IBM Data Studio and IBM's Java Persistence Strategy: pureQuery

http://www.ibm.com/software/data/studio

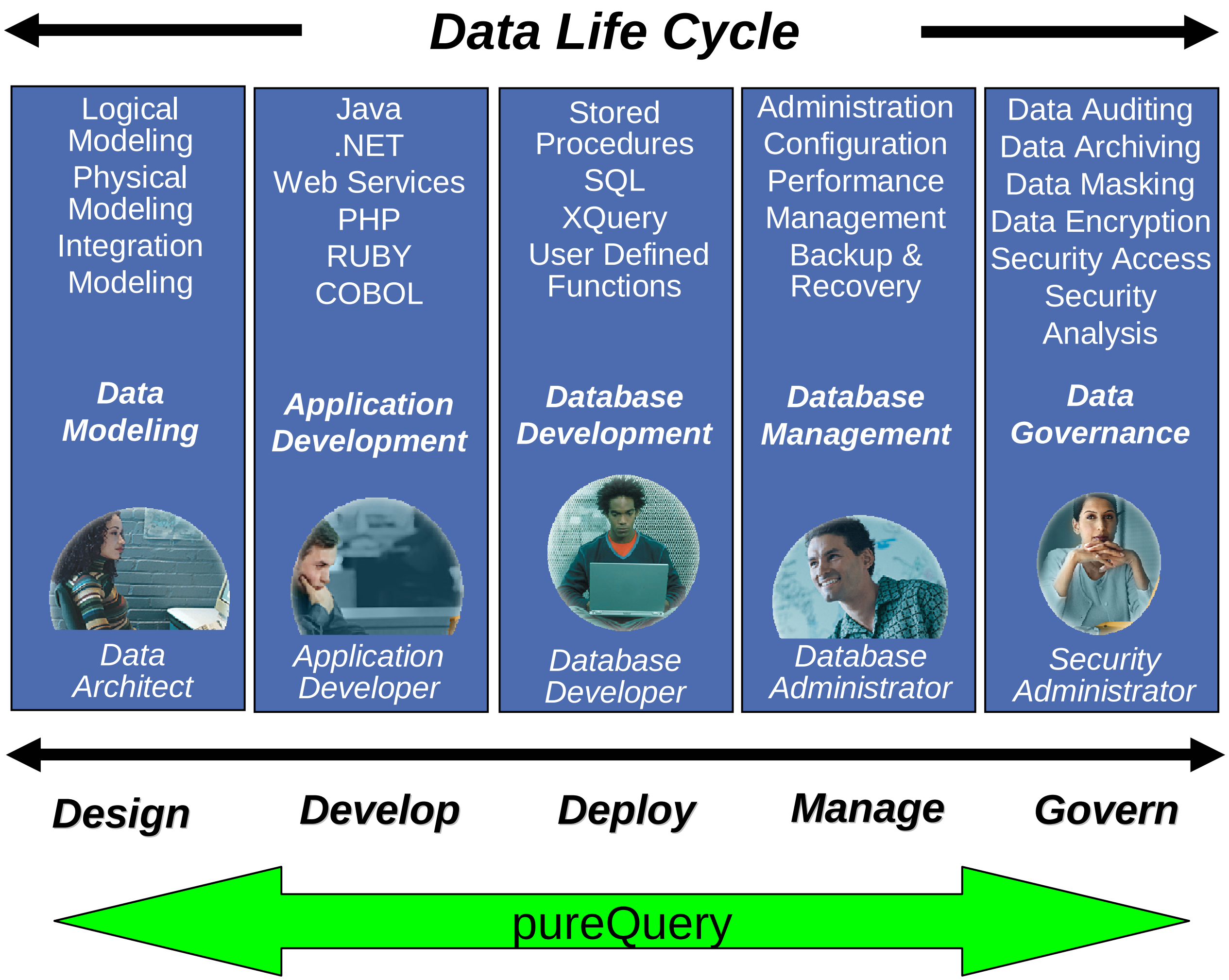
Stephen Brodsky, Architect, IBM Data Studio

December 2007

IBM Data Studio

IBM Data Studio is a comprehensive data management solution that empowers you to effectively design, develop, deploy and manage your data, databases and database applications throughout the entire application development life cycle utilizing a consistent and integrated user interface

