IBM Software Group Information Management

DB2 for z/OS – Relational Database

Christian Daser, Technical Sales DB2 z/OS
chris.daser@de.ibm.com

Hanseatic Mainframe Summit 2008

Friday, 12 September 2008

Agenda

- **Datenbank-Grundlagen**
- DB2 for z/OS Einführung
- DB2 for z/OS Architektur & Utilities
- Anwendungsentwicklung
  - Traditionell
  - Java, Web
  - Stored Procedures
- Administration
  - DB2I
  - DB2 Admin Tool
- Trends
  - WebServices
  - XML

Terminology

- **Database**
  organized collection of data
  collection of records stored in a systematic way

- **Database Management System (DBMS)**
  used to manage a database
  to optimize the storage and performance of a database
  controls high volume access to the database

- **Schema**
  structural description of the database objects

- **SQL**
  control/query language for a relational DBMS
  DML, DDL, DCL

Database requirements

- ACID
  - Atomicity
    Guarantee that either all tasks of a transaction are performed or none
  - Consistency
    Data is in a legal state when transactions begin or end, integrity is always ensured
  - Isolation
    transactions appear isolated from other operations. Noone can look into the ongoings of a TA
  - Durability
    Once a transaction ends, the data is persistent and will survive a system failure.

- Storage for important data
- Huge amount of data
- Backup, recovery
- Auditing
- High volume queries
- Multiple users

Referential integrity & joins

```
CREATE TABLE DEPTARTMENT
  (DEPTNO                 CHAR(3) FOR SBCS DATA NOT NULL ,
   DEPTNAME               VARCHAR(36) FOR SBCS DATA NOT NULL ,
   MGRNO                  CHAR(6) FOR SBCS DATA WITH DEFAULT NULL ,
   ADMRDEPT               CHAR(3) FOR SBCS DATA NOT NULL ,
   LOCATION               CHAR(16) FOR SBCS DATA WITH DEFAULT NULL ,
CONSTRAINT DEPTNO PRIMARY KEY (DEPTNO))


CREATE TABLE EMPLOYEE
  (EMPNO                  CHAR(6) FOR SBCS DATA NOT NULL ,
   FIRSTNME               VARCHAR(12) FOR SBCS DATA NOT NULL ,
   MIDINIT                CHAR(1) FOR SBCS DATA NOT NULL ,
   LASTNAME               VARCHAR(15) FOR SBCS DATA NOT NULL ,
   WORKDEPT               CHAR(3) FOR SBCS DATA WITH DEFAULT NULL ,
   PHONENO                CHAR(4) FOR SBCS DATA WITH DEFAULT NULL ,
   HIREDATE               DATE WITH DEFAULT NULL ,
   JOB                    CHAR(8) FOR SBCS DATA WITH DEFAULT NULL ,
   EDLEVEL                SMALLINT NOT NULL ,
   SEX                    CHAR(1) FOR SBCS DATA WITH DEFAULT NULL ,
   BIRTHDATE              DATE WITH DEFAULT NULL ,
   SALARY                 DECIMAL(9, 2) WITH DEFAULT NULL ,
   BONUS                  DECIMAL(9, 2) WITH DEFAULT NULL ,
   COMM                   DECIMAL(9, 2) WITH DEFAULT NULL ,
CONSTRAINT EMP_PK PRIMARY KEY (EMPNO),
CONSTRAINT EMP_REF FOREIGN KEY (WORKDEPT) REFERENCES DEPARTMENT (DEPTNO))
```

EMPLOYEE

EMPNO	FIRSTNME	MIDINIT	LASTNAME	WORKDEPT	PHONENO	HIREDATE	JOB		SEX
000010	CHRISTINE	I	HAAS	A00	3978	Jan 1, 1965	PRES	18	F
000020	MICHAEL	L	THOMPSON	B01	3476	Oct 10, 1973	MANAGER	18	M
000030	SALLY	A	KWAN	C01	4738	Apr 5, 1975	MANAGER	20	F
000050	JOHN	B	GEYER	E01	6789	Aug 17, 1949	MANAGER	16	M
000060	IRVING	F	STERN	D11	6423	Sep 14, 1973	MANAGER	16	M
000070	EVA	D	PULASKI	D21	7831	Sep 30, 1980	MANAGER	16	F
000090	EILEEN	W	HENDERSON	E11	5498	Aug 15, 1970	MANAGER	16	F
000100	THEODORE	Q	SPENSER	E21	0972	Jun 19, 1980	MANAGER	14	M
000110	VINCENZO	G	LUCCHESSI	A00	3490	May 16, 1958	SALESREP	19	M
000120	SEAN		O'CONNELL	A00	2167	Dec 5, 1963	CLERK	14	M
000130	DOLORES	M	QUINTANA	C01	4578	Jul 28, 1971	ANALYST	16	F
000140	HEATHER	A	NICHOLLS	C01	1793	Dec 15, 1976	ANALYST	18	F
000150	BRUCE		ADAMSON	D11	4510	Feb 12, 1972	DESIGNER	16	M
000160	ELIZABETH	R	PIANKA	D11	3782	Oct 11, 1977	DESIGNER	17	F
000170	MASATOSHI	J	YOSHIMURA	D11	2890	Sep 15, 1978	DESIGNER	16	M
000180	MARILYN	S	SCOUTTEN	D11	1682	Jul 7, 1973	DESIGNER	17	F

DEPARTMENT

DEPTNO	DEPTNAME	MGRNO	ADMRDEPT	LOCATION
A00	SPIFFY COMPUTE...	000010	A00	
B01	PLANNING	000020	A00	
C01	INFORMATION CE...	000030	A00	
D01	DEVELOPMENT CE...		A00	
D11	MANUFACTURING...	000060	D01	
D21	ADMINISTRATION ...	000070	D01	
E01	SUPPORT SERVIC...	000050	A00	
E11	OPERATIONS	000090	E01	
E21	SOFTWARE SUPP...	000100	E01	

```
select e.firstname, e.lastname
from employee e, department d
where e.workdept = d.deptno
and d.admrdept= ' D01'
```

Agenda

- Datenbank-Grundlagen
- **DB2 for z/OS Einführung**
- DB2 for z/OS Architektur & Utilities
- Anwendungsentwicklung
  - Traditionell
  - Java, Web
  - Stored Procedures
- Administration
  - DB2I
  - DB2 Admin Tool
- Trends
  - WebServices
  - XML

Guess Who Said This ?

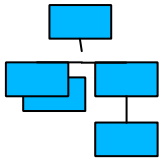
eWEEK (www.eweek.com) 31-Oct-2003:

I make fun of a lot of other databases - all other databases, in fact, except the mainframe version of DB2. It's a first-rate piece of technology.

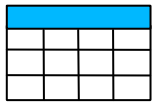
Larry Ellison, Oracle's Founder and CEO

IBM‘s database history

- <1970 IMS



- 1970 Codd, theory of relational databases, SQL
- 1983 DB2 MVS Version 1
- 1987 DB2 OS/2 („DB2/2“)
- 1988 DB2 AS/400
- 90s DB2 LUW, Everyplace



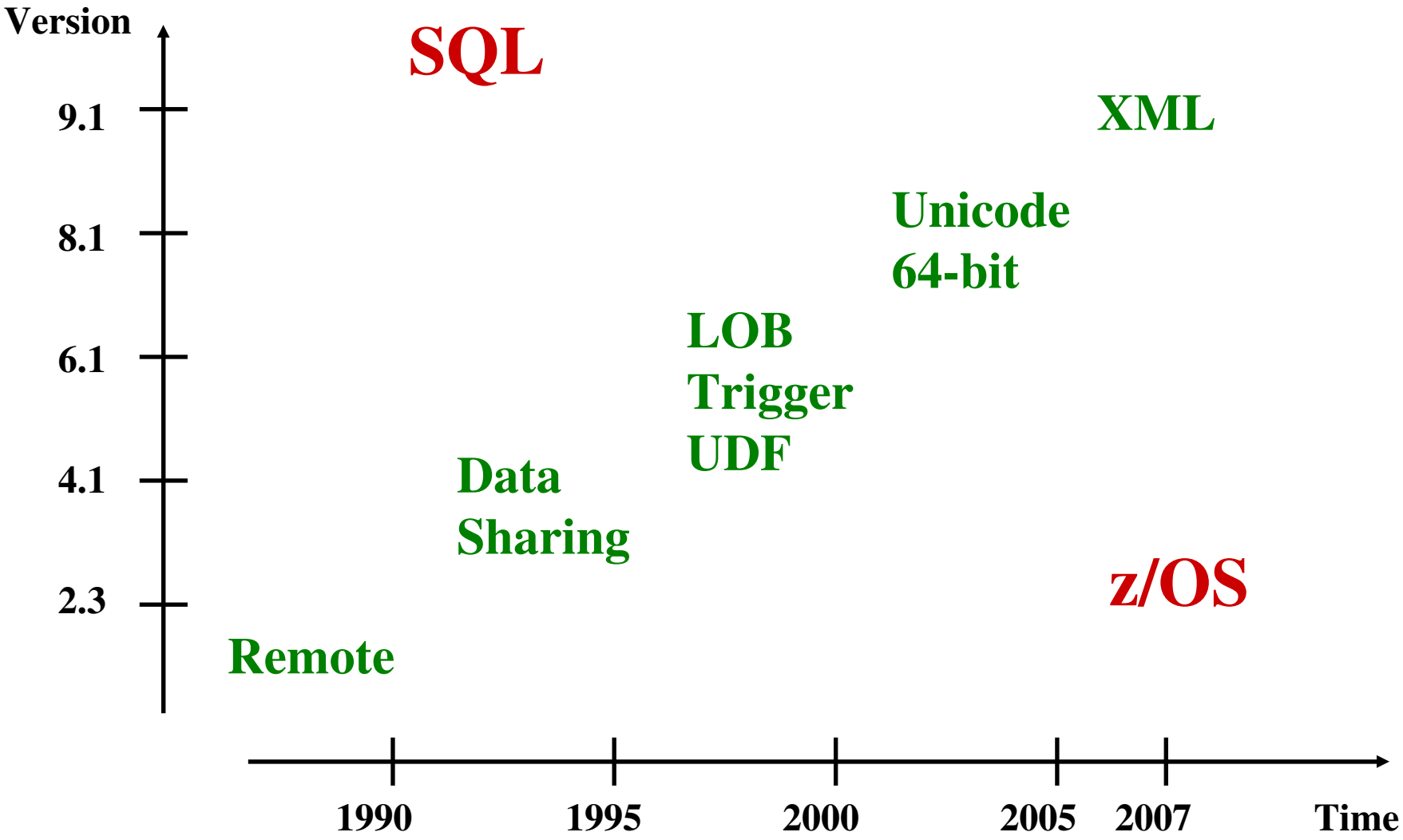
- today DB2 for z/OS Version 9
  DB2Express-C
  XML

Warum setzen Kunden Datenbanken auf System z ein?

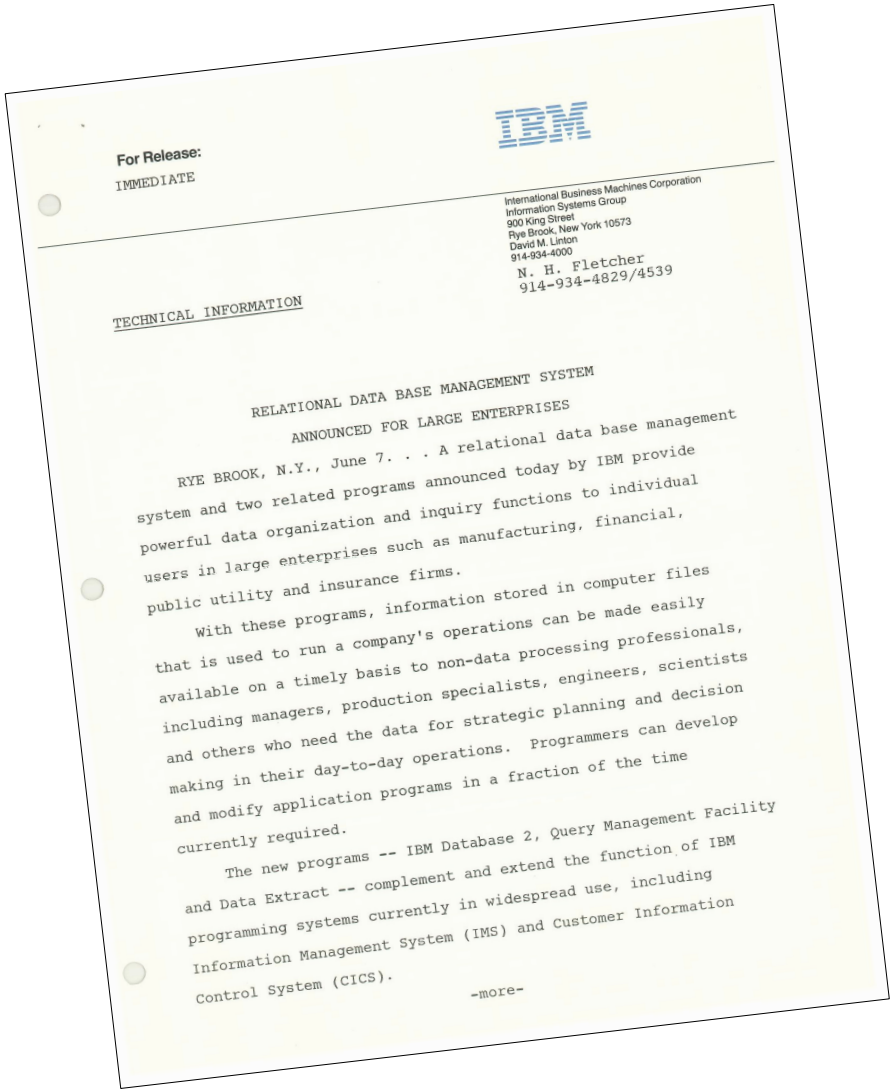
- Continuous Availability
  - 7 x 24
  - Reduces outages
- High Performance and Scalability
  - Scalable
  - Growth
  - Single and multi-system
  - Huge amount of databases
  - Huge amount of users
- Wide Range of Workloads
  - OLTP and Batch
  - Web Workload with peaks
  - ERP/CRM Solutions (SAP...)
  - BI & Warehousing
  - ODS and mixed workloads
  - **...**

- Database Technology Leadership
  - Cost-based Optimizer, Query-Rewrite
  - Parallelism
  - Web-enablement, XML
- Highly automated database environment
  - Complete utilities and tools portfolio
  - Offerings
  - z/OS Systemmanagement
- Platform compatibility
- Accounting
- Support for Open Standards
  - SQL/ANSI, ODBC, JDBC/SQLJ
  - ...,

DB2 z/OS family history



DB2 Silver Anniversary - 25 years of Innovation



For Release:
IMMEDIATE

IBM

International Business Machines Corporation
Information Systems Group
900 King Street
Rye Brook, New York 10573
David M. Linton
914-934-4000

N. H. Fletcher
914-934-4829/4539

TECHNICAL INFORMATION

RELATIONAL DATA BASE MANAGEMENT SYSTEM
ANNOUNCED FOR LARGE ENTERPRISES

RYE BROOK, N.Y., June 7. . . A relational data base management system and two related programs announced today by IBM provide powerful data organization and inquiry functions to individual users in large enterprises such as manufacturing, financial, public utility and insurance firms.

