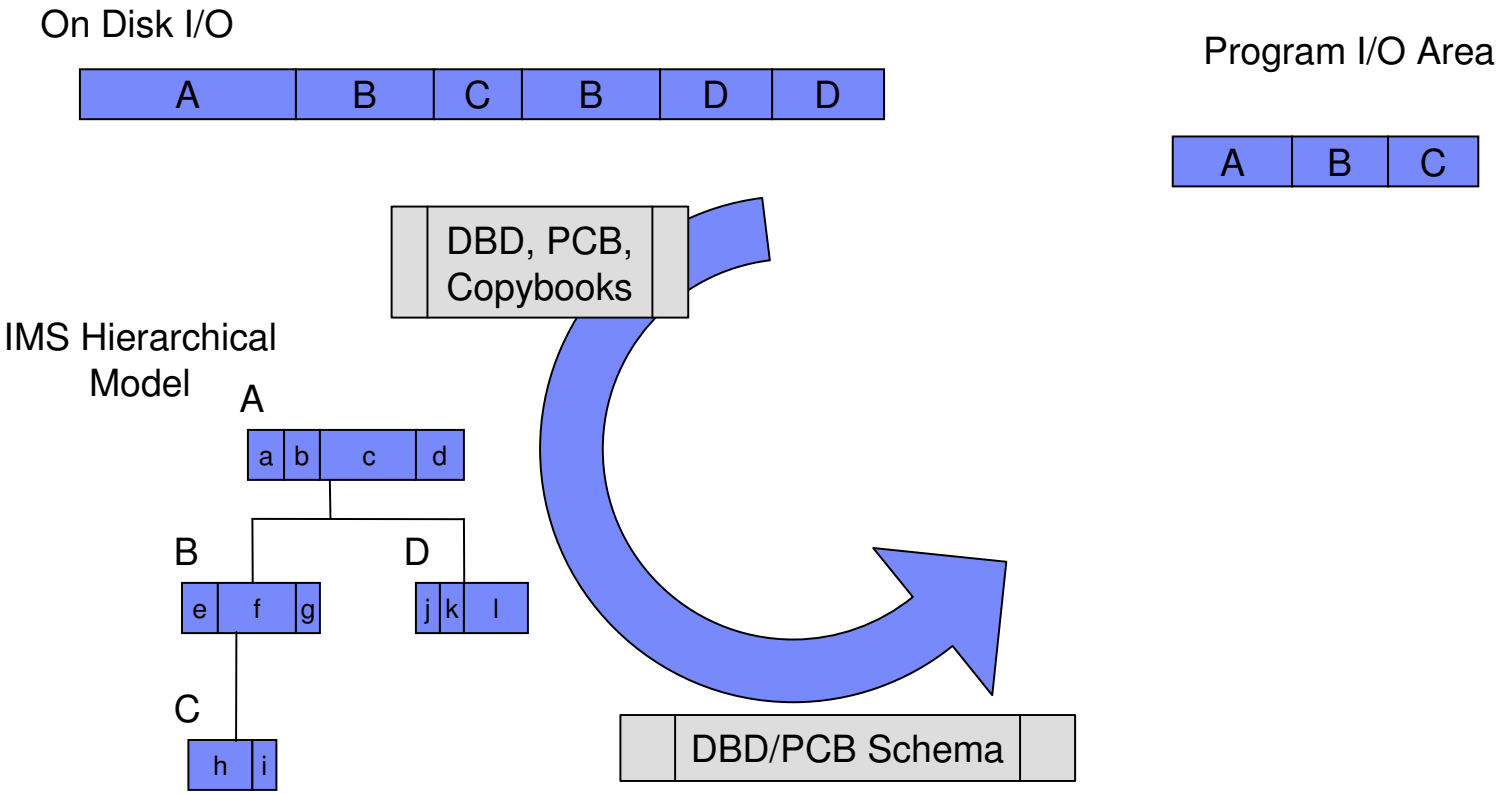
IBM's XML Solution in IMS V10

- IMS Version 10 allows you to use XQuery (W3C).
  - Per **Wikipedia** - **XQuery** is a query language (with some programming language features) that is designed to query collections of XML data. It is semantically similar to SQL.
  - Accessing IMS data is no longer a training issue – it is cutting edge!
- Further aligns IMS with industry direction
  - XML, SOA, Web Services, etc.
- Enables customers to leverage emerging standard skill set
- You can do:
  - Selects with where clauses
  - Updates
  - Write bad SQL just like in DB2

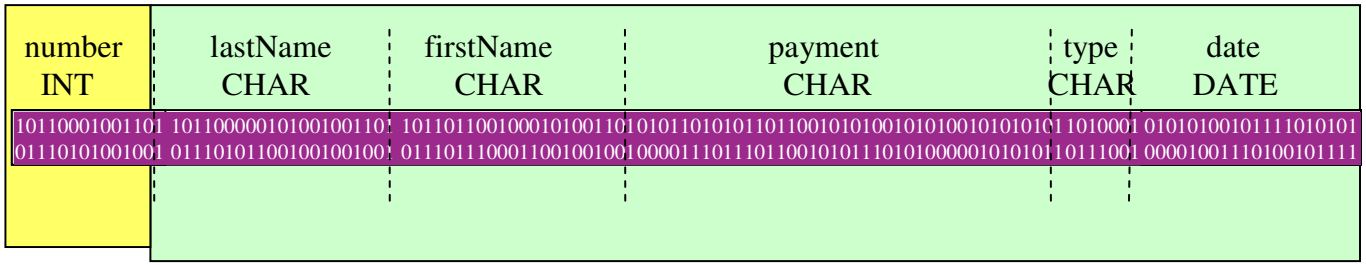
Schema Definition Language

- An IMS Schema:
  - defines database hierarchy
  - defines segments and fields in a database
  - defines data types for key fields
- An XML Schema:
  - defines document hierarchy
  - defines data types for elements and attributes
  - defines elements and attributes that can appear in a document

DBD + PSB = IMS DB Schema



It's the Metadata




```
<PurchaseOrder number="113246 ">
   <lastName> IBM          </lastName>
   <firstName> IMS          </firstName>
   <date>04/26/1968 </date>
   <payment type="MC ">5414 2263 4895 1145 </payment>
```

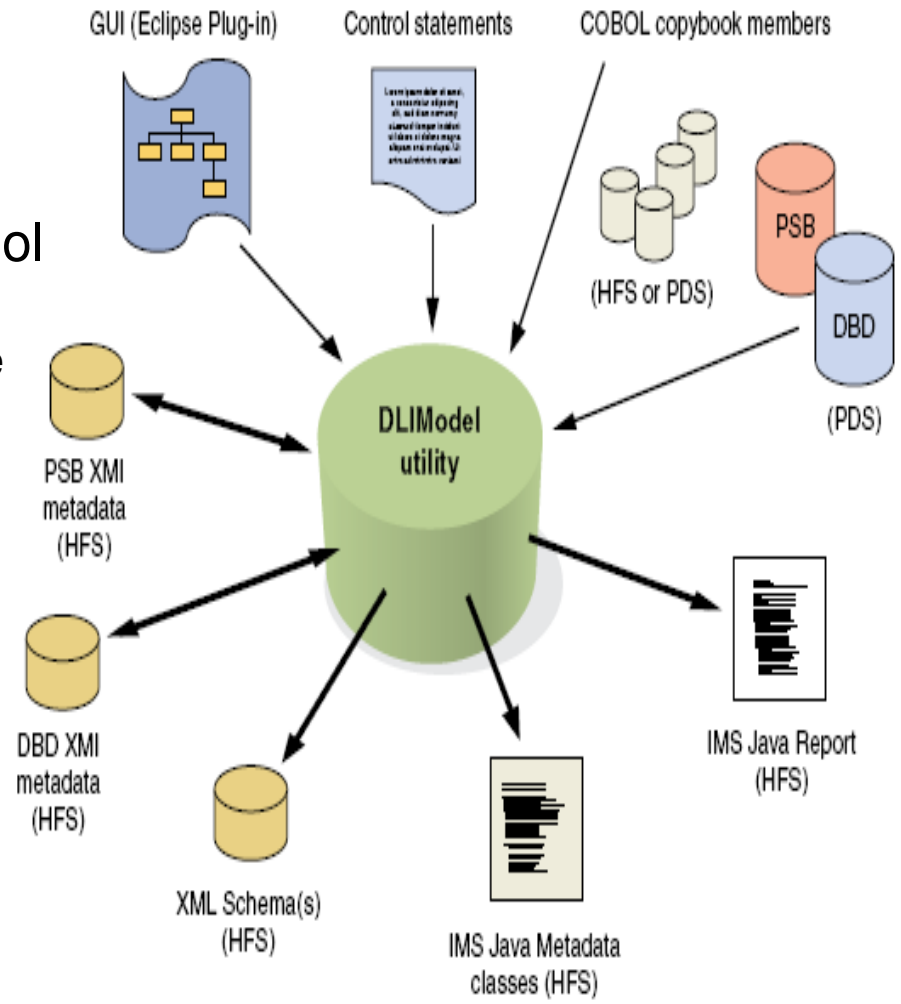
Defined in DBD

Defined in Copylibs (IMS Java)