IBM Data Studio



Data Life Cycle

Logical Modeling
Physical Modeling
Integration Modeling

Data Modeling

Data Architect

Java
.NET
Web Services
PHP
RUBY
COBOL

Application Development

Application Developer

Stored Procedures
SQL
XQuery
User Defined Functions

Database Development

Database Developer

Administration
Configuration
Performance Management
Backup & Recovery

Database Management

Database Administrator

Data Auditing
Data Archiving
Data Masking
Data Encryption
Security Access
Security Analysis

Data Governance

Security Administrator

Design ***Develop*** ***Deploy*** ***Manage*** ***Govern***

pureQuery

IBM Data Studio

A Consistent, Integrated Solution

Application Developer

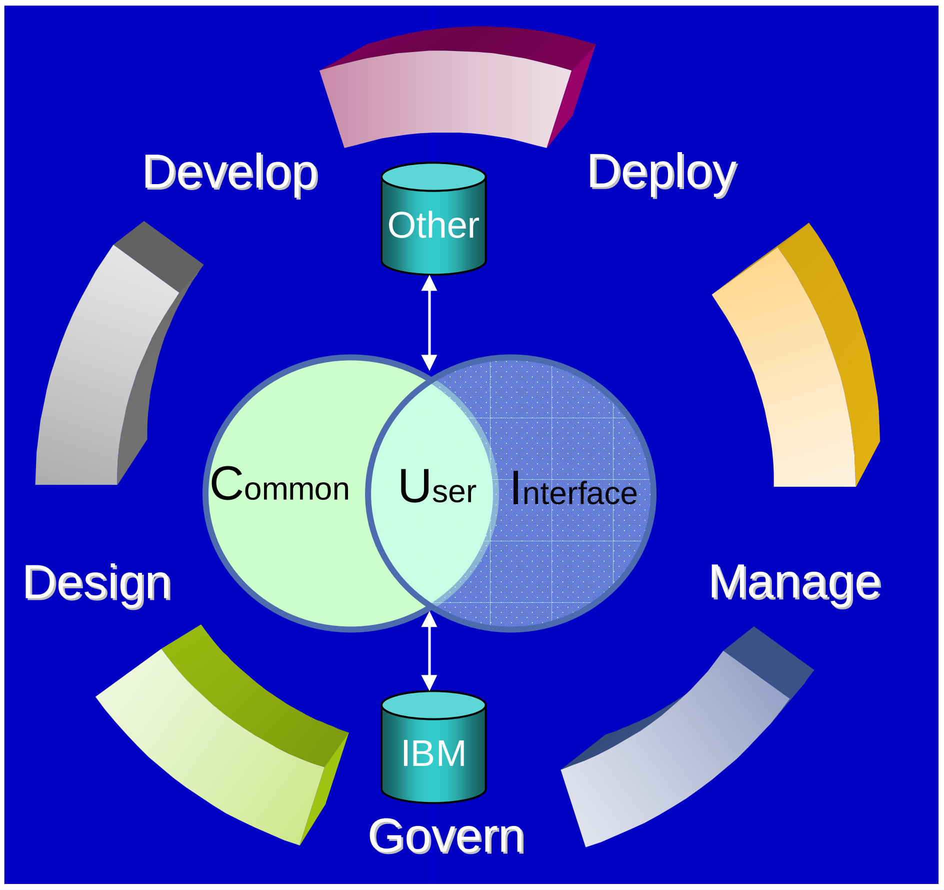
Database Developer

Develop
- Coding
- Debugging
- Teaming
- Testing
- Tuning

Design
- Logical Modeling
- Physical Modeling
- Integration Modeling

Business Analyst

Database Architect



Database Administrator

Manage
- Database Administration
- Data Management
- Change Management
- Recovery Management
- Storage Management
- Performance Management

Govern
- Security Access
- Security Analysis
- Data Auditing
- Data Archiving
- Data Masking
- Data Encryption

Security Administrator

IBM Data Studio v1.1

- Empowering developers and database administrators
- Complimentary and available in October of 2007
- Support for DB2 and IDS on all platforms

DB2 for LUW

- Physical Data Modeling
- Data Distribution Viewer
- Integrated Query Editor
- SQL Builder
- SQL Routine Debugger
- Java Routine Debugger
- XML Editor
- XML Schema Editor
- pureQuery for Java
- Data Web Services
- Object Management
- Data Management
- Update Statistics
- Health Monitoring *
- Visual Explain
- Security Access Controls
- Project Management