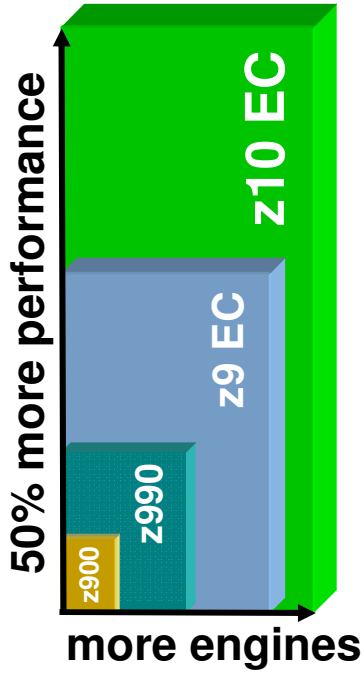
With these programs, information stored in computer files that is used to run a company's operations can be made easily available on a timely basis to non-data processing professionals, including managers, production specialists, engineers, scientists and others who need the data for strategic planning and decision making in their day-to-day operations. Programmers can develop and modify application programs in a fraction of the time currently required.

The new programs -- IBM Database 2, Query Management Facility and Data Extract -- complement and extend the function of IBM programming systems currently in widespread use, including Information Management System (IMS) and Customer Information Control System (CICS).

-more-

- 14th release
  - DB2 for z/OS Version 9
  - pureXML
- 10000+ licenses (mostly FSS)
- Typical DB2 for z/OS customer
  - Team of 2-10
  - DASD 200TB
  - 10-20 Mio Online Transactions per day
  - 20 DB2 Systems
  - Availability: 99,9%
  - Response Time < 1s: 99,9%
  - CPU Utilization > 90%
  - Threads per day: 3.5 Mio
  - SQLs per day: 3.900 Mio

DB2 for z/OS Version 9 and System z10



- ✓ Up to 2x performance increase in CPU intensive tasks because of new 4.4 GHz chips
- ✓ Up to 64 processor and up to 1,5 Terabyte real memory per server

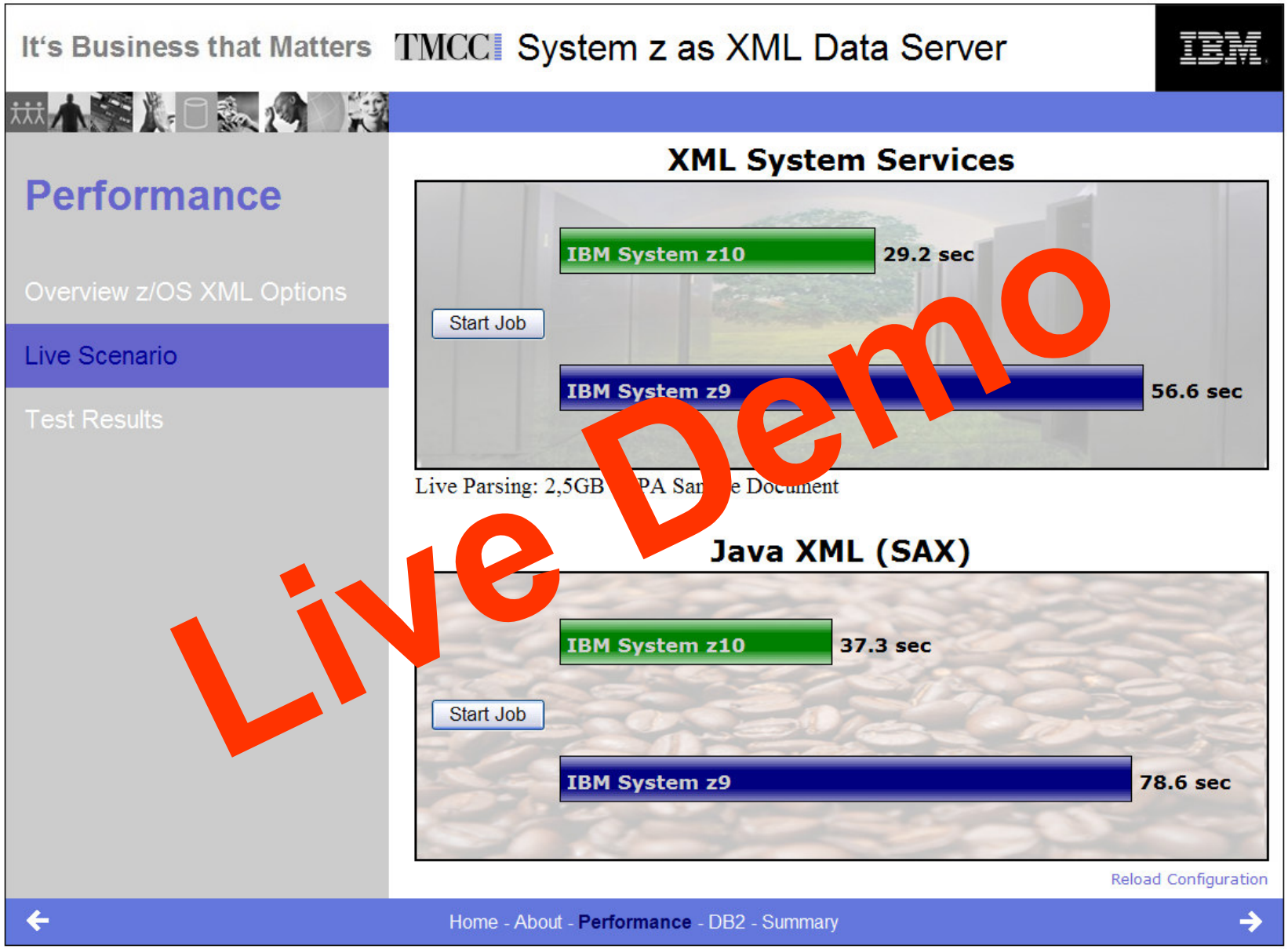
→ **Improved SQL & XML parsing performance**

- ✓ Further extended exploitation of zIIP & zAAP specialty engines

→ **Cost-efficient processing of DB2 workloads**

- ✓ Less (planned) outages and more flexibility (e.g. On/Off CoD)

→ **Higher DB2 availability**

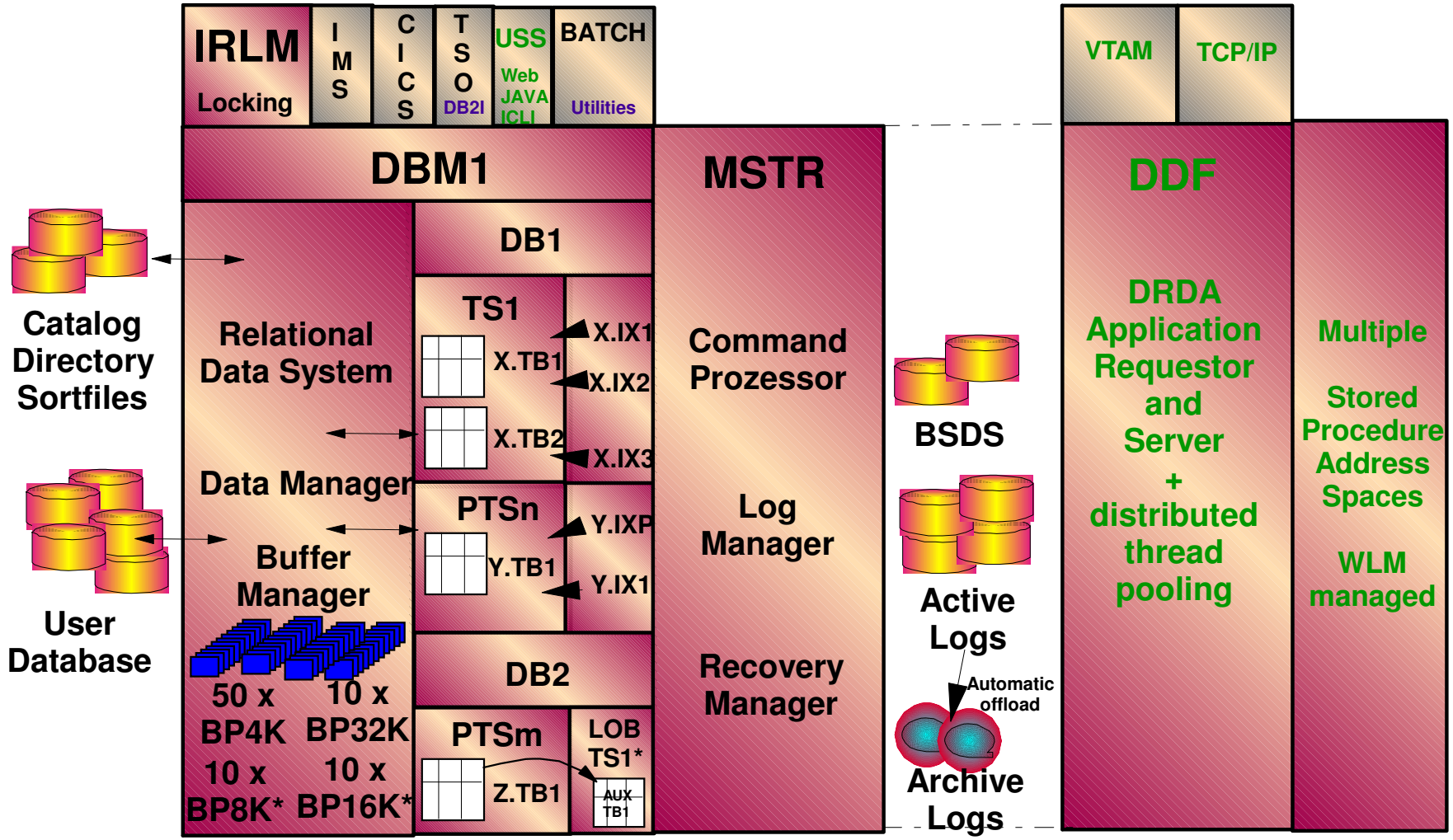


Live Demo

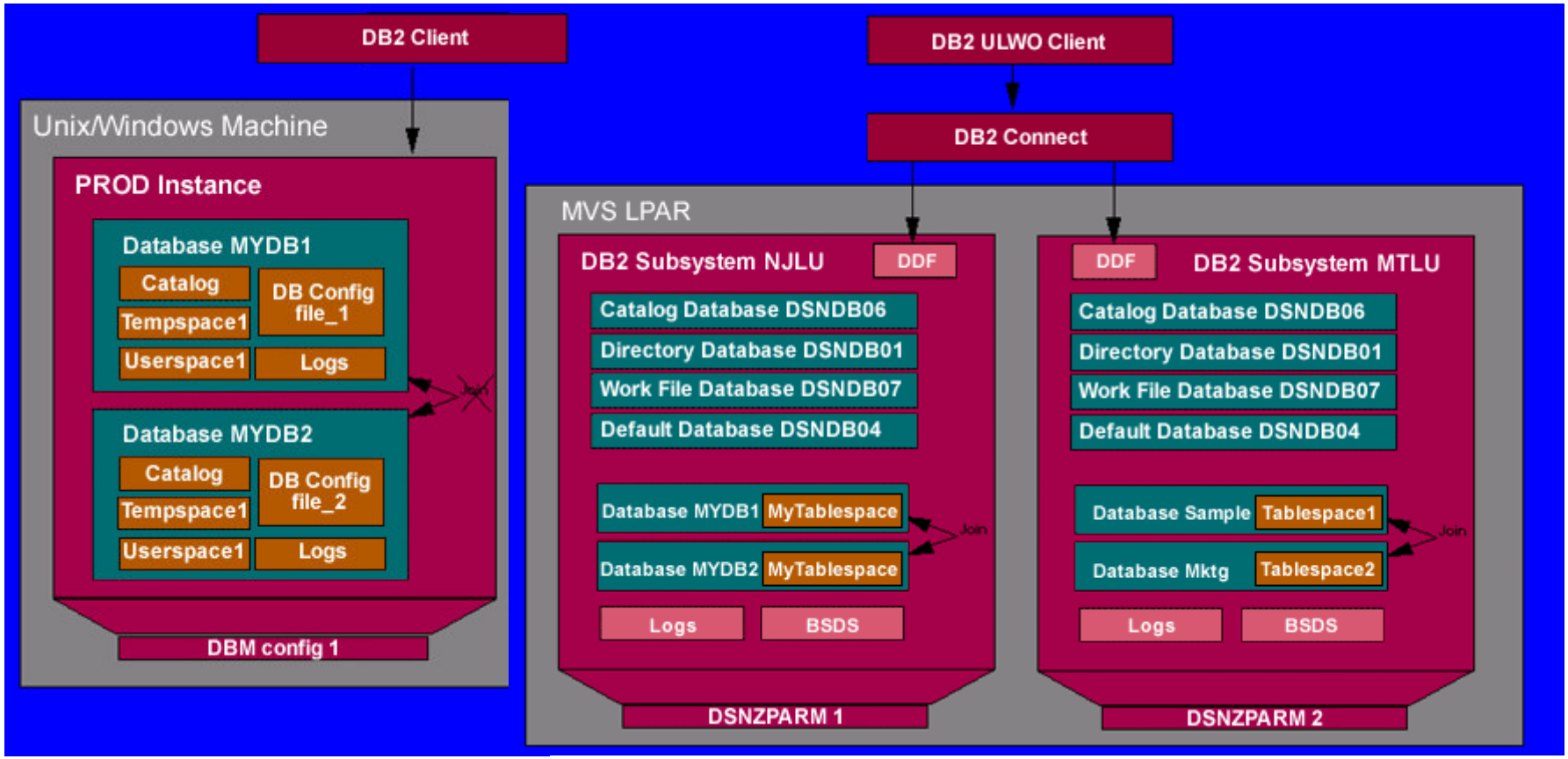
Agenda

- Datenbank-Grundlagen
- DB2 for z/OS Einführung
- **DB2 for z/OS Architektur & Utilities**
- Anwendungsentwicklung
  - Traditionell
  - Java, Web
  - Stored Procedures
- Administration
  - DB2I
  - DB2 Admin Tool
- Trends
  - WebServices
  - XML