Defined in XML Schema

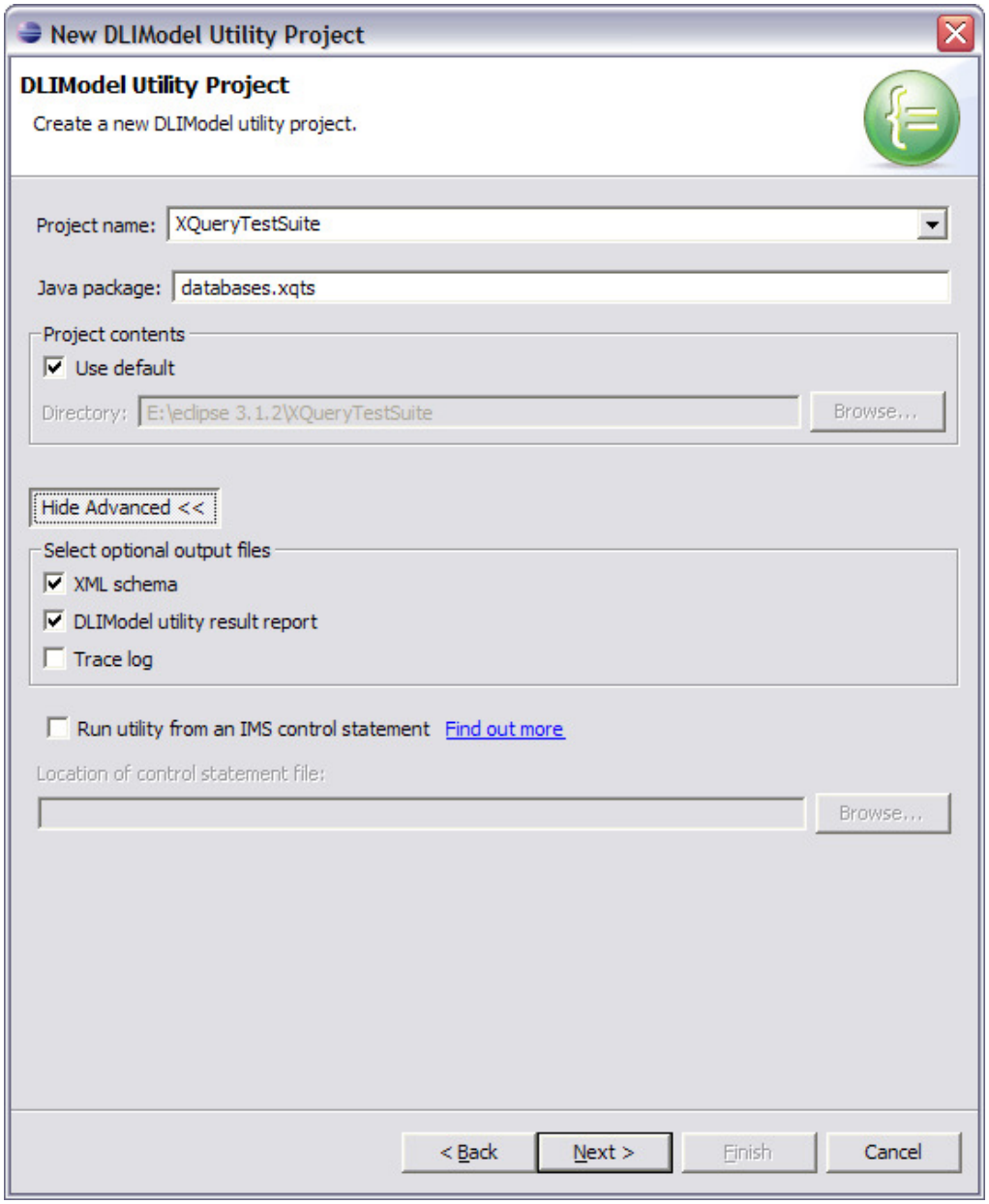
IMS Enterprise Suite DLIModel Utility

- IMS database visualization tool
  - Visualize an entire IMS PSB
  - Can view each PCB individually
    - Hierarchy, segments, fields, types, etc
- IMS database metadata generation tool
  - Generates the necessary metadata that is consumed at runtime by IMS DB Resource Adapter, XML-DB support
    - Database metadata
    - XML schema
- Bottom up tooling approach
  - Parses PSB and DBD source
  - Optionally COBOL copybook definitions of segments
- An Eclipse 3.x plug-in