* Technical Preview

DB2 for z/OS

- Physical Data Modeling
- Data Distribution Viewer
- Integrated Query Editor
- SQL Builder
- SQL Routine Debugger
- Java Routine Debugger
- XML Editor
- XML Schema Editor
- pureQuery for Java
- Data Web Services
- Object Management
- Data Management
- Update Statistics
- Visual Explain
- Security Access Controls
- Project Management

DB2 for i5/OS

- Physical Data Modeling
- Data Distribution Viewer
- Integrated Query Editor
- SQL Builder
- SQL Routine Debugger
- Java Routine Debugger
- XML Editor
- XML Schema Editor
- pureQuery for Java
- Data Web Services
- Object Management
- Data Management
- Security Access Controls
- Project Management

IDS

- Physical Data Modeling
- Data Distribution Viewer
- Integrated Query Editor
- SQL Builder
- XML Editor
- XML Schema Editor
- pureQuery for Java
- Data Web Services
- Object Management
- Data Management
- Security Access Controls
- Project Management

IBM Data Studio

Increase productivity for all roles throughout the data life cycle

- Slash development time up to 50% with an integrated data management environment
- Promote collaboration across roles to optimize data server and application performance
- Accelerate Java development productivity with new pureQuery data access
- Simplify development of applications implementing industry specific XML standards
- Monitor data server operation & performance anywhere, anytime from a Web browser

Simplify and speed development of new skills

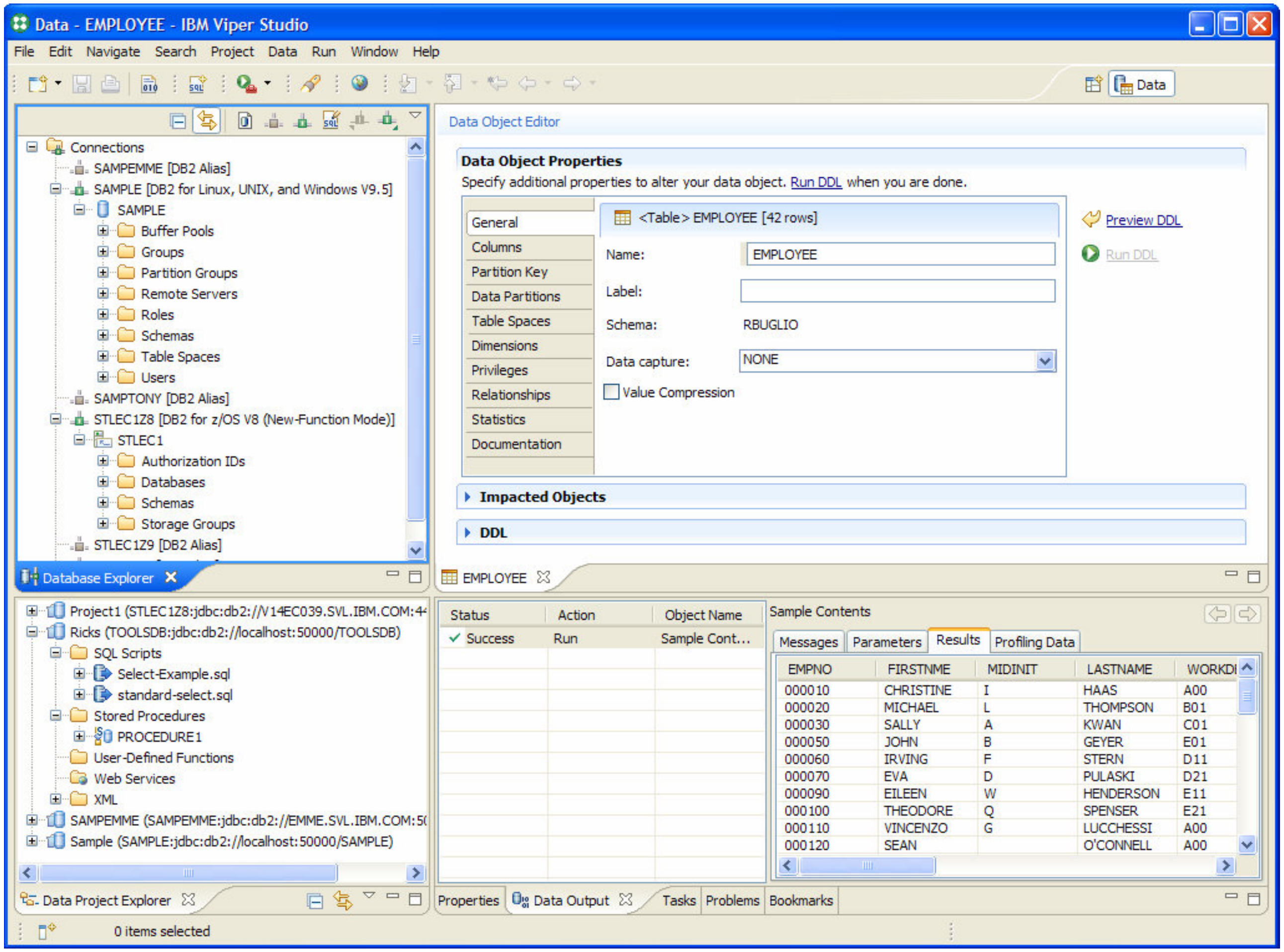
- Learn once, use with all supported data servers
- Easy-to-use and integrated user interface, compatible with Rational Software Development Platform
- Extensible with Eclipse plug-ins to customize the environment for each team member