DB2 for z/OS Architecture

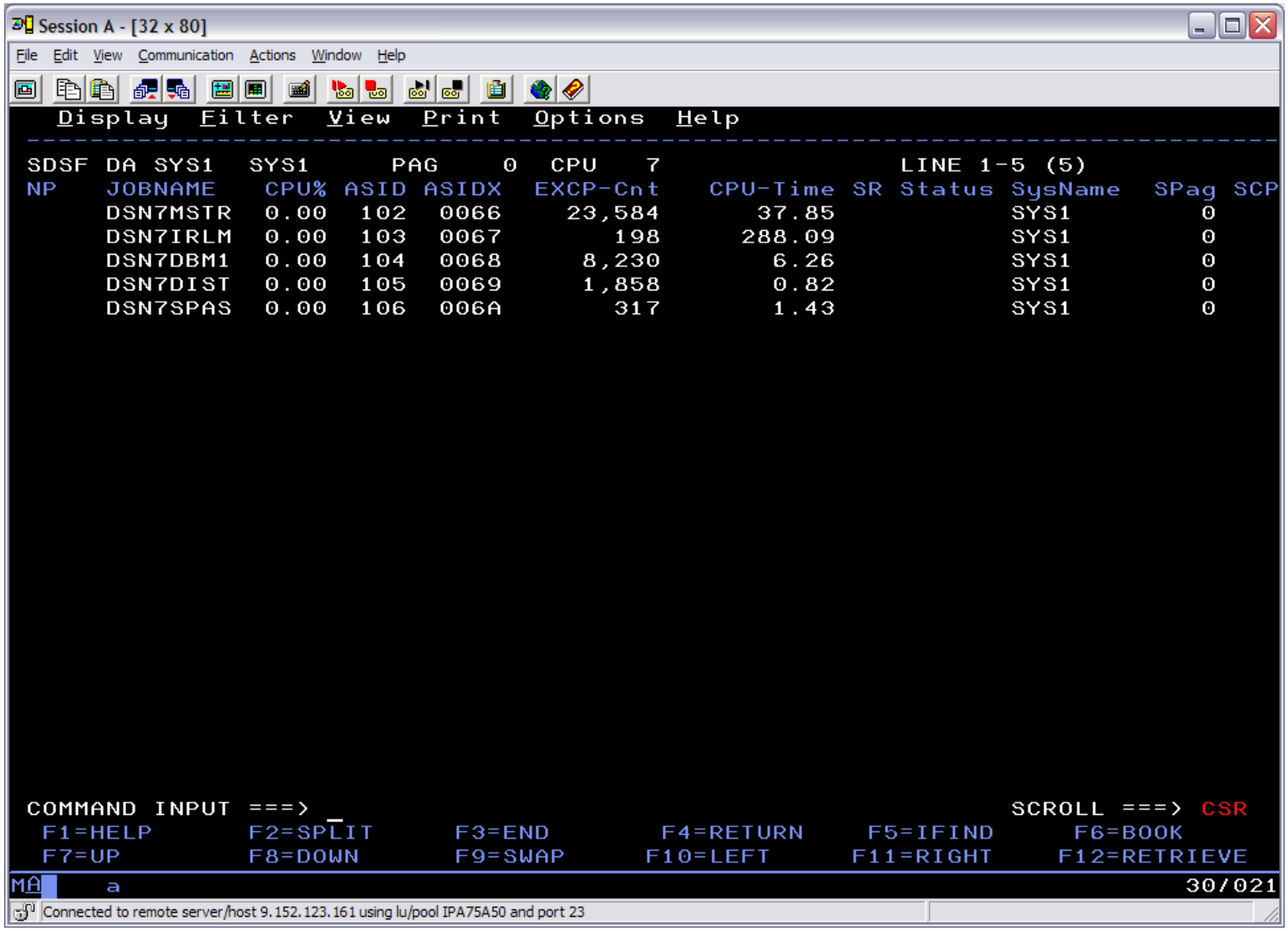


Architecture comparison DB2 LUW vs. DB2 z/OS



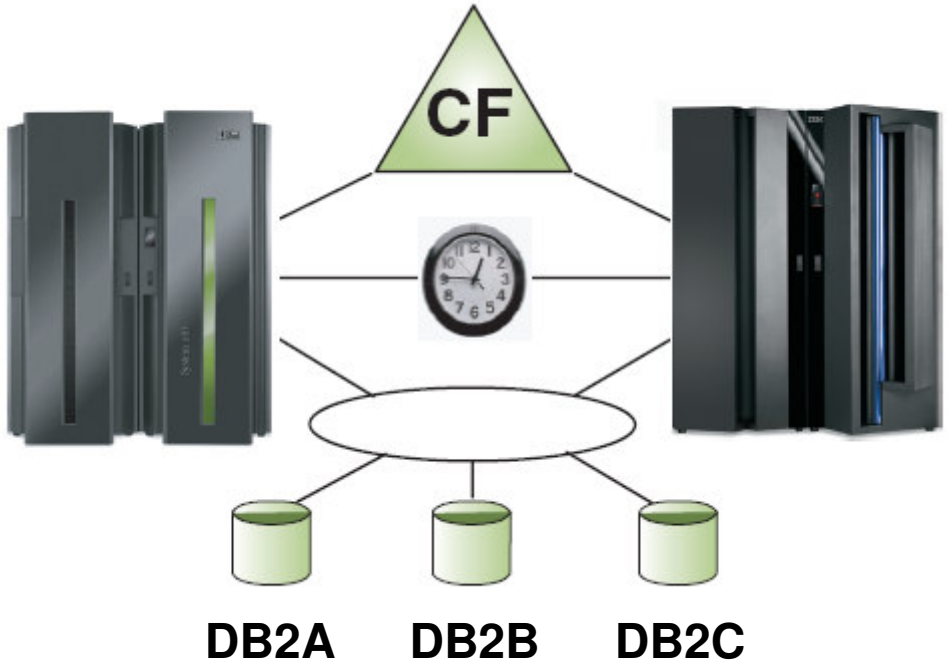
DB2 LUW **DB2 for z/OS**

DB2 components in the Job Entry Subsystem (JES)

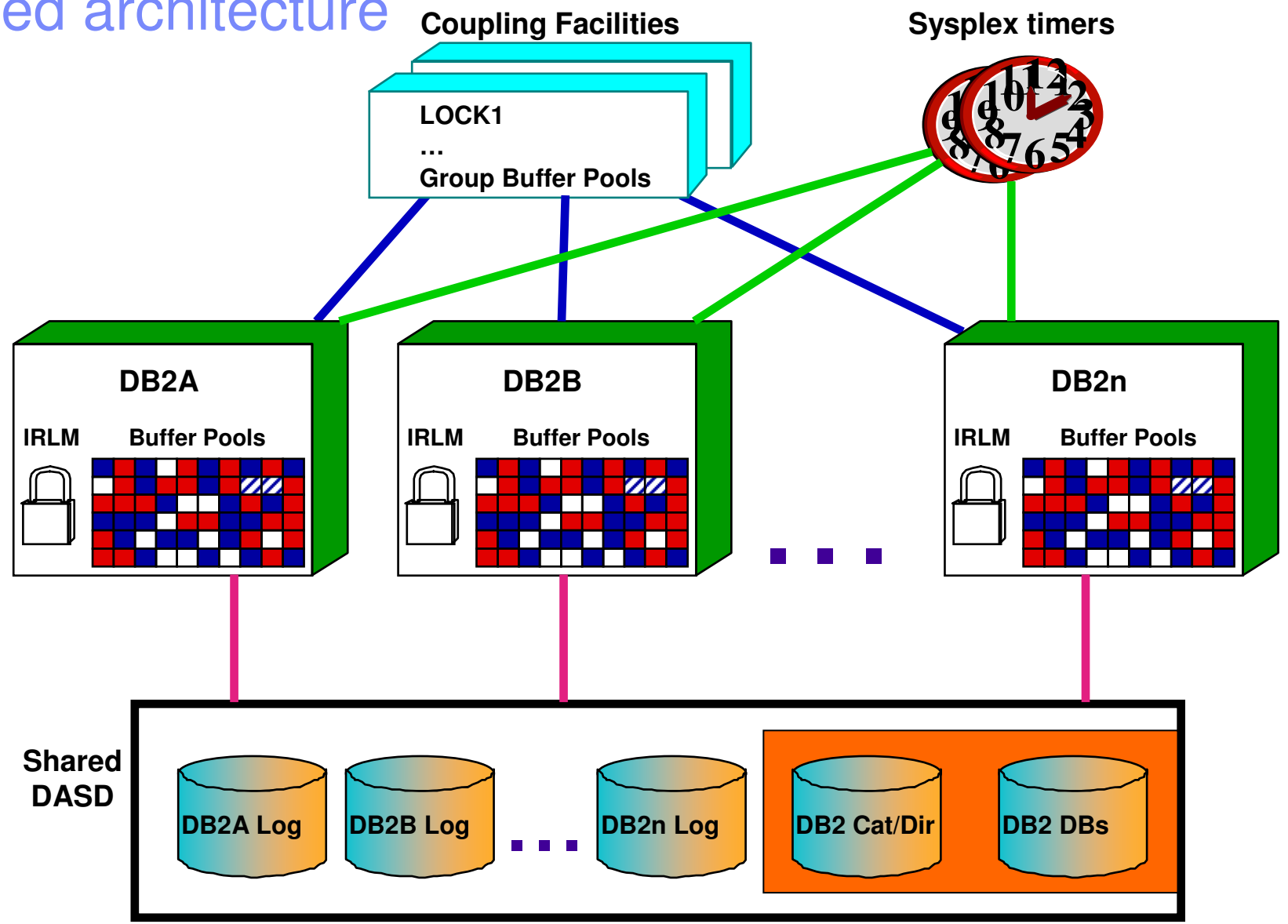


DB2 for z/OS Data Sharing

- Leveraging the **Coupling Facility** technology and **Parallel Sysplex** concept (basicly a cluster of mainframes that act together)
- z/OS Workload Manager allows for highly efficient workload balancing
- Applications connect to the Data Sharing Group
- Significantly increases database performance & availability
- Allows for rolling/non-disruptive migrations and upgrades



Detailed architecture



DB2 Utilities

- Useful service routines that help maintaining and managing DB2 for z/OS
- Are run either as batch jobs or as DB2 stored procedures

- **LOAD/UNLOAD** (Store data into or extract data from DB2 via batch)
- **RUNSTATS** (Generate statistical information about tables and indexes)
- **REORG** (Reorganize the physical allocation of data)
- **COPY/RECOVER** (Generate/recover to backup copies of table spaces)
- **BACKUP/RESTORE SYSTEM** (Backup/restore the whole DB2 subsystem)
- **...**

Agenda

- Datenbank-Grundlagen
- DB2 for z/OS Einführung
- DB2 for z/OS Architektur & Utilities
- **Anwendungsentwicklung**
  - **Traditionell**
  - **Java, Web**
  - **Stored Procedures**
- Administration
  - DB2I
  - DB2 Admin Tool
- Trends
  - WebServices
  - XML