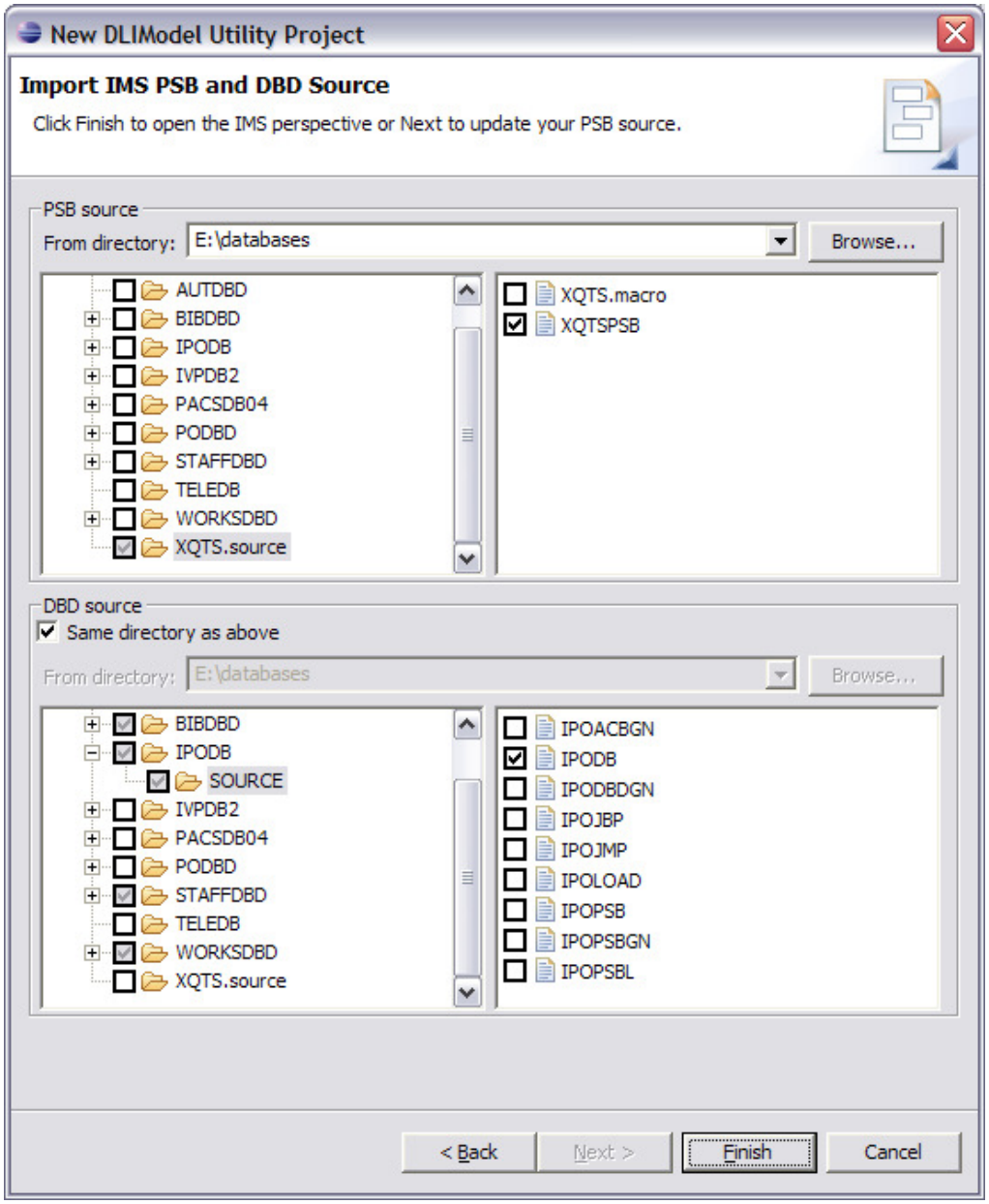
GUI DL/I Model Utility

- Install eclipse Plug-in
- Create new DL/I Utility Model Project

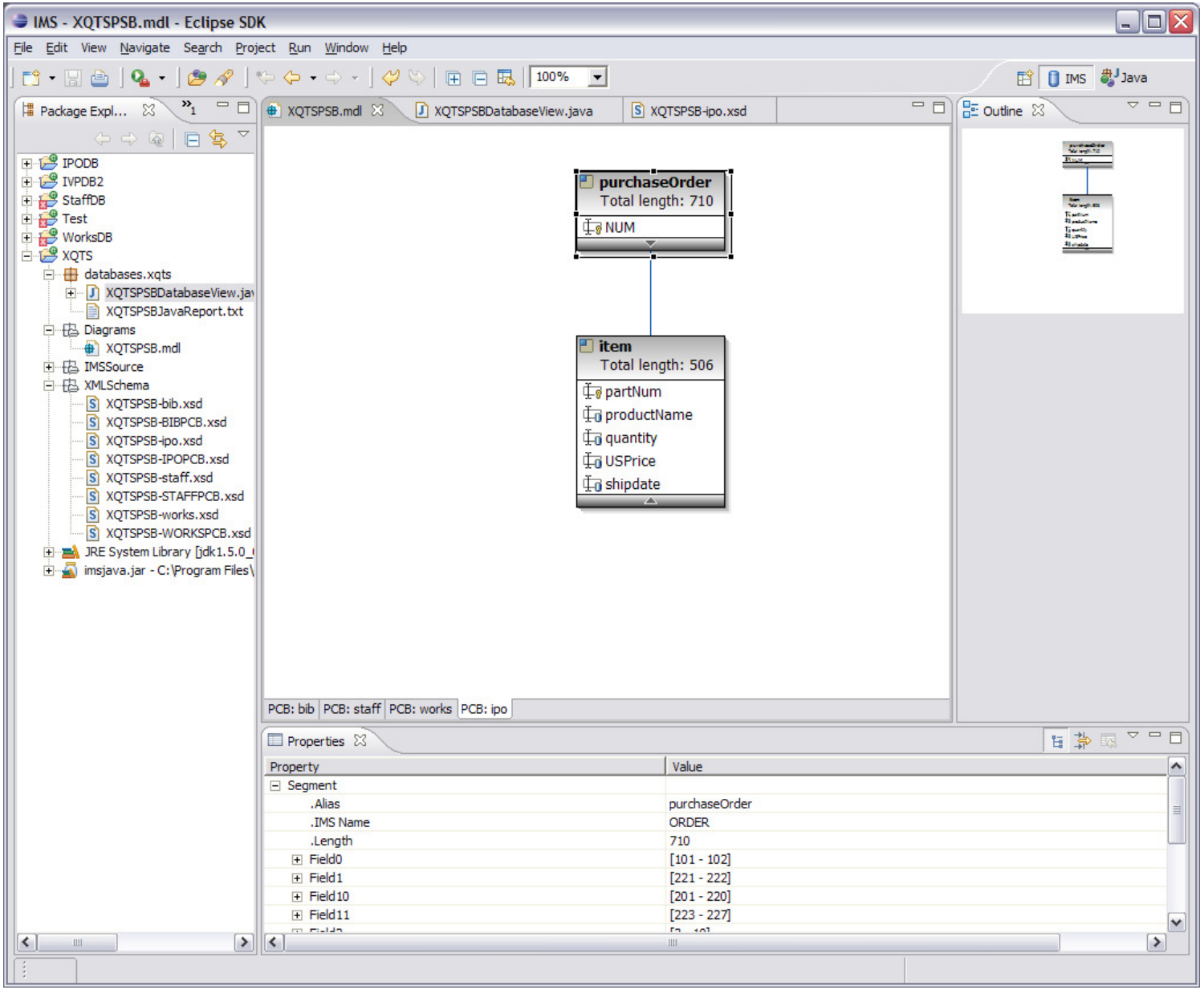


GUI DL/I Model Utility

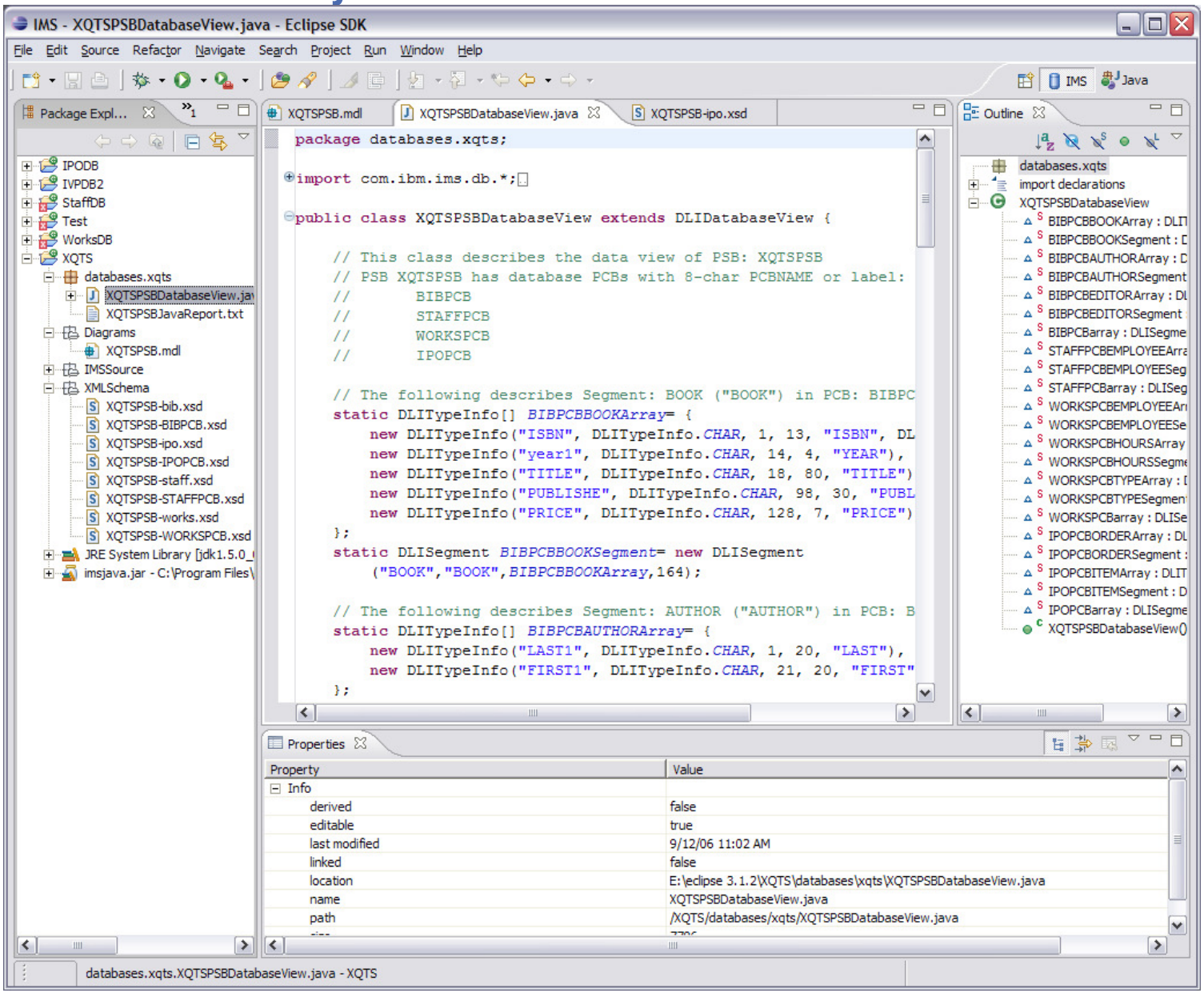
- Select DBDs / PSBs
  - must be FTPed locally
- Source is Parsed
  - any errors are reported
- XMI Metamodel
  - generated
  - opened for editing