Accelerate data as a service for Service Oriented Architecture

- Develop and publish data as a Web service without programming
- Info 2.0 Ready - support for Web 2.0 protocols and format

IBM Data Studio – Workbench

A single productive work environment



Data Object Properties

Specify additional properties to alter your data object. Run DDL when you are done.

<Table> EMPLOYEE [42 rows]

Name: EMPLOYEE

Label:

Schema: RBUGLIO

Data capture: NONE

□ Value Compression

Status	Action	Object Name
✓ Success	Run	Sample Cont…

Sample Contents

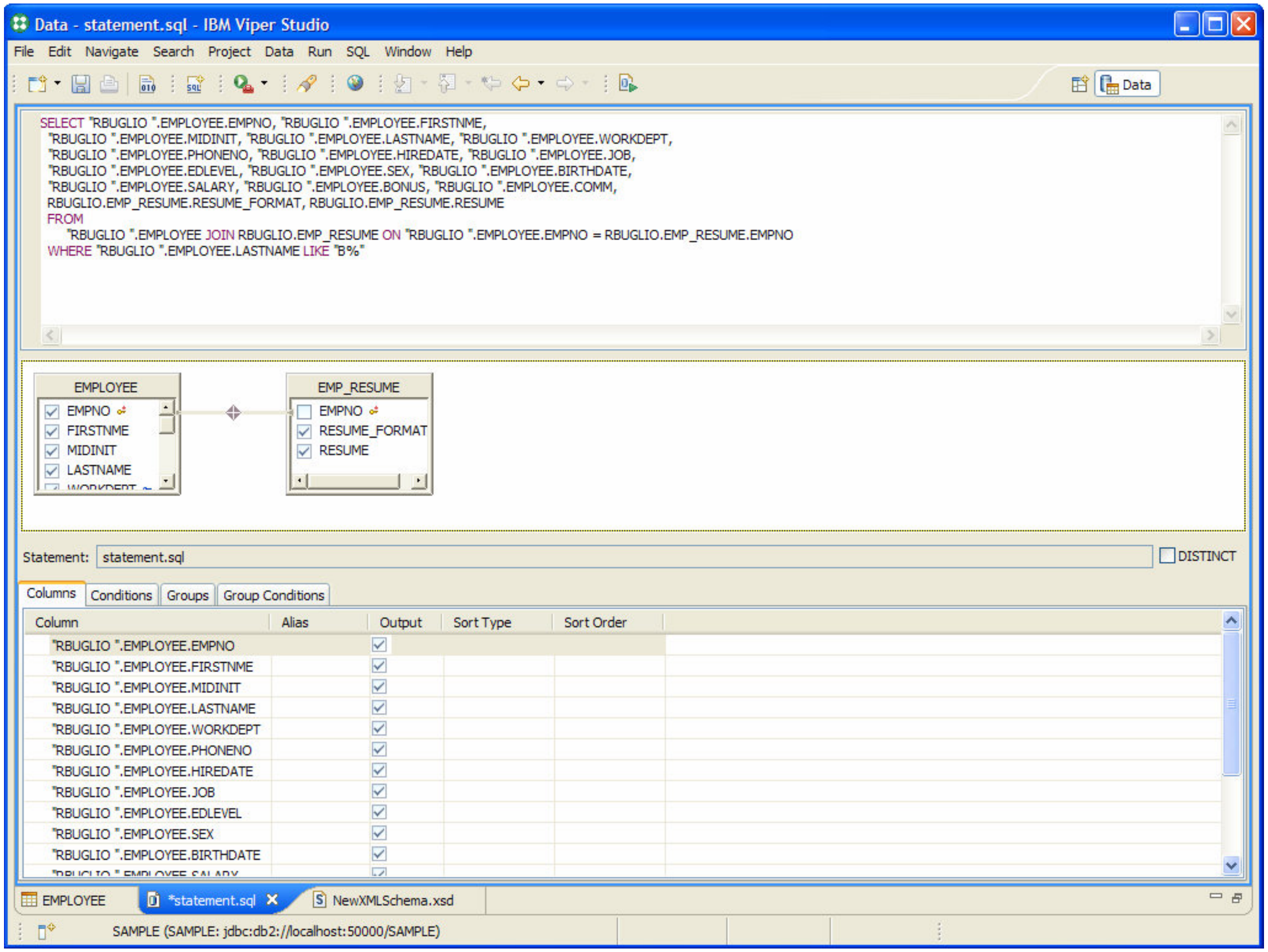
Messages | Parameters | Results | Profiling Data