Application development in COBOL with CICS

```
...
SEARCH-BOOK-CATALOG.
    *   DECLARE A CURSOR TO RETURN THE MATCHING ROWS
    *
        EXEC CICS ASKTIME END-EXEC

        EXEC SQL DECLARE BOOKSEARCH CURSOR FOR
                    SELECT ALNAME,TITLE,PRICE,INSTOCK
                    FROM ATCI170.MASTER_WB_DB_BOOK
                    WHERE ALNAME = :SEARCH-AUTHOR OR
                          TITLE = :SEARCH-TITLE
                    ORDER BY TITLE
                    END-EXEC
    *
        EXEC SQL OPEN BOOKSEARCH
                  END-EXEC
    *
                       ...


    *
        EXEC SQL FETCH BOOKSEARCH INTO :AUTHOR-LAST-NAME,
                                :TITLE-OF-BOOK,
                                :PRICE,
                                :INSTOCK
                                END-EXEC
...
```

heise online

06.08.2008 11:04

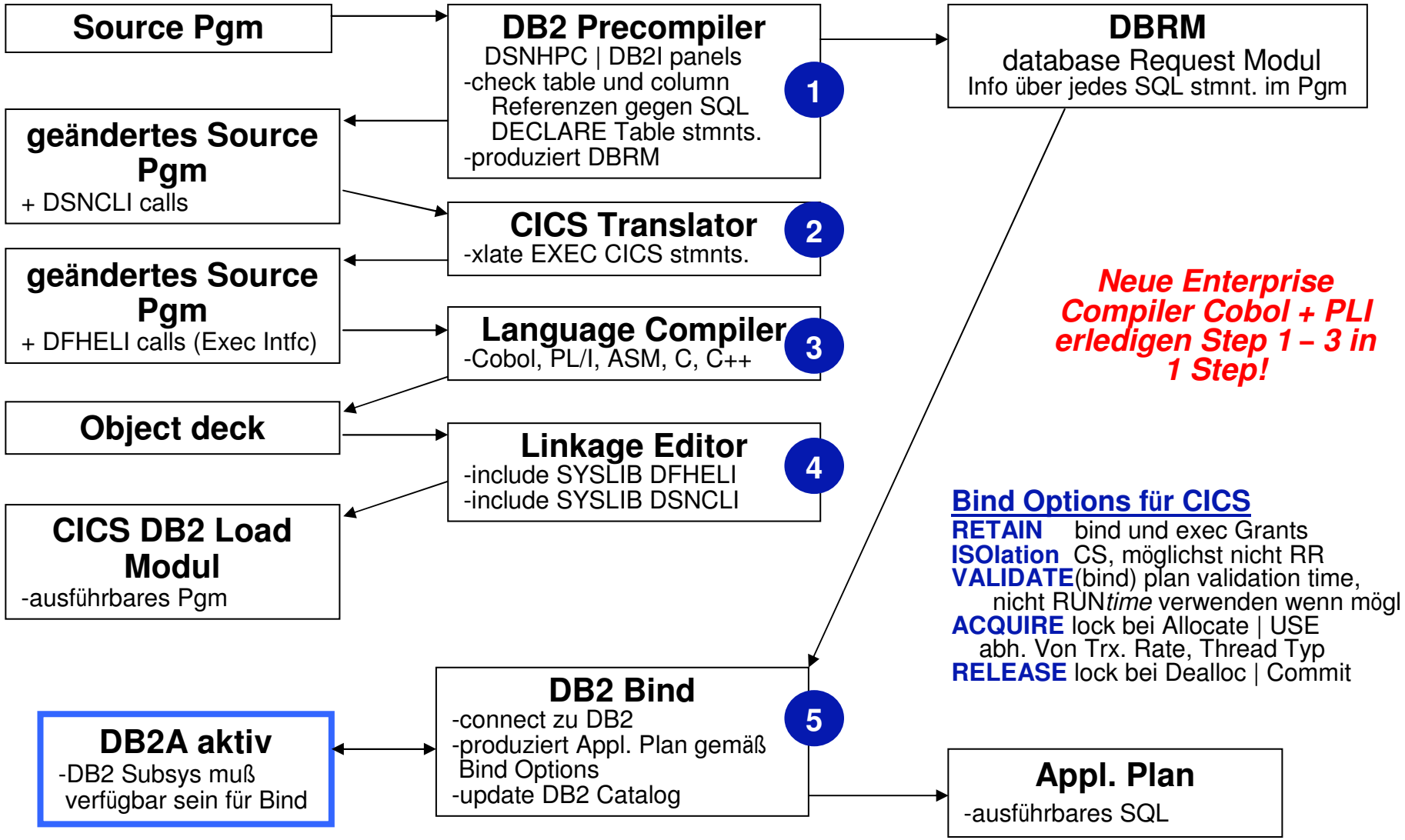
COBOL-Buchhaltungssoftware verzögert kalifornische Gehaltskürzungen

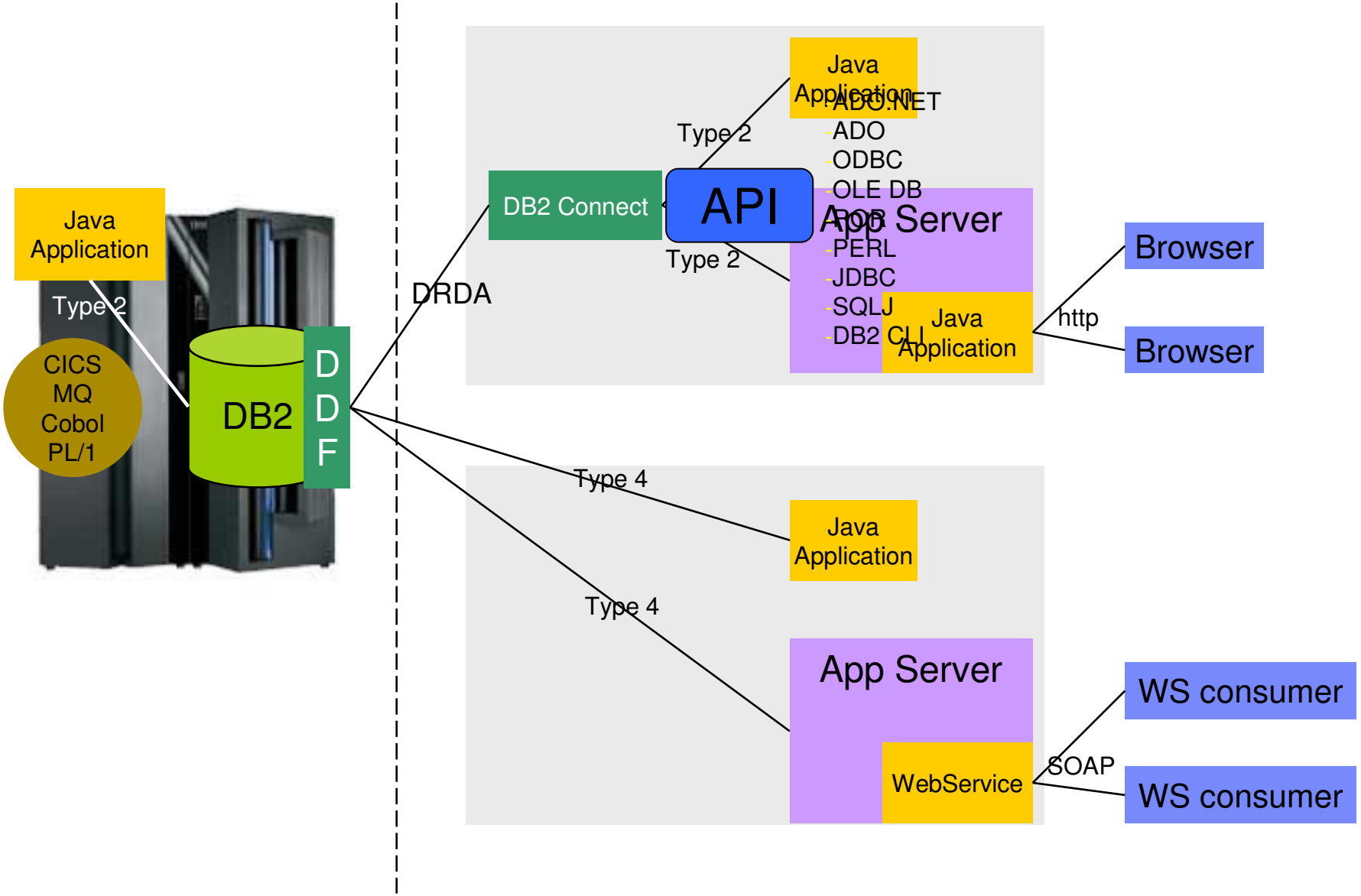
Dem US-Bundesstaat Kalifornien fehlen 17 Milliarden Dollar, die Gouverneur Arnold Schwarzenegger durch übergangsweise Kürzungen des Gehalts auf 6,55 Dollar pro Stunde bei 200.000 Staatsbediensteten sowie Entlassungen wieder reinholen will. Um die drohende Haushaltskrise abzuwenden, sollen die Gehaltsschecks ab dem kommenden Monat also einen erheblich niedrigeren Betrag als bisher ausweisen. Doch das ist schwieriger, als man denkt. Denn die Lohnbuchhaltung ist in COBOL geschrieben, der (wohl zu Recht als antiquiert angesehenen) Programmiersprache für betriebswirtschaftliche Anwendungen (COmmon Business Oriented Language), berichtet die lokale Tageszeitung **Sacramento Bee[1]**.

Noch im Beruf stehende COBOL-Experten muss man mit der Lupe suchen. Schwarzenegger wird sie wohl aus der Rente holen müssen. Im Übrigen krankt COBOL-Code typischerweise an fehlender oder lückenhafter Dokumentation. Wohl auch deshalb werde die Umstellung des Abrechnungssystems mehrere Monate beanspruchen, schreibt der demokratische Schatzmeister

For example, Cobol remains the most widely deployed programming language in big business, accounting for 75% of all computer transactions – and it is not going to go away. Cobol is pervasive in the financial sector (accounting for 90% of all financial transactions), in defence, as well as within established manufacturing and insurance sectors. We estimate that there are over 200 billion lines of Cobol in production today, and this number continues to grow by between three and five percent a year.

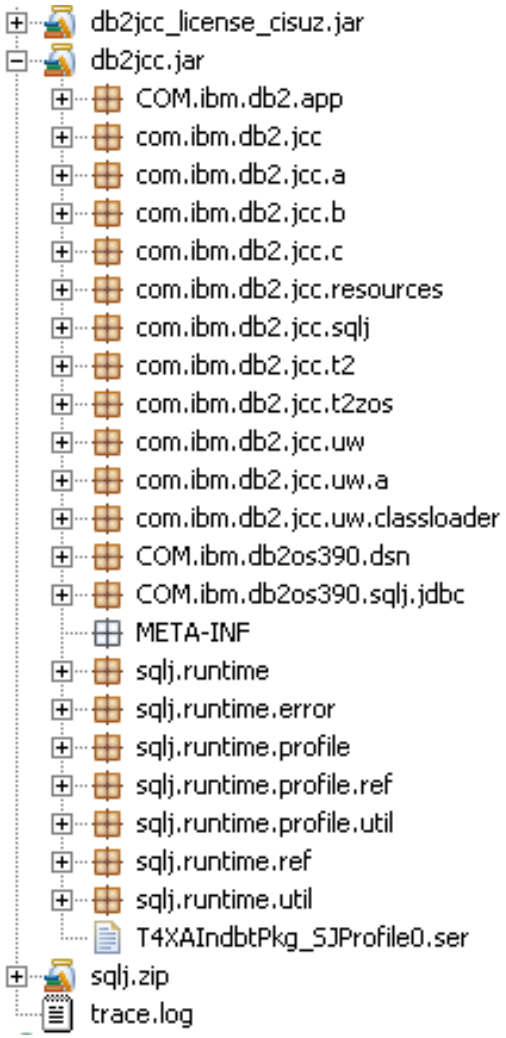
Preparing a CICS DB2 program





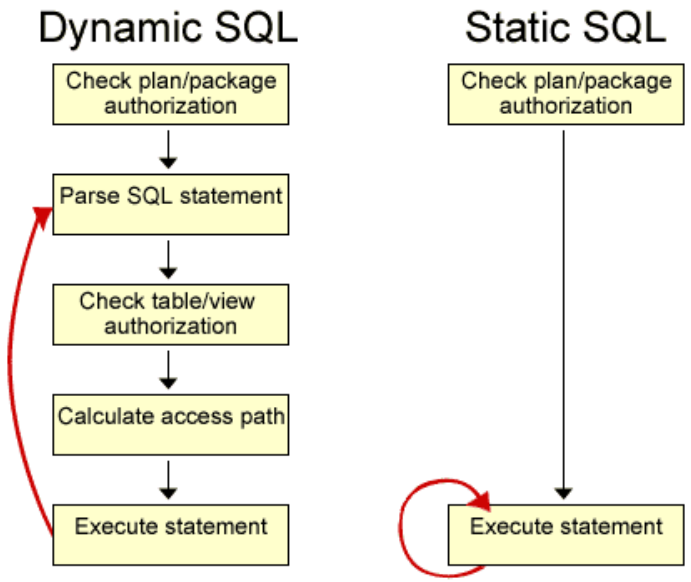
DB2 Universal Driver

- One driver for DB2 family
- Also known as Java Common Client, Combined Client, JCC driver
- Support for JDBC 3.0 and SQLJ 3.0
- Type 2 and Type 4 drivers
  - Type4 URL: `jdbc:db2://SERVER:PORT/DATABASE`
  - Type2 URL: `jdbc:db2:DATABASE`
- Improved deployment and ease of use
- Integration with WebSphere



Dynamic SQL vs. Static SQL

- SQLJ
  - More concise syntax
  - Easier to code/understand
  - Better security/authorization characteristics
  - Better performance
  - Accounting



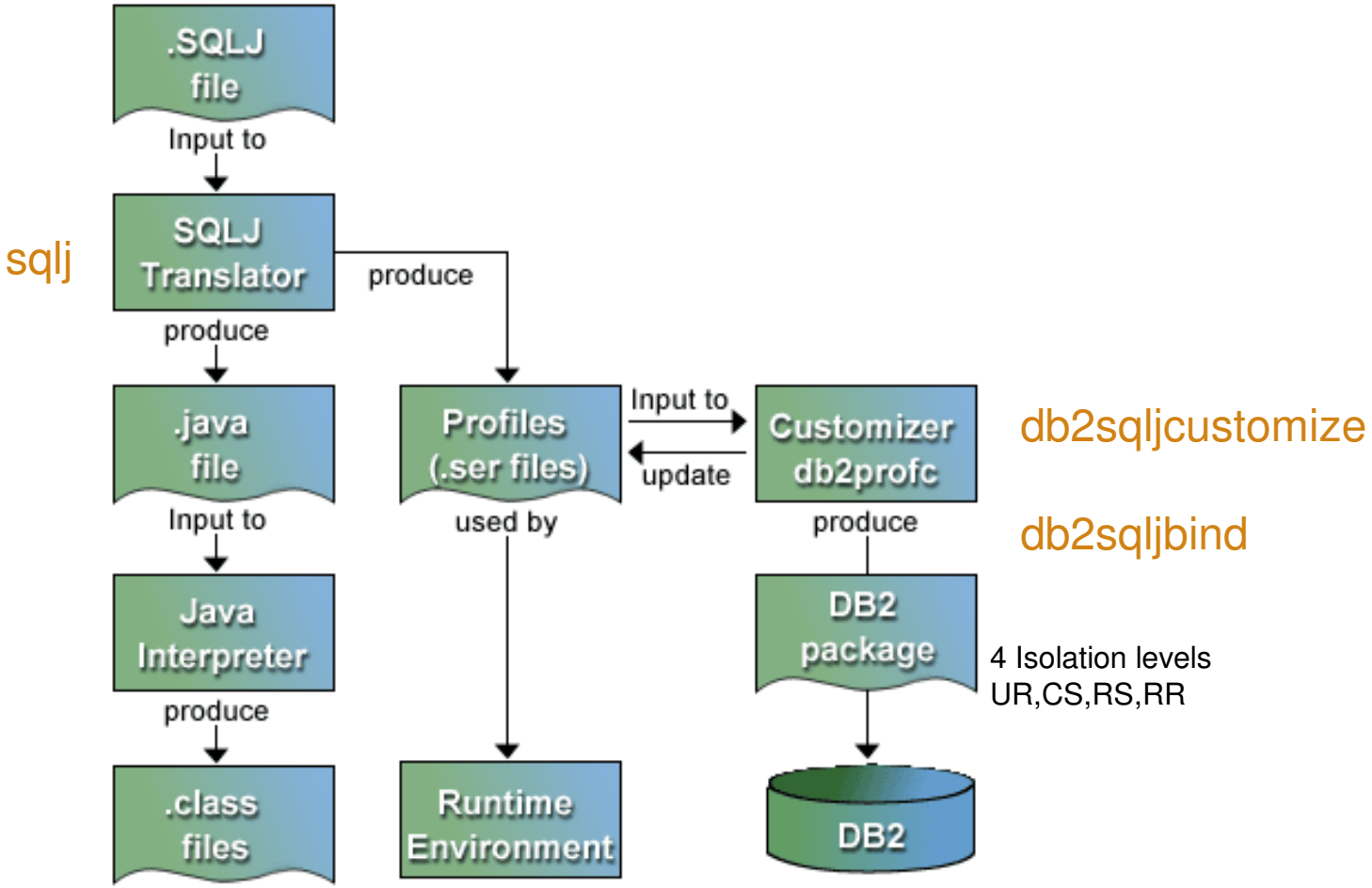
- JDBC
  - Widely known, integrated in IDEs
  - Most Tooling uses JDBC
  - Less application development