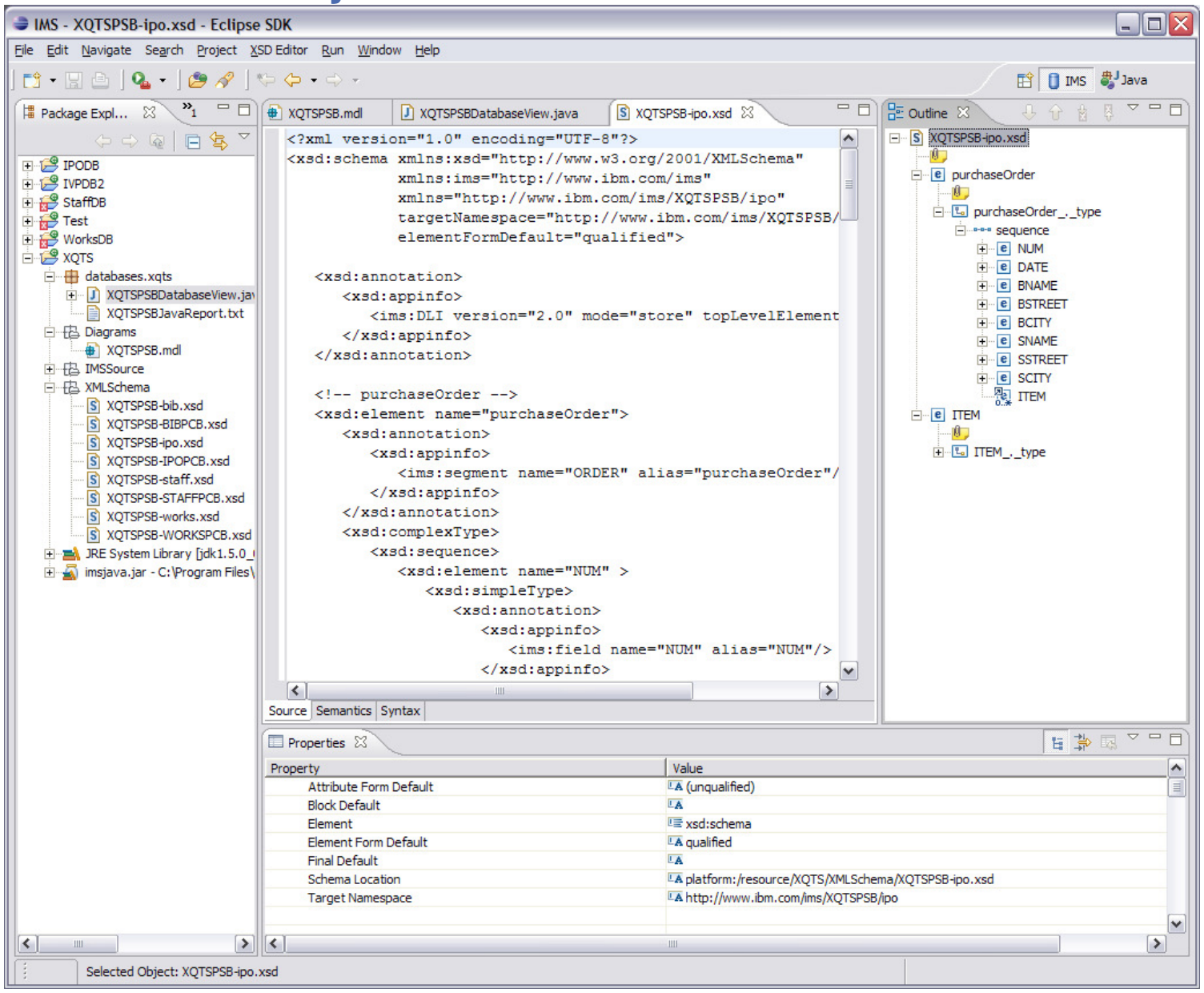
GUI DL/I Model Utility



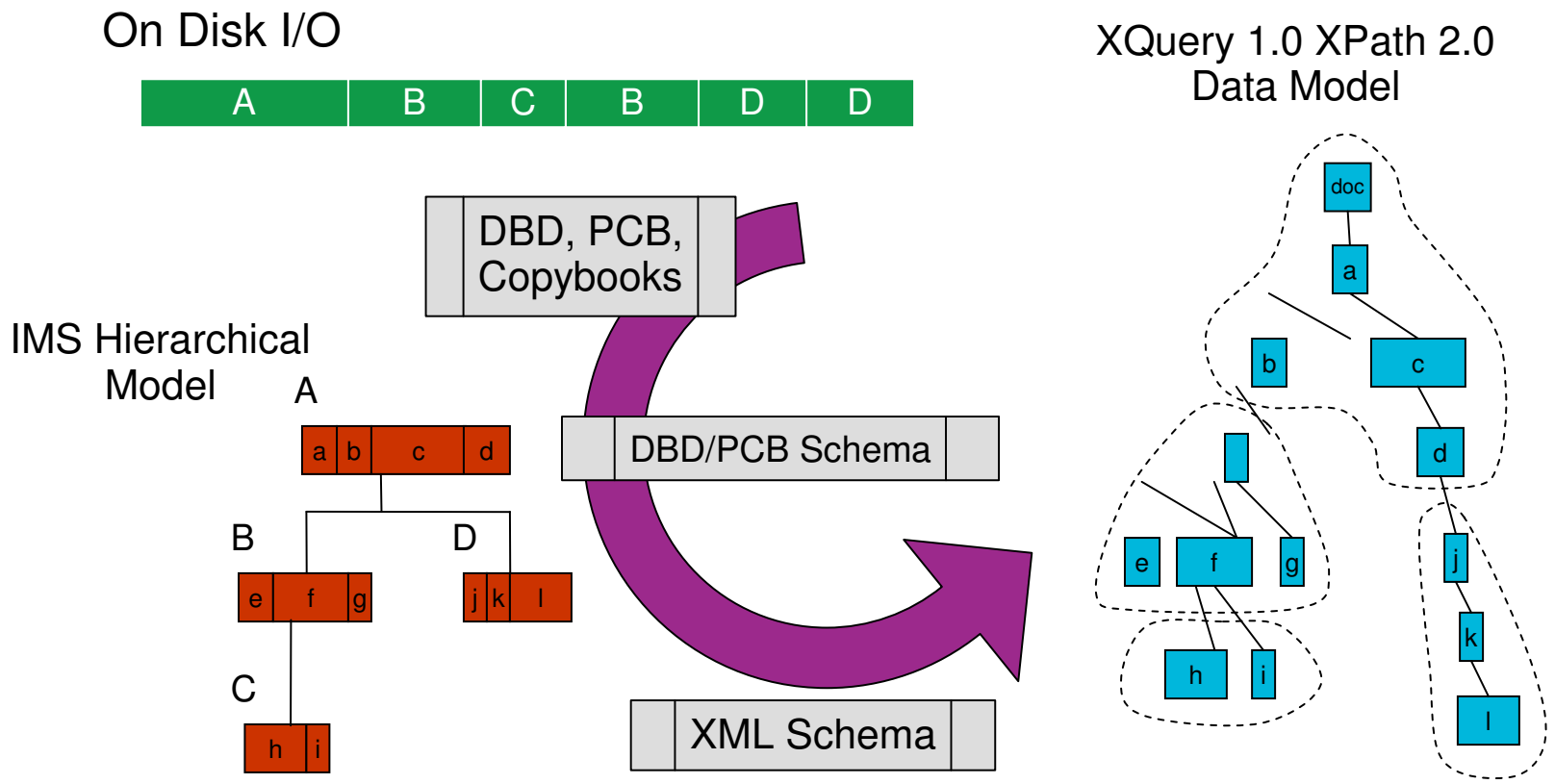
GUI DL/I Model Utility



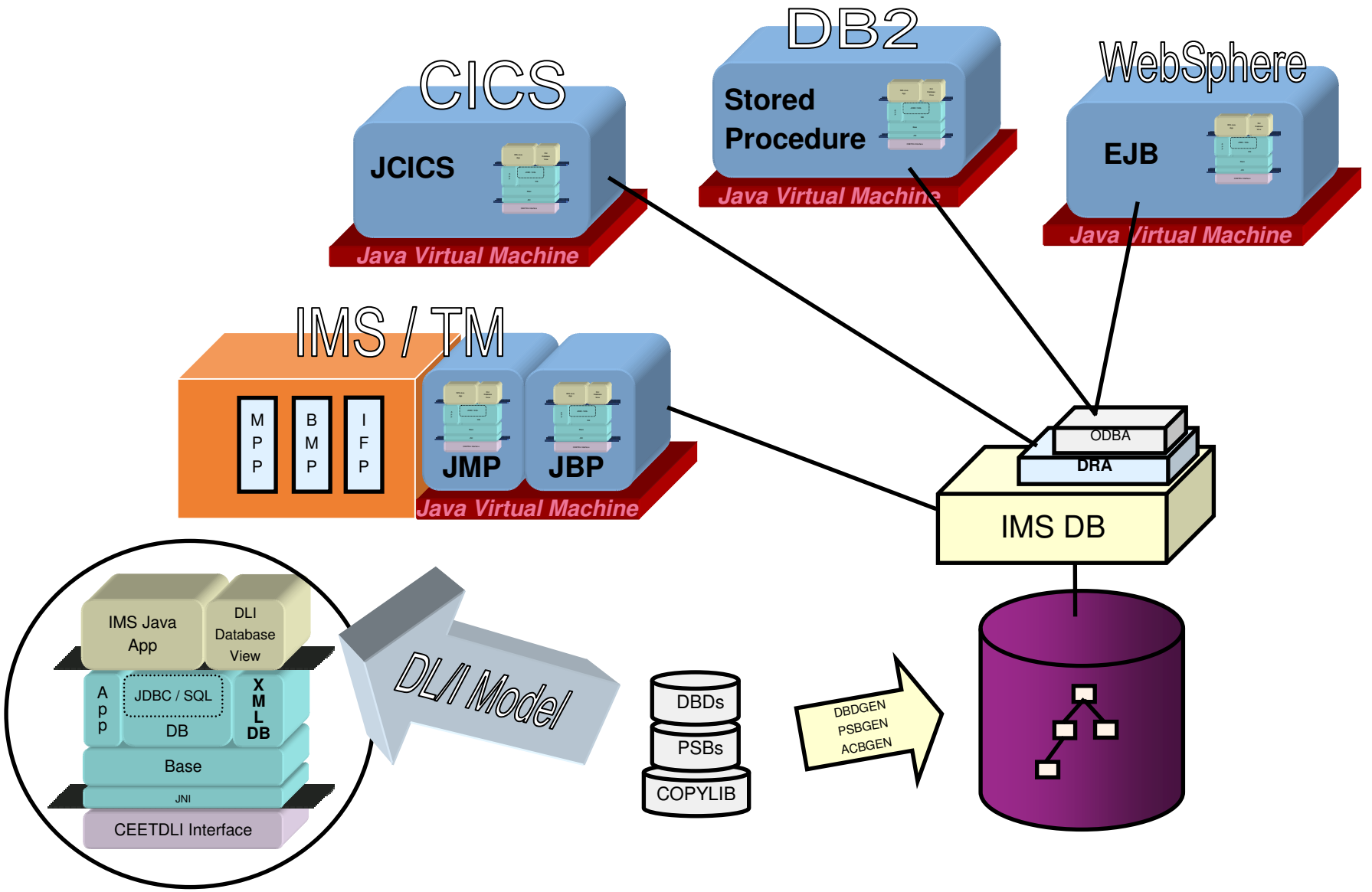
GUI DL/I Model Utility


IMS - XQTSPSB-ipo.xsd - Eclipse SDK
File Edit Navigate Search Project XSD Editor Run Window Help
IMS
Java
Package Expl...
IPODB
IVPDB2
StaffDB
Test
WorksDB
XQTS
databases.xqts
XQTSPSBDatabaseView.ja
XQTSPSBJavaReport.txt
Diagrams
XQTSPSB.mdl
IMSSource
XMLSchema
XQTSPSB-bib.xsd
XQTSPSB-BIBPCB.xsd
XQTSPSB-ipo.xsd
XQTSPSB-IPOPCB.xsd
XQTSPSB-staff.xsd
XQTSPSB-STAFFPCB.xsd
XQTSPSB-works.xsd
XQTSPSB-WORKSPCB.xsd
JRE System Library [jdk1.5.0_
imsjava.jar - C:\Program Files\
XQTSPSB.mdl
XQTSPSBDatabaseView.java
XQTSPSB-ipo.xsd
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:ims="http://www.ibm.com/ims"
xmlns="http://www.ibm.com/ims/XQTSPSB/ipo"
targetNamespace="http://www.ibm.com/ims/XQTSPSB/
elementFormDefault="qualified">
<xsd:annotation>
<xsd:appinfo>
<ims:DLI version="2.0" mode="store" topLevelElement
</xsd:appinfo>
</xsd:annotation>
<!-- purchaseOrder -->
<xsd:element name="purchaseOrder">
<xsd:annotation>
<xsd:appinfo>
<ims:segment name="ORDER" alias="purchaseOrder"/
</xsd:appinfo>
</xsd:annotation>
<xsd:complexType>
<xsd:sequence>
<xsd:element name="NUM" >
<xsd:simpleType>
<xsd:annotation>
<xsd:appinfo>
<ims:field name="NUM" alias="NUM"/>
</xsd:appinfo>
Source Semantics Syntax
Outline
XQTSPSB-ipo.xsd
purchaseOrder
purchaseOrder_._type
sequence
NUM
DATE
BNAME
BSTREET
BCITY
SNAME
SSTREET
SCITY
ITEM
ITEM
ITEM_._type
Properties
Property
Value
Attribute Form Default
(unqualified)
Block Default
Element
xsd:schema
Element Form Default
qualified
Final Default
Schema Location
platform:/resource/XQTS/XMLSchema/XQTSPSB-ipo.xsd
Target Namespace
http://www.ibm.com/ims/XQTSPSB/ipo
Selected Object: XQTSPSB-ipo.xsd