- Integrated development environment for both SQLJ and JDBC
  - Significant improvements with IBM Data Studio and pureQuery

JDBC - correct usage

```
DB2Connection conn = null;
PreparedStatement stmt = null;
ResultSet rs = null;
...
        // Prepare the SELECT statement.
        stmt = conn.prepareStatement("SELECT LASTNAME, FIRSTNME, SALARY"
                + " FROM SYSADM2.EMPLOYEE" + " WHERE SALARY BETWEEN ? AND ?");
        // Set parameters for the SELECT statement.
        stmt.setBigDecimal(1, new BigDecimal(30000));
        stmt.setBigDecimal(2, new BigDecimal(50000));

        // Execute the query to retrieve a ResultSet.
        rs = stmt.executeQuery();
        // Iterate over the ResultSet.
        while (rs.next()) {
                String lastname = rs.getString(1); // LASTNAME
                String firstname = rs.getString(2); // FIRSTNME
                System.out.println(lastname + ", "+ firstname);
        ...
        if (rs != null)   rs.close();
        if (stmt != null) stmt.close();
        if (conn != null) conn.close();
        ...
```

SQLJ – from code to package



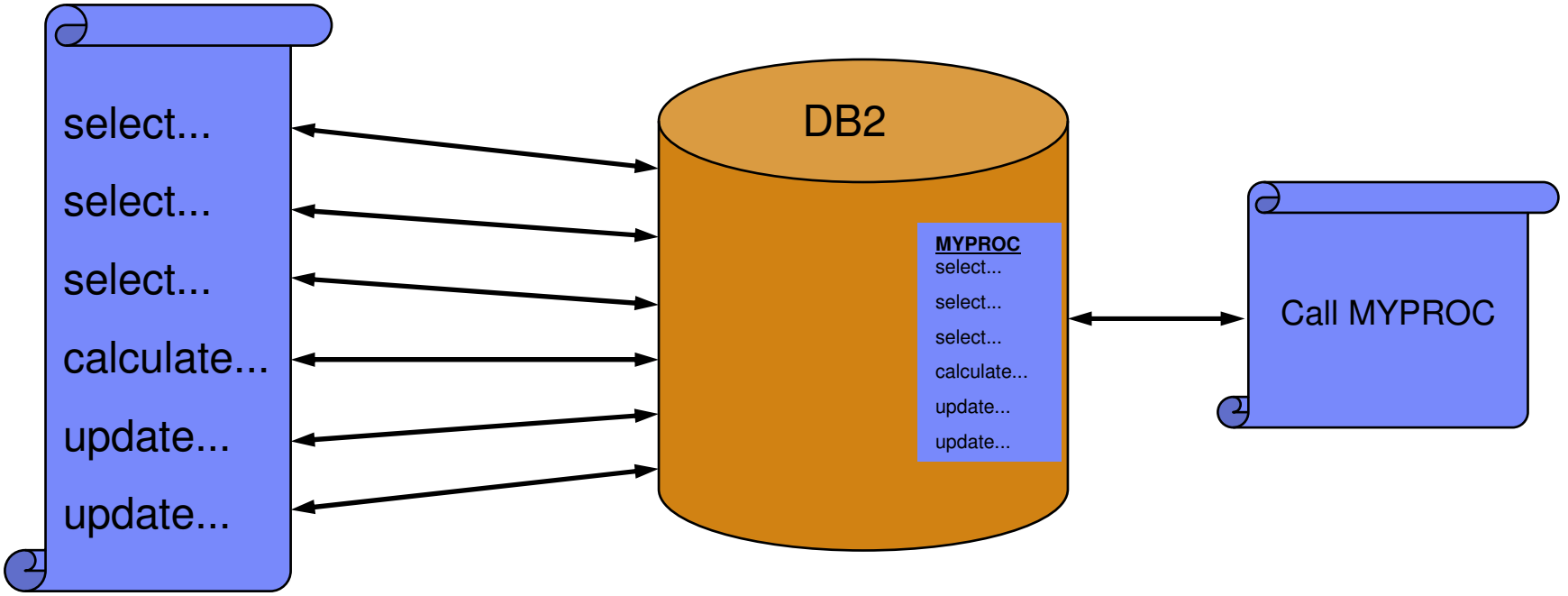
fully integrated into IDEs like WSAD, RAD, ...

SQLJ sample SimpleSQLJ.sqlj

```
#sql iterator EmpRecs( String firstnme, String lastname);
...
DB2Connection con = null;
DefaultContext ctx = null;
int lowSalary = 30000;
int highSalary = 50000;

...
        // Set the default context
        ctx = new DefaultContext(con);
        DefaultContext.setDefaultContext(ctx);

        EmpRecs employees;
        #sql [ctx] employees = { SELECT firstnme, lastname, salary
                FROM EMPLOYEE WHERE salary BETWEEN :lowSalary and :highSalary};
        // Go through ResultSet
        while (employees.next()) {
        System.out.println(employees.firstnme() + " "
                        + employees.lastname() + " " +employees.salary());}
...
ctx.close();
con.close();
```

Stored Procedures

- DB program physically stored in the DBMS
- Contain a business logic
- Avoids network traffic
- Used for performance improvements
- Simplifies application development



Types of Stored Procedures

- Cobol, PL/I
- SQL
- Java (JDBC / SQLJ)

```
public static void getuuid ( String[] uuid ) throws SQLException, Exception {
        // Set return parameter
        uuid[0] = UUID.randomUUID().toString();
}
public static String newUUID (  ) {
}
```

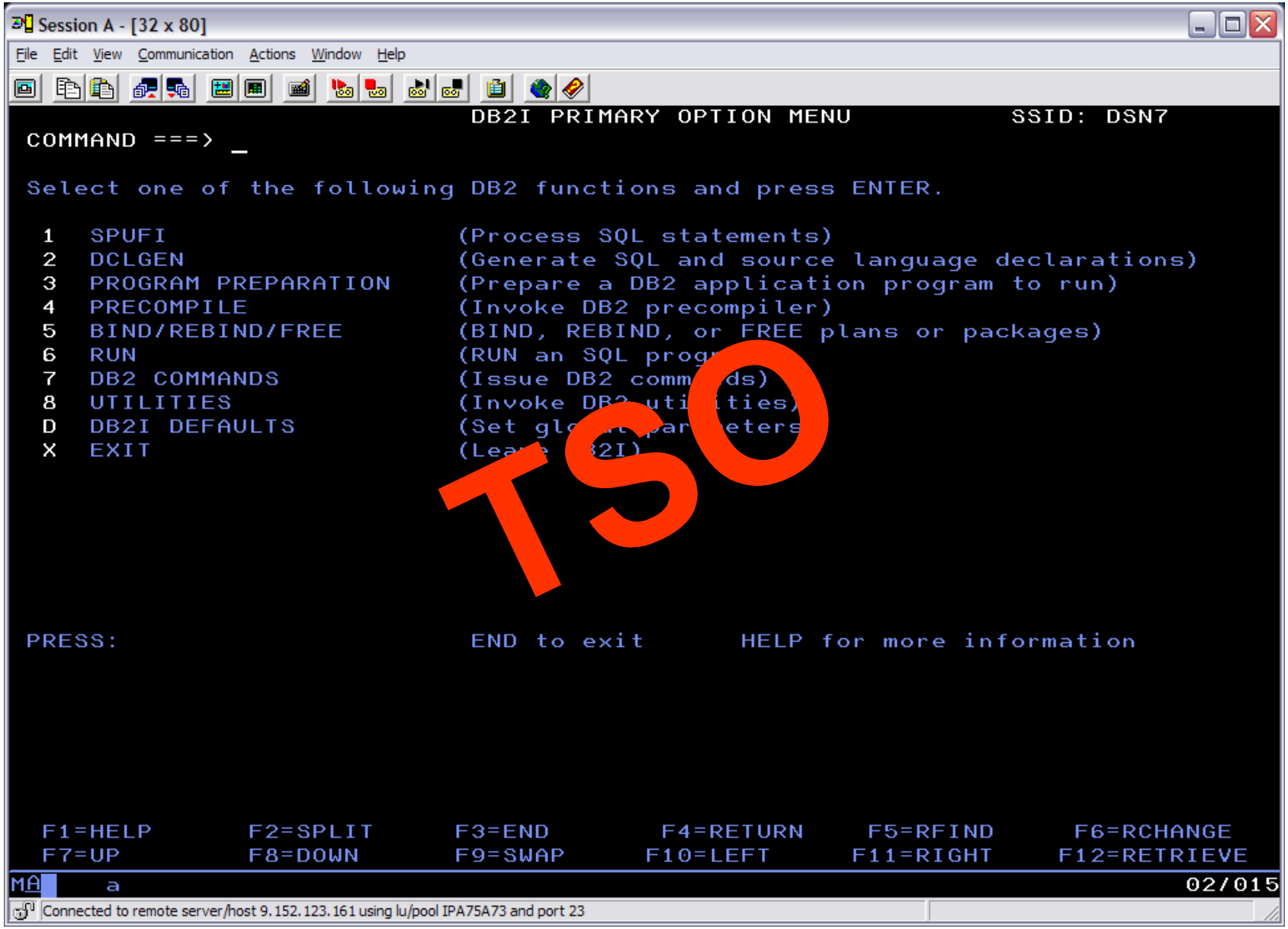
```
CREATE PROCEDURE "USER"."GETUUID" (OUT UUID VARCHAR(40))
EXTERNAL NAME '"USER"."GETUUID":user.Getuuid.getuuid'
LANGUAGE JAVA
MODIFIES SQL DATA
PARAMETER STYLE JAVA
FENCED
WLM ENVIRONMENT DSN7JSPC

GRANT EXECUTE ON PROCEDURE "USER"."GETUUID" TO "USER" WITH
GRANT OPTION
```

Agenda

- Datenbank-Grundlagen
- DB2 for z/OS Einführung
- DB2 for z/OS Architektur & Utilities
- Anwendungsentwicklung
  - Traditionell
  - Java, Web
  - Stored Procedures
- **Administration**
  - **DB2I**
  - **DB2 Admin Tool**
- Trends
  - WebServices
  - XML

DB2 Administration

- DB2 Interactive
  - SPUFI (SQL processing using file input)
  - Program preparation
  - Bind
  - DB2 Commands
  - Utilities (LOAD, REORG, RECOVER, RUNSTATS...)
- DB2 Administration Tool
  - Browse DB2 System Catalog
  - Explain
  - DDL creation, GEN, Authorizations
  - Utility Generation
- Batch administration
- ISV Tools

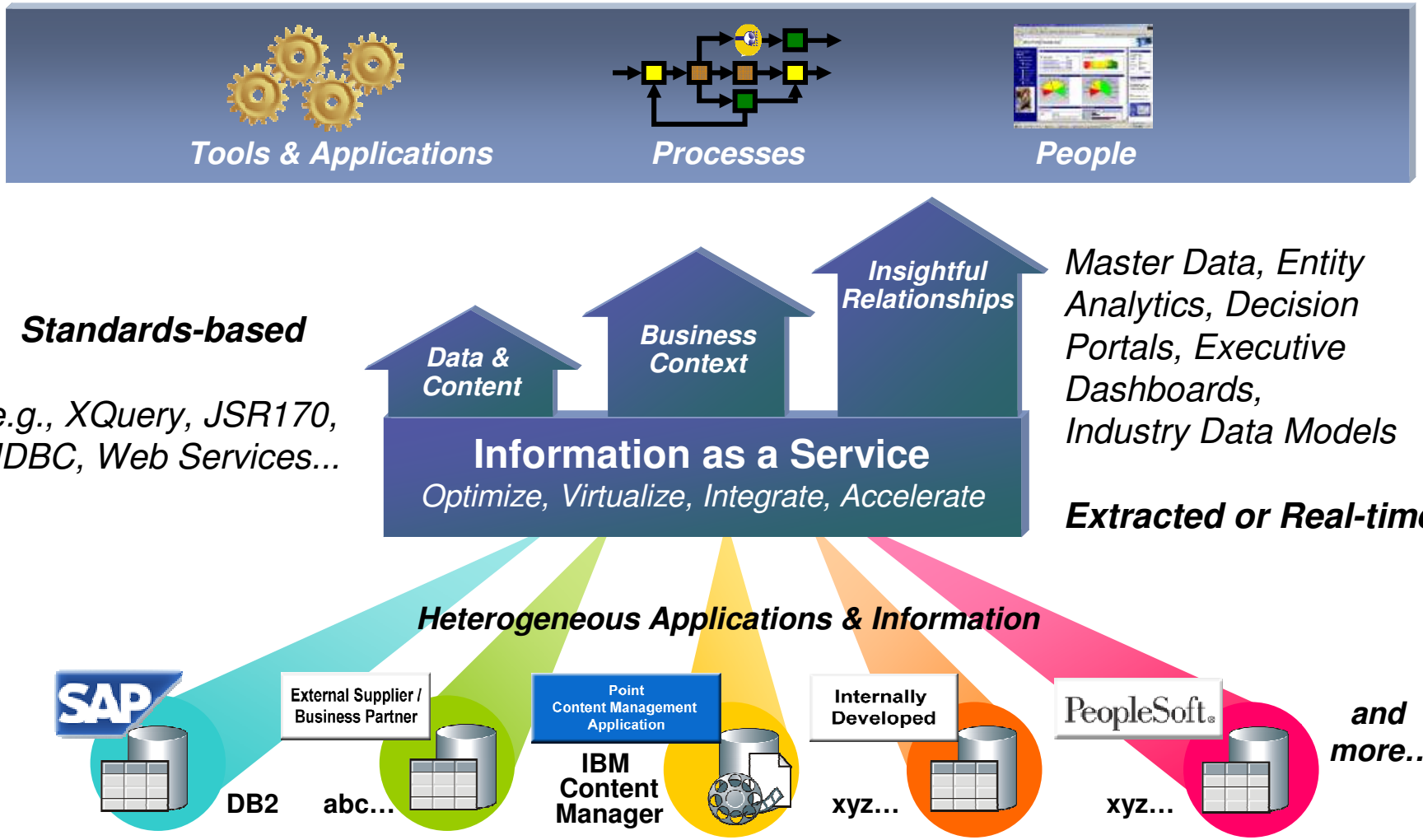


Agenda

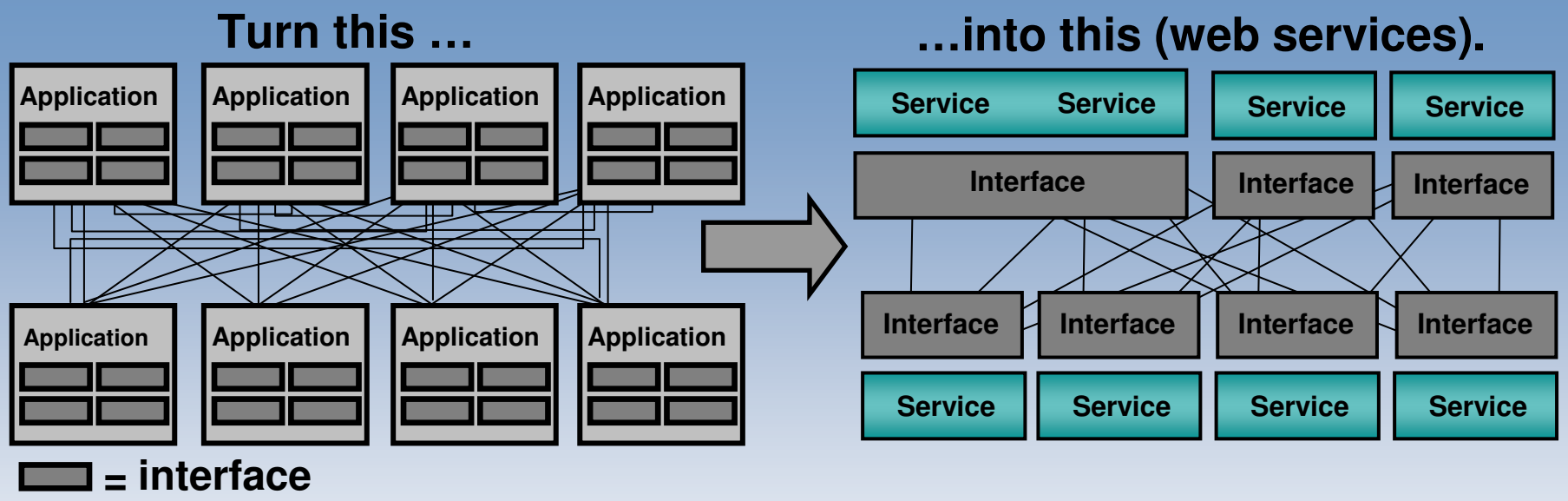
- Datenbank-Grundlagen
- DB2 for z/OS Einführung
- DB2 for z/OS Architektur & Utilities
- Anwendungsentwicklung
  - Traditionell
  - Java, Web
  - Stored Procedures
- Administration
  - DB2I
  - DB2 Admin Tool
- **Trends**
  - **WebServices**
  - **XML**

Information as a Service

Moving From a Project-Based to a Flexible Architecture (SOA)



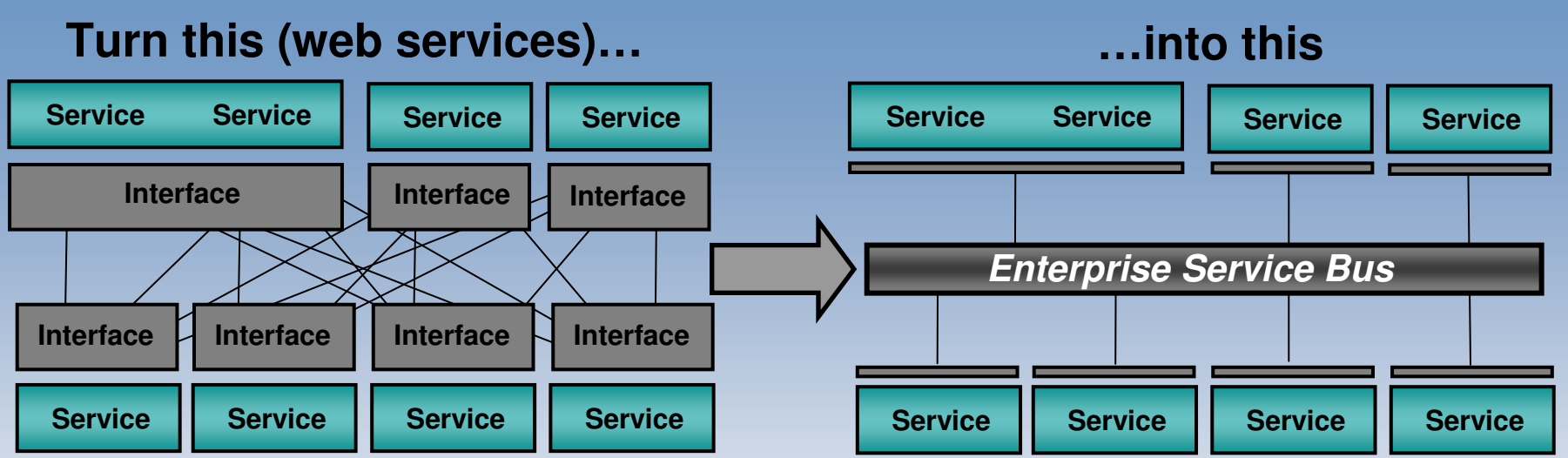
WebServices decouple interfaces from applications



- ✓ Rich business abstractions describe the application interface
- ✓ Decouples the interfaces from the business applications
- ✓ The number and complexity of the interfaces is reduced
- ✓ Business applications and their interfaces become reusable

But separate connection points still leaving bloated interfaces ….

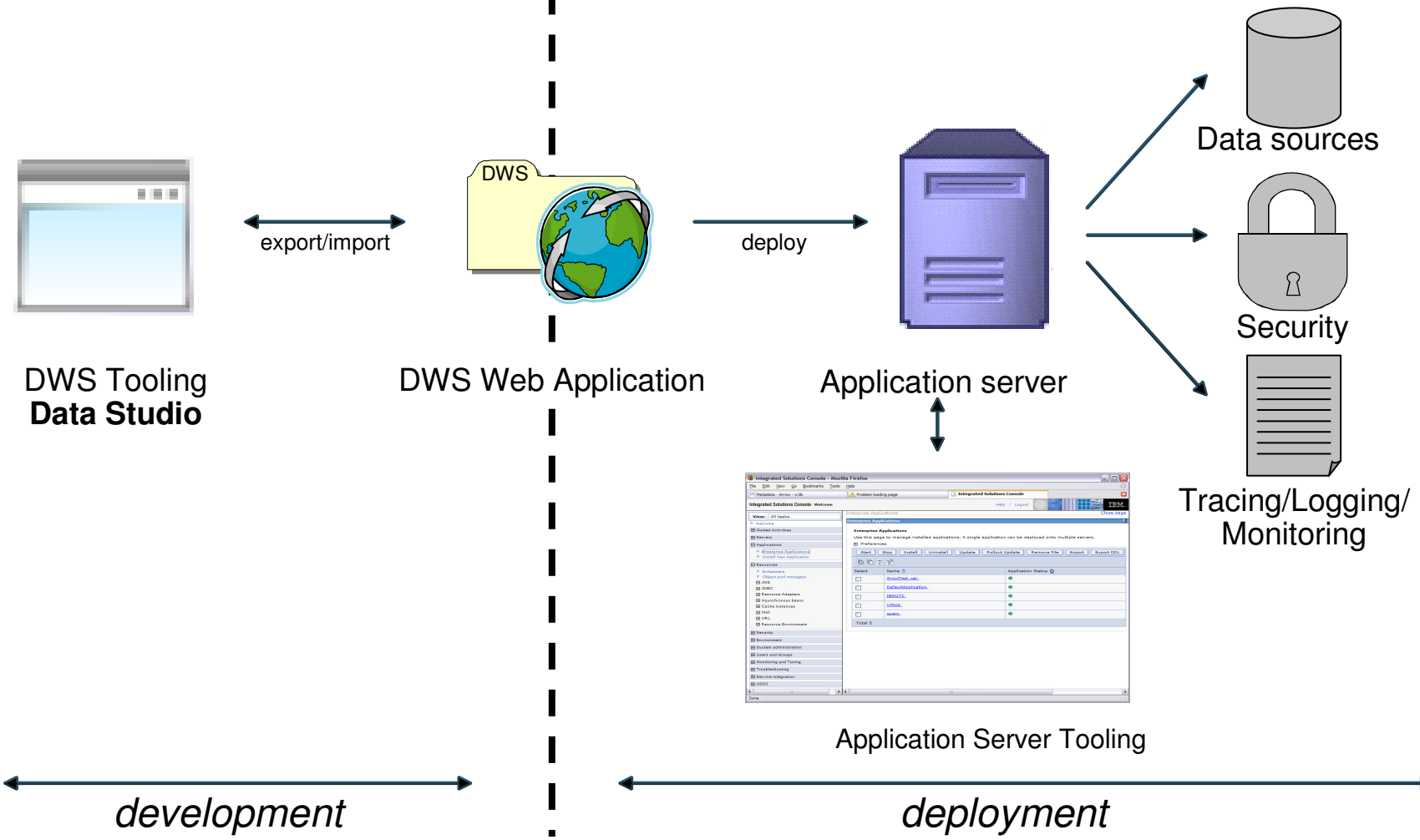
The ESBus shrinks those interfaces further



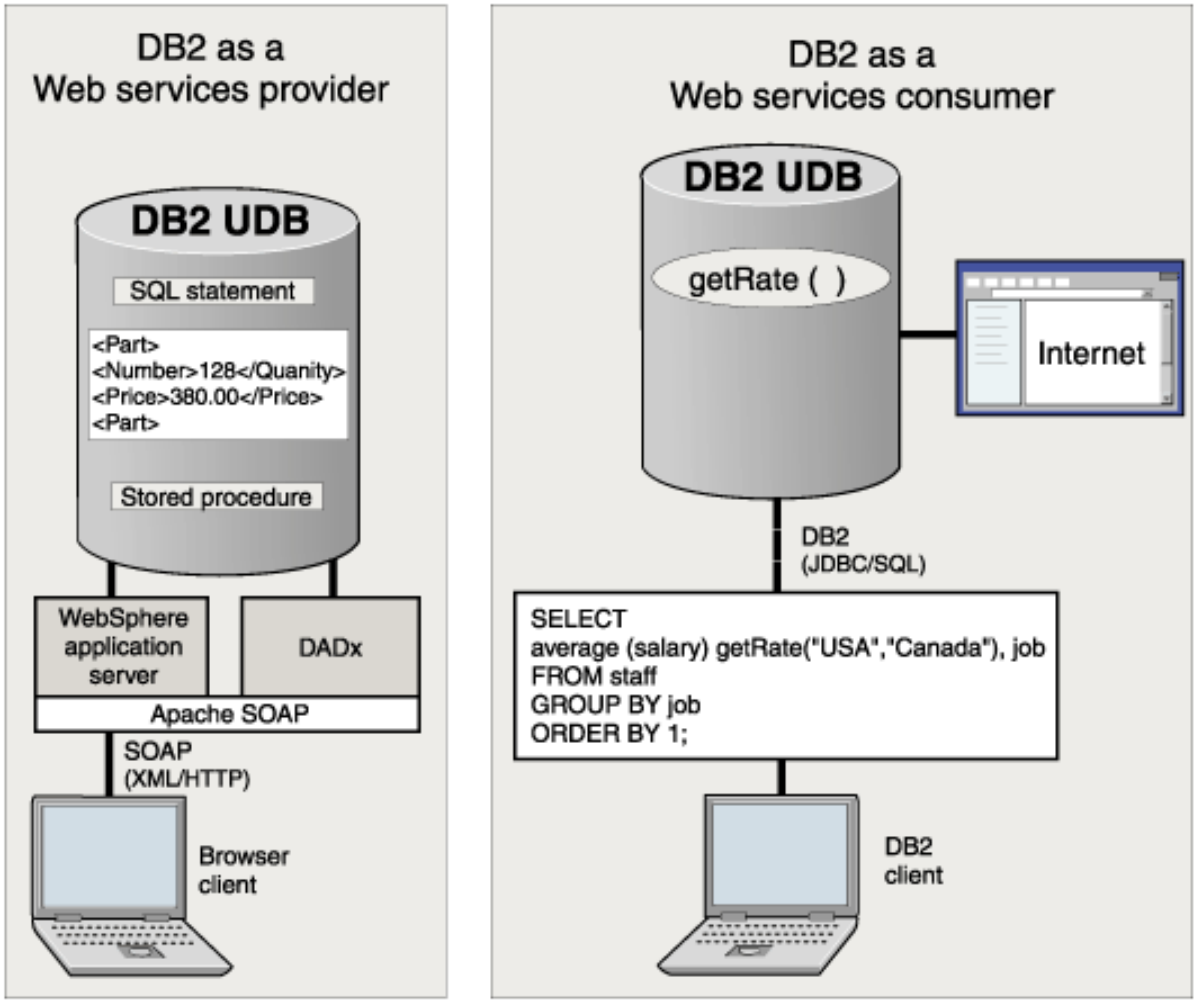
- ✓ Decouples the point-to-point connections from the interfaces
- ✓ Allows for dynamic selection, substitution, and matching
- ✓ Enables more flexible coupling and decoupling of the applications
- ✓ Enables you to find both the applications and the interfaces for re-use