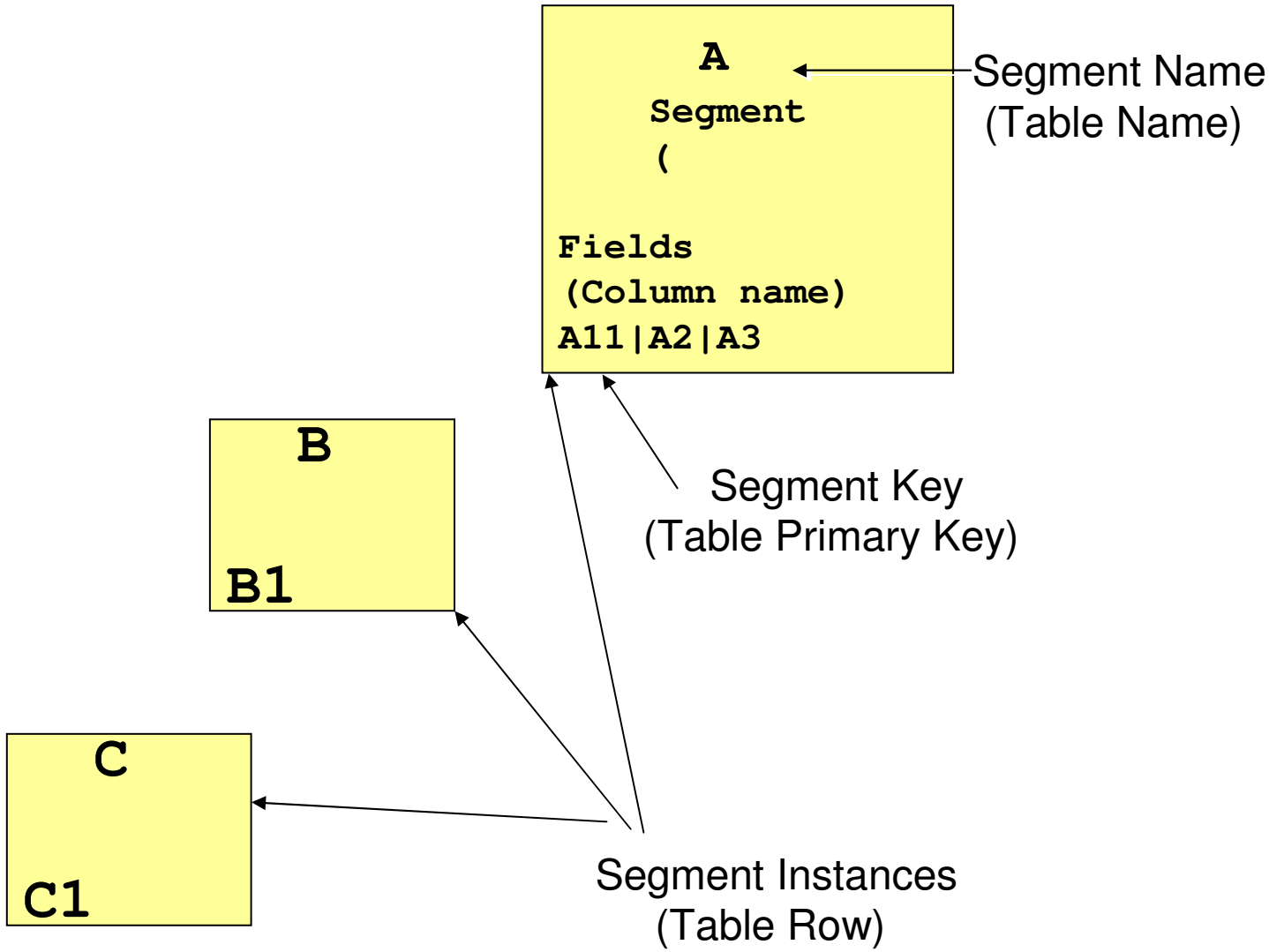
XML Visualization



IMS JDBC Runtime



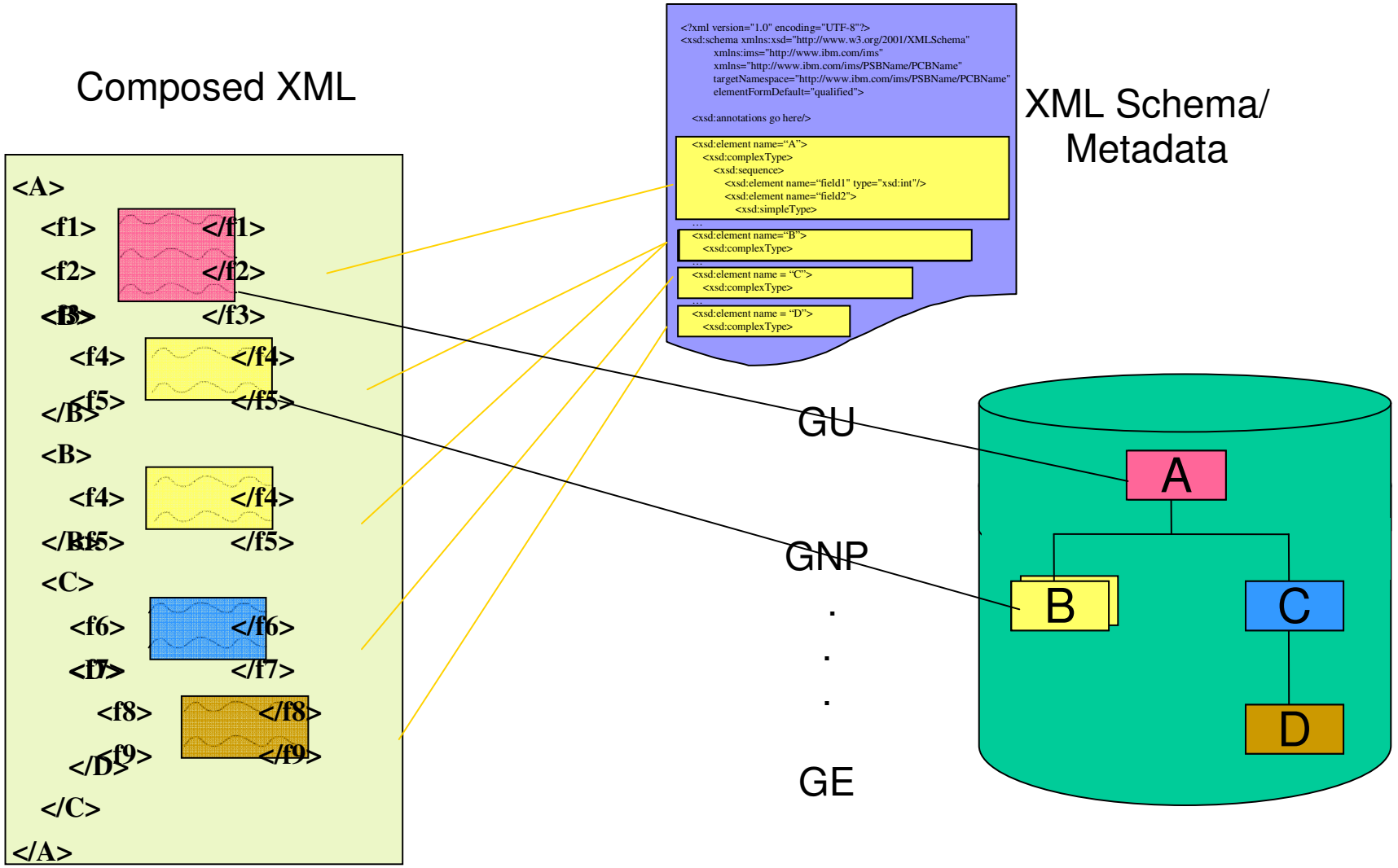
IMS and JDBC SQL Concepts



Decomposed Storage

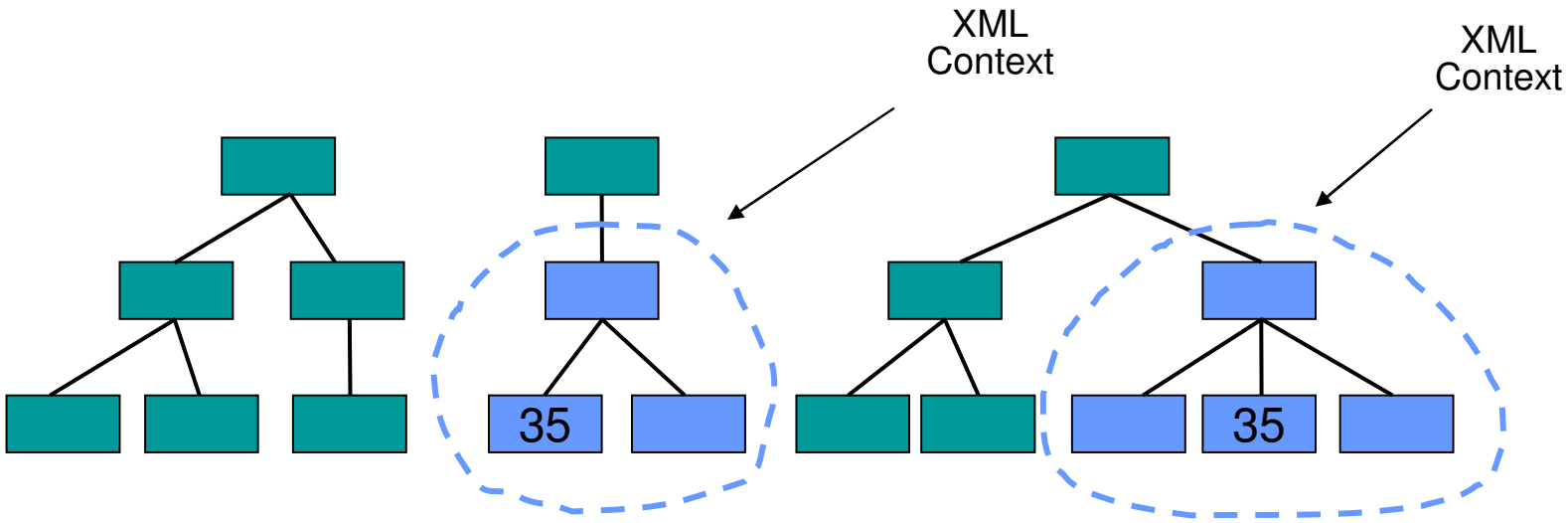
- XML document must be parsed and validated.
- Data is converted to *traditional* IMS types
  - COMP-1, COMP-2, etc.
  - EBCDIC CHAR, Picture Strings
- Stored data is searchable by IMS and transparently accessible by non-XML enabled applications.

Decomposed XML Retrieval in IMS



RetrieveXML() UDF

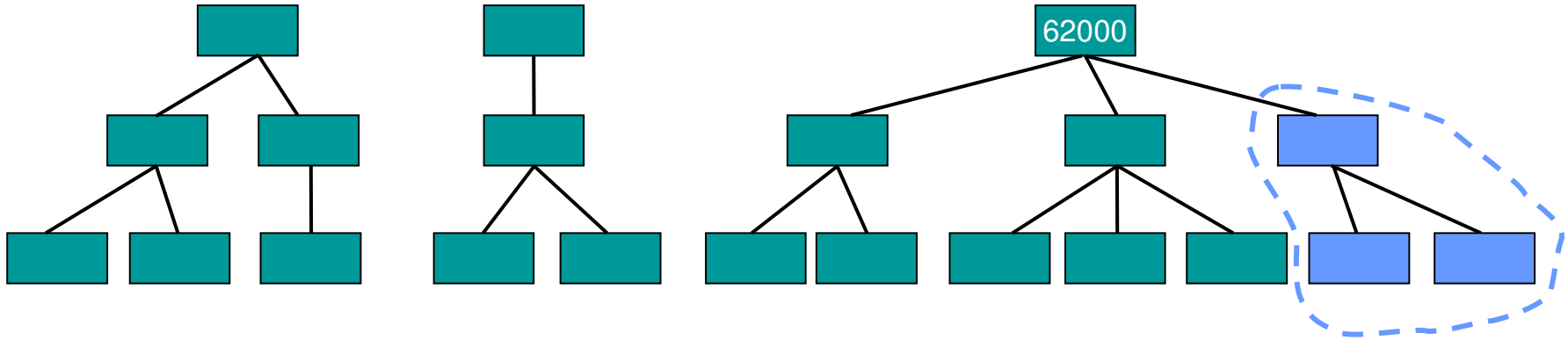
```
SELECT retrieveXML(Dealer)
FROM Order
WHERE Order.Ordernum = '35'
```