Development and Deployment

Certain app-server specific settings required



DB2 and WebServices



Data storage

Only main content

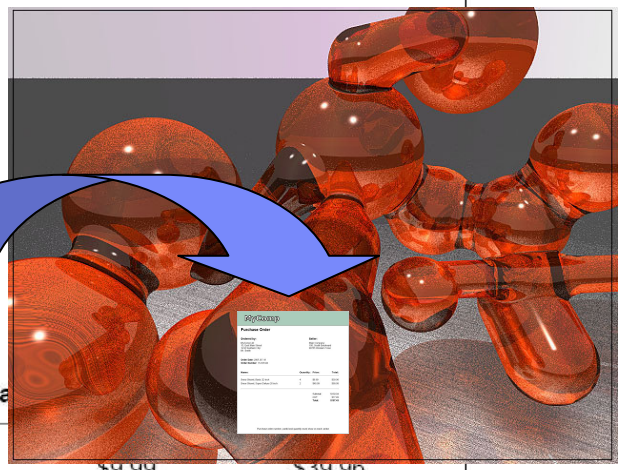
MyComp

Purchase Order

Ordered by:

MyComp Ltd
10, East Main Street
1234 Southern City
Mr. Smith

Order Date: 2007-07-18
Order Number: 11223344



Order

OrderNr	CustomerID	OrderDate
11223344	98765	2007-07-18

OrderDetails

OrderNr	Item	Quantity	Price
11223344	Snow Shovel, Basic 22 inch	4	9.99
11223344	Snow Shovel, Super Deluxe 26 inch	2	49.99

Customer

CustomerID	Name	Street	City
98765	MyComp Ltd	10, East Main Street	1234 Southern City

Qua

4	$9.99	$39.96
2	$49.99	$99.98

Subtotal:	$139.94
GST:	$17.49
Total:	**$157.43**

Purchase order number, partid and quantity must show on each carton

Black box

XML advantages

Optional content possible

No fixed amount of lines

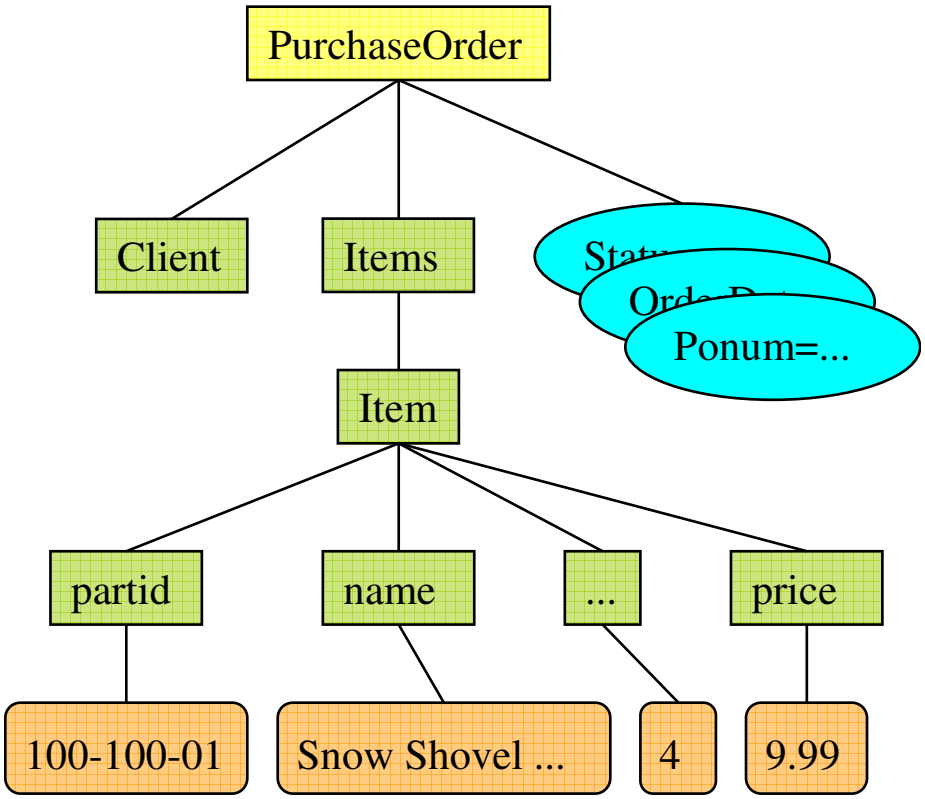
Can keep data types

Even storing comment lines

```
<PurchaseOrder PoNum="11223344"
               OrderDate="2007-07-18" Status="Unshipped">
   <client>
     <companyname> MyComp Ltd </companyname>
     <address>
        <line> 10, East Main Street </line>
        <line> 1234 Southern City </line>
        <line> Mr. Smith </line>
     </address>
  </client>
  <items>
     <item>
        <partid> 100-100-01 </partid>
        <name> Snow Shovel, Basic 22 inch </name>
        <quantity> 4 </quantity>
        <price> 9.99 </price>
     </item>
     <item>
        <partid> 100-103-01 </partid>
        <name> Snow Shovel, Super Deluxe 26 inch </name>
        <quantity> 2 </quantity>
        <price> 49.99 </price>
     </item>
  </items>
  <comment>
  Purchase order number, partid and
  quantity must show on each carton
  </comment>
</PurchaseOrder>
```

```
<PurchaseOrder PoNum="11223344"
               OrderDate="2007-07-18" Status="Unshipped">
   <client>
     <companyname> MyComp Ltd </companyname>
     <address>
        <line> 10, East Main Street </line>
        <line> 1234 Southern City </line>
        <line> Mr. Smith </line>
     </address>
   </client>
   <items>
     <item>
        <partid> 100-100-01 </partid>
        <name> Snow Shovel, Basic 22 inch </name>
        <quantity> 4 </quantity>
        <price> 9.99 </price>
     </item>
     <item>
        <partid> 100-103-01 </partid>
        <name> Snow Shovel, Super Deluxe 26 inch </name>
        <quantity> 2 </quantity>
        <price> 49.99 </price>
     </item>
   </items>
   <comment>
   Purchase order number, partid and
   quantity must show on each carton
   </comment>
</PurchaseOrder>
```

Parsing



Serialization

Document Node

Element Node

Text Node

Attribute Node

Why XML in databases?

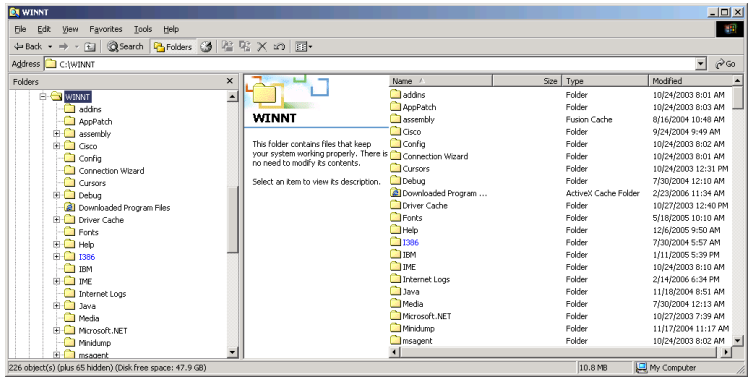
- **Easy to share XML between applications, businesses, processes, …**

→ SOA

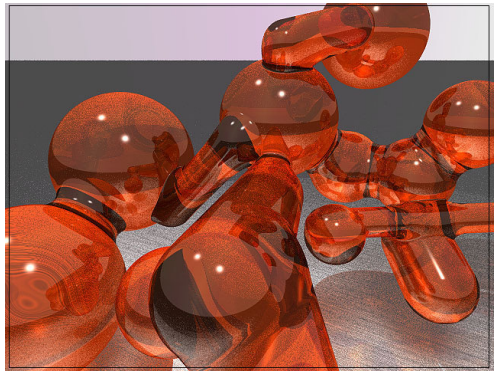
- **Flexibility, Flexibility**
- **Easy to extend**
- **XML is vendor and platform independent**
- **XML is self-describing**
- **Easy to transform XML documents into other formats (HTML, etc.)**

- **Managing large volumes of XML data is a DB problem!**
  - Efficient Search & Retrieval of XML
  - Persistency, Recovery, Transactions, ACID
  - Performance, Scalability
  - *…all the same reasons as for relational data!*

Where was XML in the past?



In files…
- Storage not managed and not secure



In LOBS…
- Content and business value locked up

Shred to tables
- Complex and fragile mapping

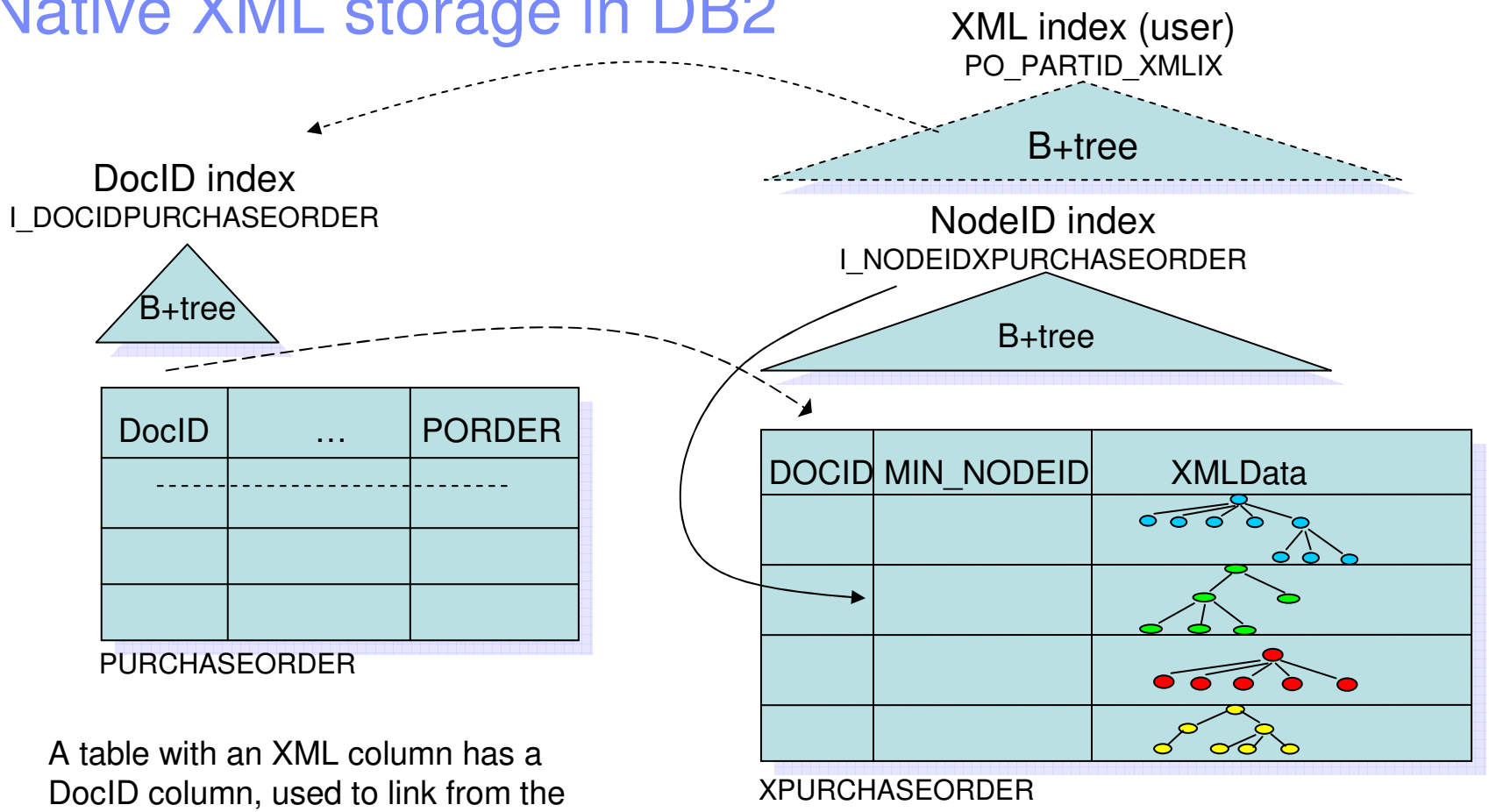
XML DB
- Scalability & integration concerns

XML in DB2 for z/OS

- Build-in datatype, with no design limit in size
- Allows for well-formed XML documents
  - E.g. Start tag: <...> and end tag: </...>
  - Parsing is done by z/OS XML parser
- Register schemas in XML Schema Repository (XSR) and use DSN_XMLVALIDATE in SQL for validation of XML
- XML Value Index – user defined index for document in XML column, used to improve the query performance on XML document

```
CREATE TABLE PURCHASEORDER (
    POID       BIGINT NOT NULL,
    STATUS     VARCHAR(10) NOT NULL WITH DEFAULT 'UNSHIPPED',
    CUSTID     BIGINT,
    ORDERDATE DATE,
    PORDER     XML )
PARTITION BY RANGE
(POID ASC)
(PARTITION 1 ENDING ( 00223344 ) ,
 PARTITION 2 ENDING ( 11223344 ) ,
 PARTITION 3 ENDING ( 22334455) )
IN XMLDB.PORDER ;
```

Native XML storage in DB2



A table with an XML column has a DocID column, used to link from the base table to the XML table.
A DocID index is used for getting to base table rows from XML indexes.