**Two Rows of XML CLOBs in the ResultSet*

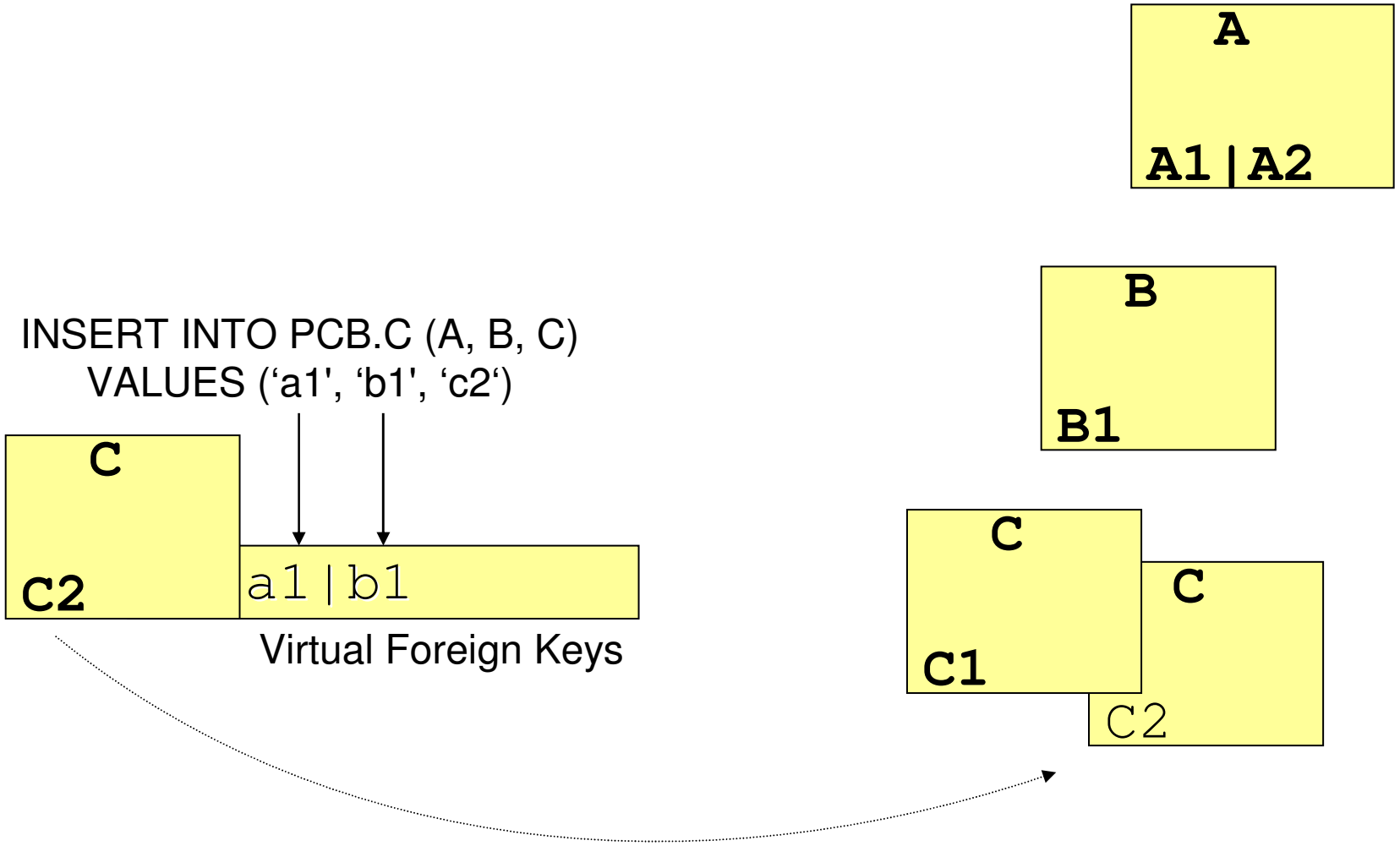
StoreXML() UDF

```
INSERT INTO B (storeXML())
VALUES (?)
WHERE A.fieldA = ’62000’
```

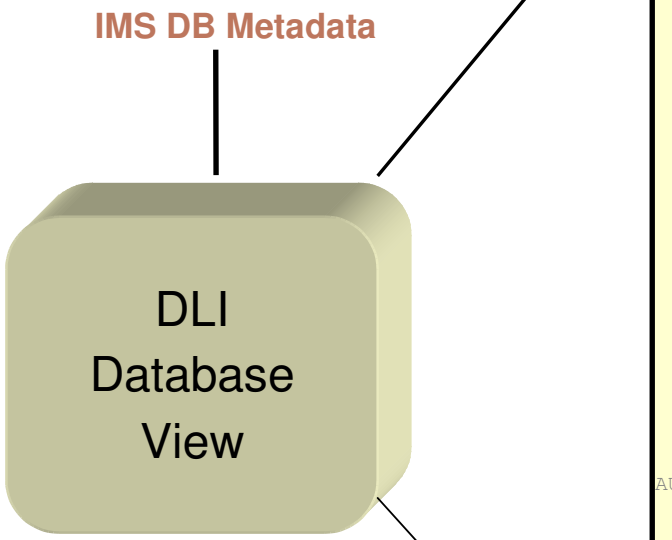


**Insert Statement must be a Prepared Statement*

IMS 11 Virtual Foreign Keys
INSERT Example



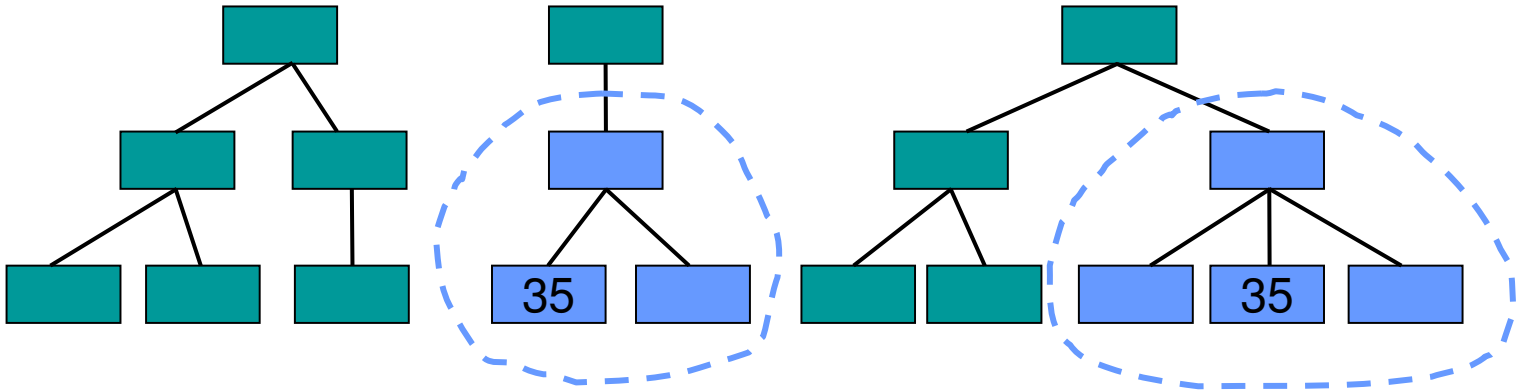
Follow on support
DLIDatabaseView with XML datatype




```
package samples.dealership;

import com.ibm.ims.db.*;
import com.ibm.ims.base.*;

public class AUTPSB11DatabaseView extends DLIDatabaseView {
  // The following DLITypeInfo[] array describes Segment: DEALER in PCB: AUTOLPCB
  static DLITypeInfo[] AUTOLPCBDEALERArray= {
      new DLITypeInfo("DealerNo",     DLITypeInfo.CHAR,  1,  4, "DLRNO"),
      new DLITypeInfo("DealerName",   DLITypeInfo.CHAR,  5, 30, "DLRNAME"),
      new DLITypeInfo("DealerCity",   DLITypeInfo.CHAR, 35, 10, "CITY"),
      new DLITypeInfo("DealerZip",    DLITypeInfo.CHAR, 45, 10, "ZIP"),
      new DLITypeInfo("DealerPhone",  DLITypeInfo.CHAR, 55,  7, "PHONE"),
      new DLITypeInfo("DealerXML", "Dealer.xsd", DLITypeInfo.XML)
  };
  static DLISegment AUTOLPCBDEALERSegment= new DLISegment
          ("DealerSegment","DEALER",AUTOLPCBDEALERArray,61);
  ...

  // An array of DLISegmentInfo objects follows to describe the view for PCB:
AUTOLPCB
  static DLISegmentInfo[] AUTOLPCBarray = {
      new DLISegmentInfo(AUTOLPCBDEALERSegment,DLIDatabaseView.ROOT),
      new DLISegmentInfo(AUTOLPCBMODELSegment,0),
      new DLISegmentInfo(AUTOLPCBORDERSegment,1),
      new DLISegmentInfo(AUTOLPCBSALESSegment,1),
      new DLISegmentInfo(AUTOLPCBSTOCKSegment,1),
      new DLISegmentInfo(AUTOLPCBSTOCSALESegment,4),
      new DLISegmentInfo(AUTOLPCBSALESINFSegment,5)
  };
  ...
}
```

Follow on IMS JDBC syntax for XML retrieval

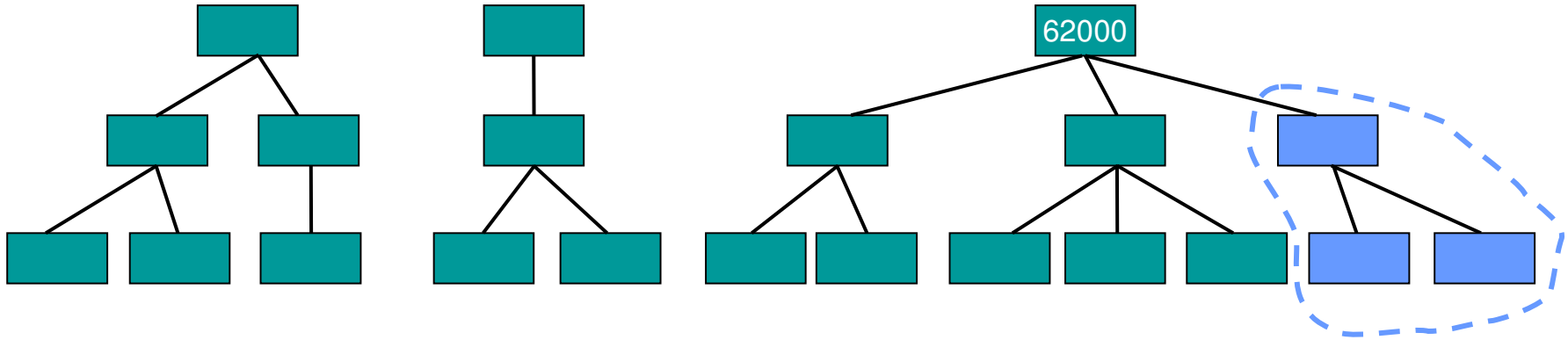
```
SELECT Dealer.DealerXML,
FROM Dealer, Order
WHERE Order.Ordernum = ’35’
```



**Two Rows of XML CLOBs in the ResultSet*

Follow IMS JDBC syntax for XML insert

```
INSERT INTO B (B.BXML, A_fieldA)
VALUES (?, “62000”)
```

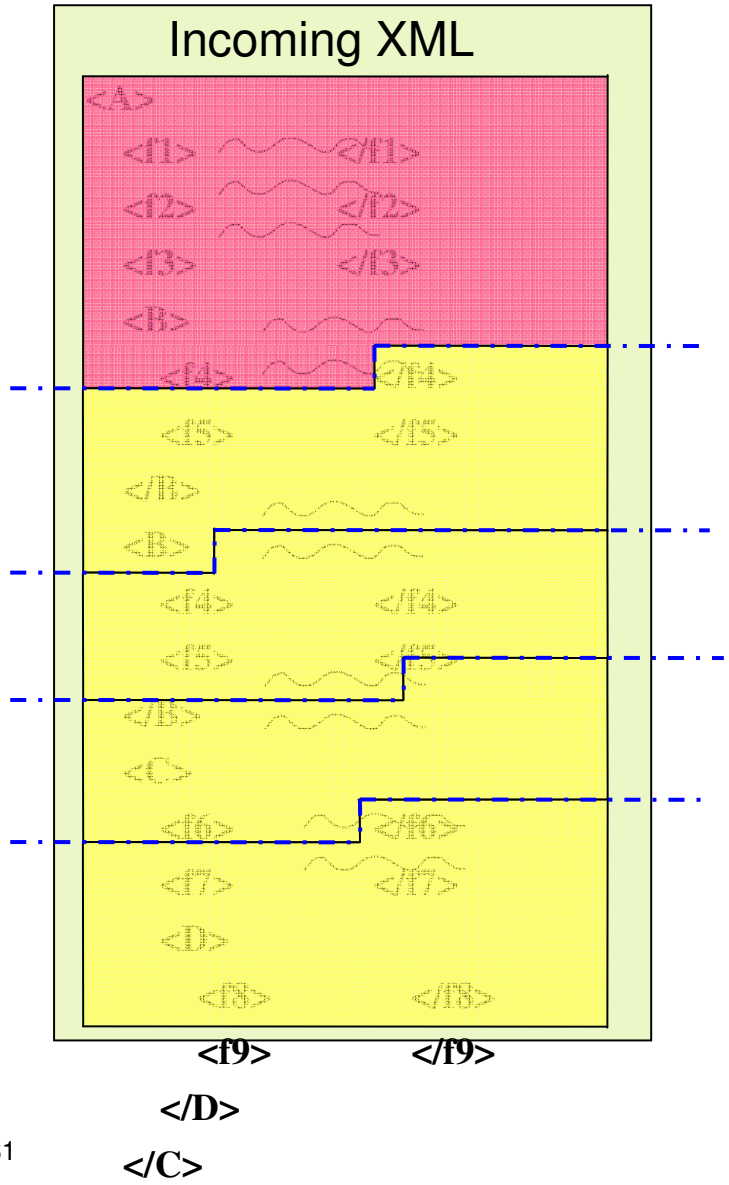


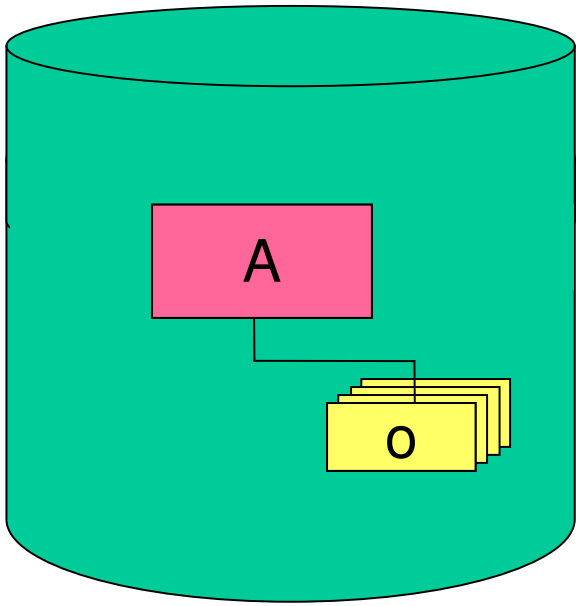
**Insert Statement must be a Prepared Statement*

Intact Storage

- No (or little) XML Parsing or Schema validation
  - Storage and Retrieval Performance
- No (or little) data type conversions
  - Unicode storage
- Stored documents are no longer searchable by IMS and only accessible to XML-enabled applications
  - XPath side segments

Intact XML Storage in IMS





Decomposed vs. Intact Storage

- Decomposed (*data-centric storage*)
  - XML tags are stripped from XML data
  - Identical as current IMS storage
  - Strict data-centric XML Schema validated data
  - EBCDIC encoding
  - Searching on IMS Search Fields

- Intact (*document-centric storage*)
  - Entire XML document is stored (including tags)
  - Relaxed un-validated data
  - Any desired encoding is possible
  - Searching is through XPath specified and generated Secondary Indexed Side Segments

IMS 10 XQuery FLWOR Expressions
Virtual XML Garden on IBM alphaWorks

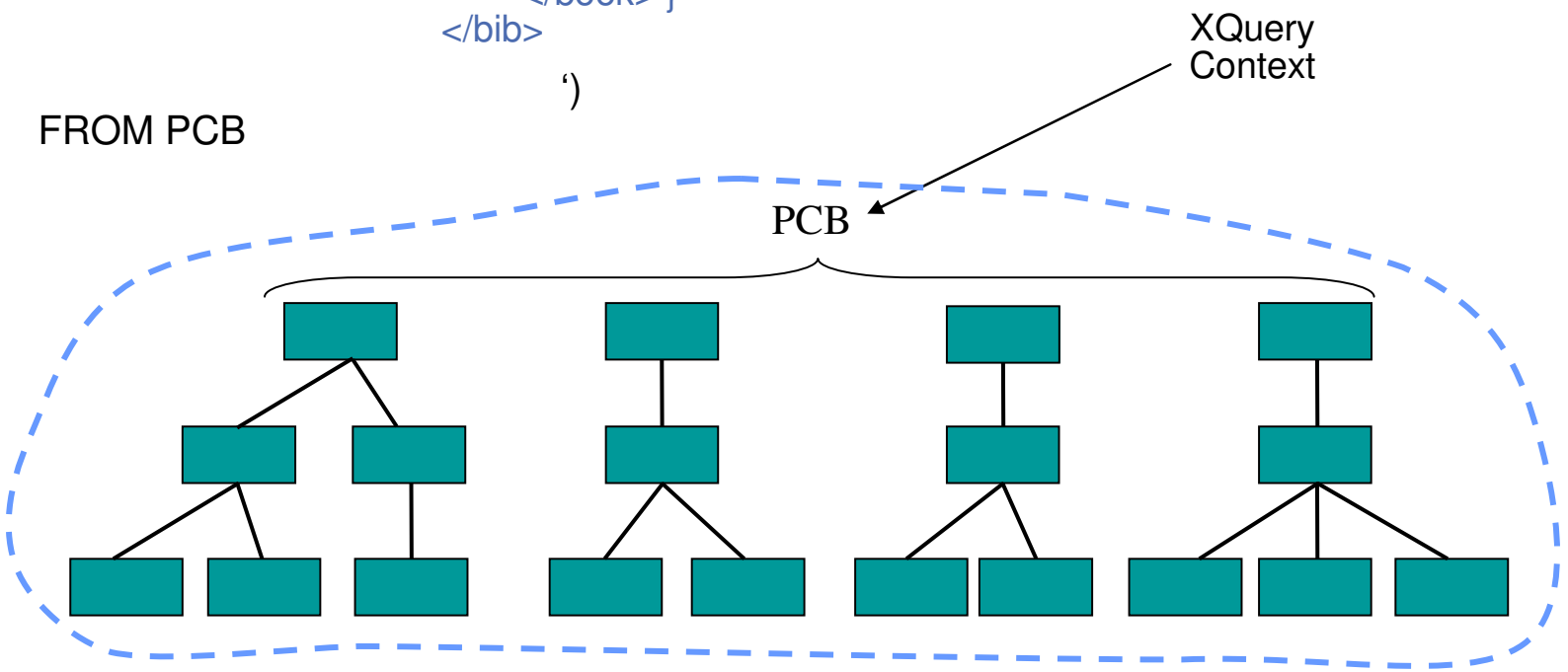
- **F**OR: iterates through a sequence, bind variable to items
- **L**ET: binds a variable to a sequence
- **W**HERE: eliminates items of the iteration
- **O**RDER BY: reorders items of the iteration
- **R**ETURN: constructs query results

```
<bib> {
   for $b in /bib/book
   let $title := $b/title
   where $b/publisher = "Addison-Wesley“
   order by $b/@year
   return
      <book year="{ $b/@year }">
         { $title }
      </book>
} </bib>
```

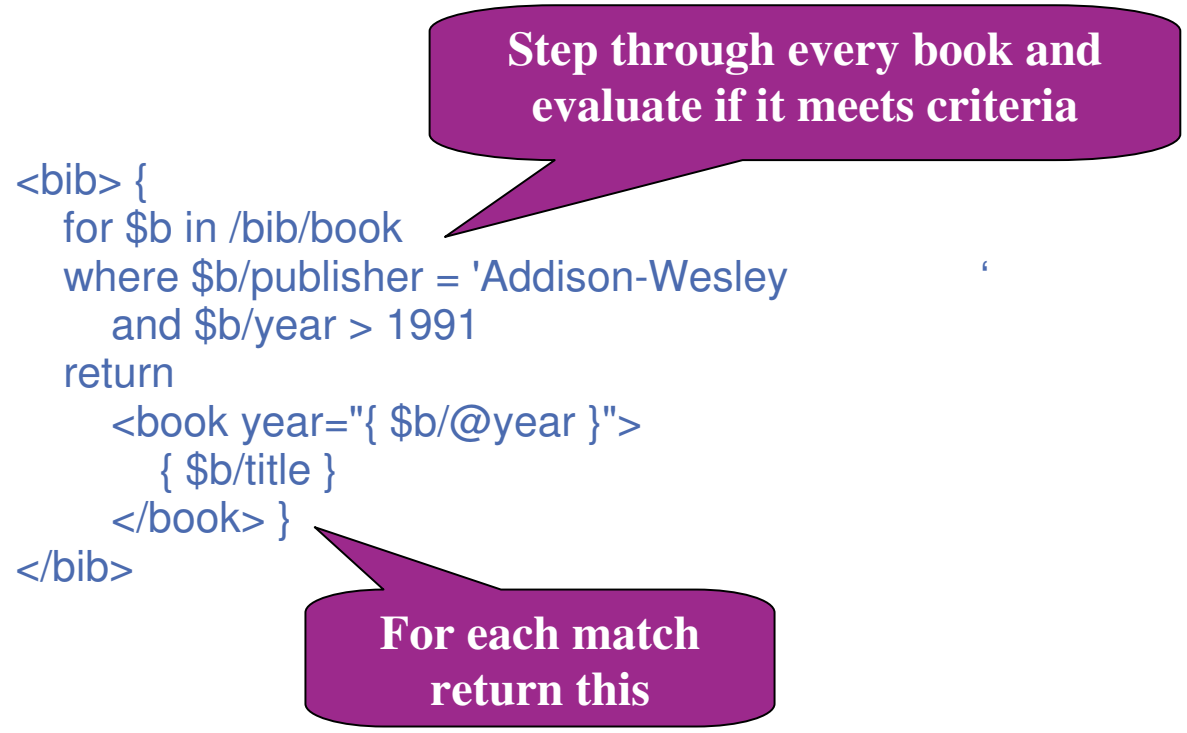
```
<bib>
   <book year=“1992">
      <title>Advanced Programming in the Unix
   </book>
   <book year=“1994">
      <title>TCP/IP Illustrated</title>
   </book>
</bib>
```