Each XMLData column is a VARBINARY, containing a subtree or a sequence of subtrees, with context path.
Rows in XML table are freely movable, linked with a NodeID index.

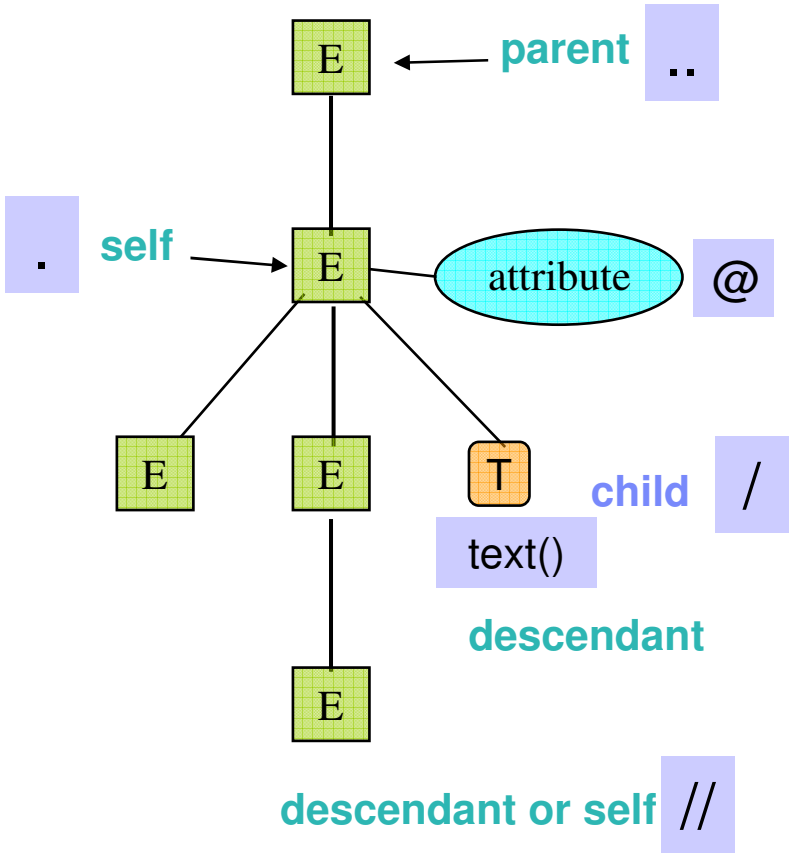
XML Indexes

```
CREATE        INDEX PO_PARTID_XMLIX
ON PURCHASEORDER(PORDER)
GENERATE KEY USING XMLPATTERN
'/PurchaseOrder/items/item/partid'
AS SQL VARCHAR(30);
```

```
<PurchaseOrder PoNum="11223344"
               OrderDate="2007-07-18" Status="Unshipped">
   <client>
     ...
  </client>
  <items>
     <item>
        <partid> 100-100-01 </partid>
        <name> Snow Shovel, Basic 22 inch </name>
        <quantity> 4 </quantity>
        <price> 9.99 </price>
     </item>
     <item>
        <partid> 100-103-01 </partid>
        <name> Snow Shovel, Super Deluxe 26 inch </name>
        <quantity> 2 </quantity>
        <price> 49.99 </price>
     </item>
  </items>
  <comment>
  ...
  </comment>
</PurchaseOrder>
```

XPATH Axis-Navigation Direction

- **ForwardAxis:**
  - child
  - descendant
  - self
  - attribute
  - text
- **ReverseAxis**:
  - parent



XPATH Expressions

XMLQUERY

POID	...	CUSTID	PORDER
11223344	...	1002	<PurchaseOrder PoNum="11223344" OrderDate="2007-07-18" Status="Unshipped"> ... <items> <item> <partid>100-100-01</partid> <name>Snow Shovel, Basic 22 inch</name> <quantity>4</quantity> <price>9.99</price> </item> ... </PurchaseOrder>
11223345	...	1002	<PurchaseOrder PoNum="11223345" OrderDate="2007-07-18" Status="Unshipped"> ... <items> <item> <partid>100-100-01</partid> <name>Snow Shovel, Basic 22 inch</name> <quantity>4</quantity> <price>9.99</price> </item> ... </PurchaseOrder>
11223346	...	1005	<PurchaseOrder PoNum="11223346" OrderDate="2007-07-18" Status="Unshipped"> ... <items> <item> <partid>100-103-01</partid> <name>Snow Shovel, Super Deluxe 26 inch</name> <quantity>4</quantity> <price>9.99</price> </item> ... </PurchaseOrder>

1. position to row according to WHERE clause

2. Evaluate XMLQUERY for XML document of positioned row

Resultset:

```
<name>Snow Shovel,
   Basic 22 inch</name>
```

```
SELECT XMLQUERY('/PurchaseOrder/items/item[partid="100-100-01"]/name'
    PASSING PORDER) from purchaseorder where poid=11223344
```

XMLEXISTS

POID	...	CUSTID	PORDER
11223344	...	1002	<PurchaseOrder PoNum="11223344" OrderDate="2007-07-18" Status="Unshipped"> ... <items> <item> <partid>100-100-01</partid> <name>Snow Shovel, Basic 22 inch</name> <quantity>4</quantity> <price>9.99</price> </item> ... </PurchaseOrder>
11223345	...	1002	<PurchaseOrder PoNum="11223345" OrderDate="2007-07-18" Status="Unshipped"> ... <items> <item> <partid>100-100-01</partid> <name>Snow Shovel, Basic 22 inch</name> <quantity>4</quantity> <price>9.99</price> </item> ... </PurchaseOrder>
11223346	...	1005	<PurchaseOrder PoNum="11223346" OrderDate="2007-07-18" Status="Unshipped"> ... <items> <item> <partid>100-103-01</partid> <name>Snow Shovel, Super Deluxe 26 inch</name> <quantity>4</quantity> <price>9.99</price> </item> ... </PurchaseOrder>

1. Evaluate all rows according to XMLEXISTS condition

3. Select columns to return

Resultset:

POID
11223344
11223345

```
SELECT POID from purchaseorder where
XMLEXISTS('/PurchaseOrder/items/item[partid="100-100-01"]' PASSING PORDER)
```

```
POID      | ... | CUSTID | PORDER
11223344  | ... | 1002   | <PurchaseOrder PoNum="11223344"
          |     |        |                OrderDate="2007-07-18" Status="Unshipped">
          |     |        |    ...
          |     |        |    <items>
          |     |        |       <item>
          |     |        |          <partid>100-100-01</partid>
          |     |        |          <name>Snow Shovel, Basic 22 inch</name>
          |     |        |          …
          |     |        |       </item>
          |     |        |       <item>
          |     |        |          <partid>100-103-01</partid>
          |     |        |          <name>Snow Shovel, Super Deluxe 26 inch</name>
          |     |        |          …
          |     |        |       </item>
          |     |        |     ...
          |     |        | </PurchaseOrder>
11223345  | ... | 1002   | <PurchaseOrder PoNum="11223345"
          |     |        |                OrderDate="2007-07-18" Status="Unshipped">
          |     |        |    ...
          |     |        |    <items>
          |     |        |       <item>
          |     |        |          <partid>100-100-01</partid>
          |     |        |          <name>Snow Shovel, Basic 22 inch</name>
          |     |        |          <quantity>4</quantity>
          |     |        |          <price>9.99</price>
          |     |        |       </item>
          |     |        |       ...
          |     |        | </PurchaseOrder>
11223346  | ... | 1005   | <PurchaseOrder PoNum="11223346"
          |     |        |                OrderDate="2007-07-18" Status="Unshipped">
          |     |        |    ...
          |     |        |          <price>9.99</price>
          |     |        |       </item>
          |     |        |       ...
          |     |        | </PurchaseOrder>
```

1. Evaluate all rows according to XMLEXISTS condition

2. select matching rows

3. Select columns to return and evaluate XMLQUERY function

Resultset:

```
POID       ITEMS
--------   ------------------------------------------------------------------------
11223344   <name>Snow Shovel, Basic 22 inch</name><name>Snow Shovel, Super Deluxe 26 inch</name>
```

```
SELECT POID, XMLQUERY('/PurchaseOrder/items/item/name' PASSING PORDER) AS ITEMS
from purchaseorder where XMLEXISTS('/PurchaseOrder[@PoNum=11223344]' PASSING PORDER)
```

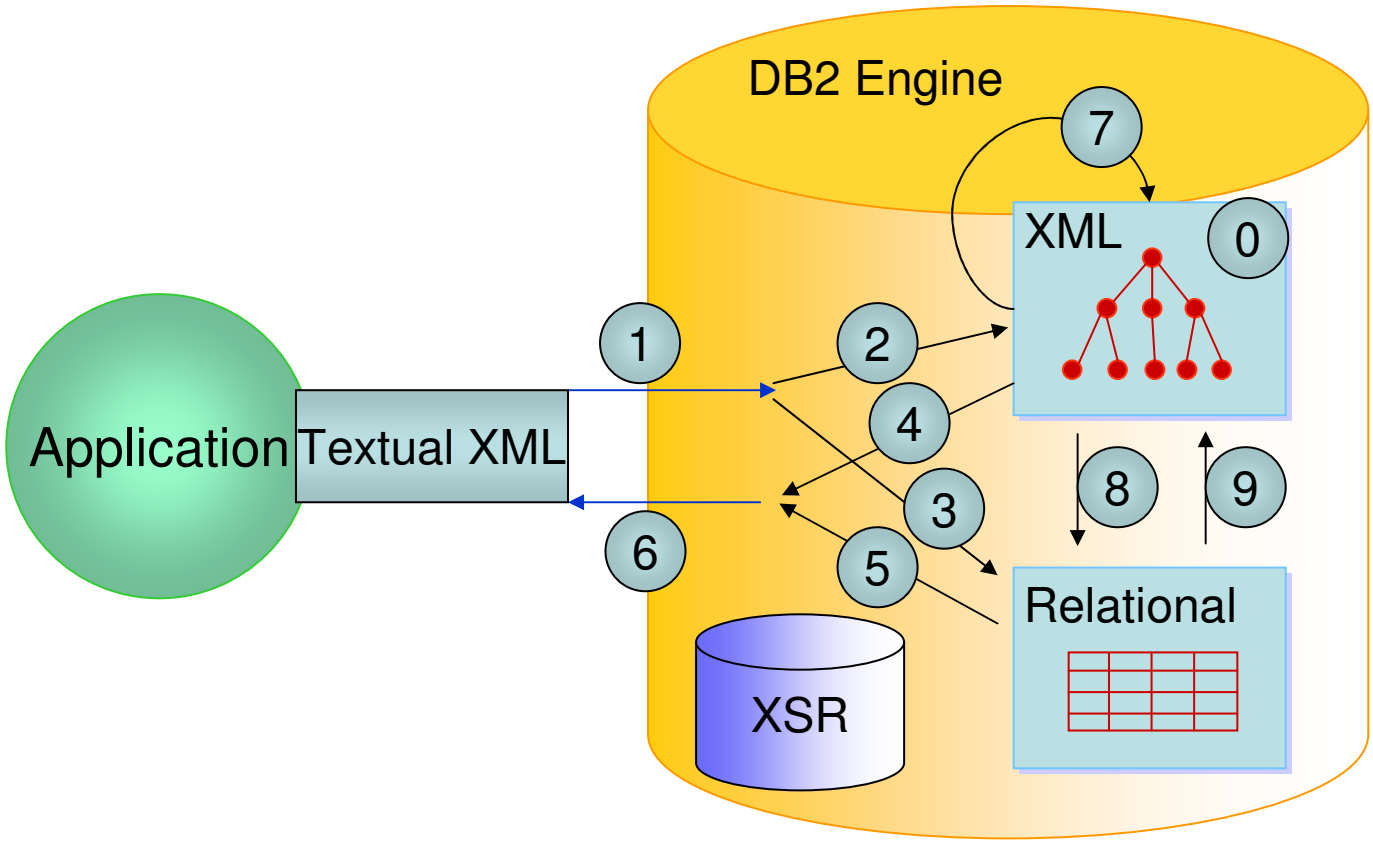
Leveraging the Power of SQL

```
CREATE VIEW ORDER_VIEW AS
SELECT PO.POID, X.*
FROM PurchaseOrders PO,
      XMLTABLE( '//item' PASSING PO.XMLPO
        COLUMNS  "orderDate"       DATE PATH '../../@orderDate',
                 "shipTo City"     VARCHAR(20) PATH '../../shipTo/city',
                 "shipTo State"    CHAR(2) PATH '../../shipTo/state',
                 "Part #"          CHAR(6) PATH '@partnum',
                 "Product Name"    CHAR(20) PATH 'productName',
                 "Quantity"        INTEGER PATH 'quantity',
                 "US Price"        DECIMAL(9,2) PATH 'USPrice',
                 "Ship Date"       DATE PATH 'shipDate',
                 "Comment"         VARCHAR(60) PATH 'comment' ) AS X;
```

```
SELECT "Product Name", "shipTo State",
        SUM("US Price" * "Quantity") AS TOTAL_SALE
FROM ORDER_VIEW
GROUP BY "Product Name", "shipTo State";
```

```
SELECT "shipTo City", "shipTo State",
         RANK() OVER(ORDER BY SUM("Quantity")) AS SALES_RANK
FROM ORDER_VIEW
WHERE "Product Name" = 'Baby Monitor'
GROUP BY "shipTo State", "shipTo City"
ORDER BY SALES_RANK;
```

Summary of SQL/XML features in DB2 for Z/OS



0 XML Storage
1 Insert XML
2 Store as XML
3 Shred into Rel
4 Retrieve XML
5 Publish XML
6 Select XML
7 XML to XML
8 XML to Rel
9 Rel to XML