Extended JDBC interface…

```
SELECT retrieveXML( ‘
                    <bib> {
                       for $b in /bib/book
                       where $b/publisher = 'Addison-Wesley                ‘
                          and $b/@year > 1991
                       return
                          <book year="{ $b/@year }">
                             { $b/title }
                          </book> }
                    </bib>
                          ‘)
FROM PCB
```

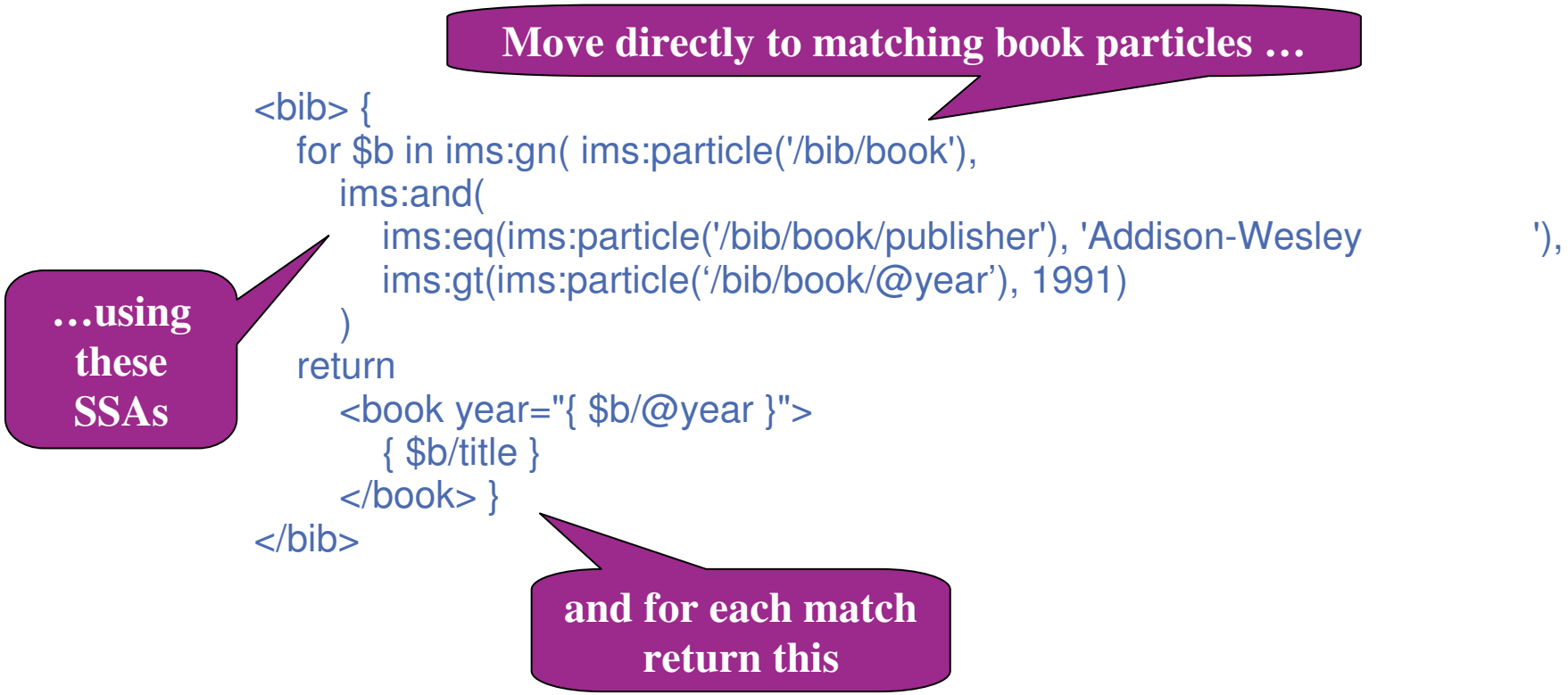


Optimizing XQuery for IMS



Optimizing XQuery for IMS

Optimized using native IMS XQuery functions



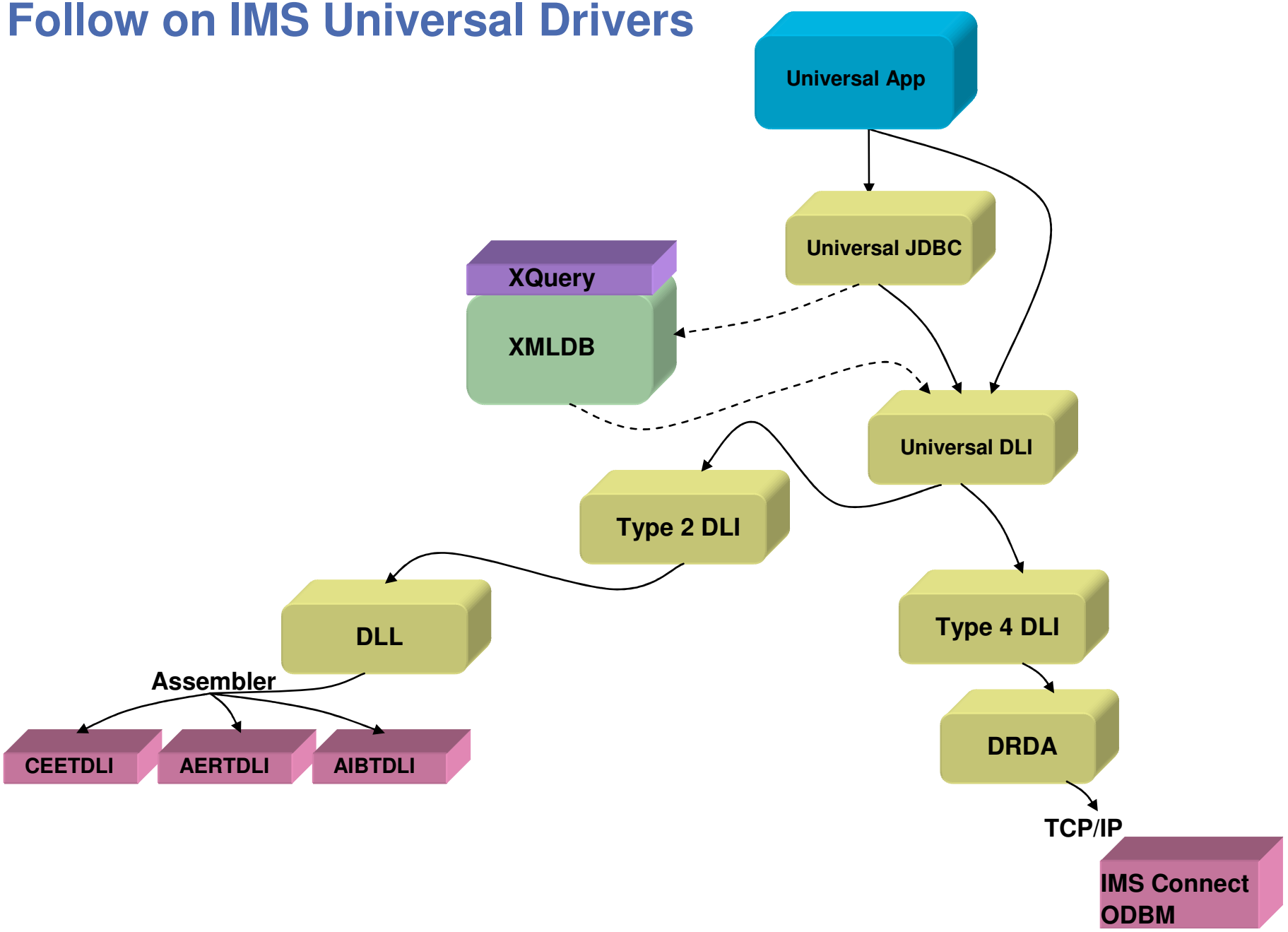
Our IMS V9 POC

- We downloaded the 'XML Garden' demo from the IBM web site.
- We extracted the Java zip file contents and loaded the JAVA jar files, schema, and XML data into the mainframe OMVS environment (Unix).
- We set up the IMS environment: HDAM database, PSBs, transaction definitions, PDS parameter members, and the IMS/TM JBP and JMP regions.
- We executed the JBP JCL on the mainframe and the Java application loaded XML data into the IMS HDAM database.
- We executed the JMP on the mainframe and retrieved the data.

IMS V10 POC plan

- Down load the DLI Model utility from the IBM web site and build an XML schema for an existing popular database.
- Write some Java and SQL to put data into the database and selectively retrieve data.
- Present to application partners to solicit collaboration and project ideas to take advantage of this new tool.
- We still have a problem but then . . . wait for it . . .

Follow on IMS Universal Drivers



IMS 11 Open Database has Type 2 and Type 4 drivers

- Right now we are still limited to running our Java only on the mainframe (OMVS) because there are only classic API drivers available.

- IMS 11 has Universal
  - Type 2 drivers for local access
  - Type 4 drivers that will open up distributed access!

IMS

- IMS is leveraging current and new technology!
  - Access by Java and X-Query.
  - Access via the IMS SOAP gateway.
  - No more space limitations.
  - A reliable DBMS.
  - A fast DBMS.