EMPNO	FIRSTNME	MIDINIT	LASTNAME	WORKDI
000010	CHRISTINE	I	HAAS	A00
000020	MICHAEL	L	THOMPSON	B01
000030	SALLY	A	KWAN	C01
000050	JOHN	B	GEYER	E01
000060	IRVING	F	STERN	D11
000070	EVA	D	PULASKI	D21
000090	EILEEN	W	HENDERSON	E11
000100	THEODORE	Q	SPENSER	E21
000110	VINCENZO	G	LUCCHESSI	A00
000120	SEAN		O'CONNELL	A00

0 items selected

IBM Data Studio – Workbench

An integrated query editor for SQL and XQuery



```
SELECT "RBUGLIO ".EMPLOYEE.EMPNO, "RBUGLIO ".EMPLOYEE.FIRSTNME,
 "RBUGLIO ".EMPLOYEE.MIDINIT, "RBUGLIO ".EMPLOYEE.LASTNAME, "RBUGLIO ".EMPLOYEE.WORKDEPT,
 "RBUGLIO ".EMPLOYEE.PHONENO, "RBUGLIO ".EMPLOYEE.HIREDATE, "RBUGLIO ".EMPLOYEE.JOB,
 "RBUGLIO ".EMPLOYEE.EDLEVEL, "RBUGLIO ".EMPLOYEE.SEX, "RBUGLIO ".EMPLOYEE.BIRTHDATE,
 "RBUGLIO ".EMPLOYEE.SALARY, "RBUGLIO ".EMPLOYEE.BONUS, "RBUGLIO ".EMPLOYEE.COMM,
 RBUGLIO.EMP_RESUME.RESUME_FORMAT, RBUGLIO.EMP_RESUME.RESUME
 FROM
    "RBUGLIO ".EMPLOYEE JOIN RBUGLIO.EMP_RESUME ON "RBUGLIO ".EMPLOYEE.EMPNO = RBUGLIO.EMP_RESUME.EMPNO
 WHERE "RBUGLIO ".EMPLOYEE.LASTNAME LIKE "B%"
```

Statement: statement.sql □ DISTINCT

Columns | Conditions | Groups | Group Conditions

Column	Alias	Output	Sort Type	Sort Order
"RBUGLIO ".EMPLOYEE.EMPNO		☑		
"RBUGLIO ".EMPLOYEE.FIRSTNME		☑		
"RBUGLIO ".EMPLOYEE.MIDINIT		☑		
"RBUGLIO ".EMPLOYEE.LASTNAME		☑		
"RBUGLIO ".EMPLOYEE.WORKDEPT		☑		
"RBUGLIO ".EMPLOYEE.PHONENO		☑		
"RBUGLIO ".EMPLOYEE.HIREDATE		☑		
"RBUGLIO ".EMPLOYEE.JOB		☑		
"RBUGLIO ".EMPLOYEE.EDLEVEL		☑		
"RBUGLIO ".EMPLOYEE.SEX		☑		
"RBUGLIO ".EMPLOYEE.BIRTHDATE		☑		
"RBUGLIO ".EMPLOYEE.SALARY		☑		

SAMPLE (SAMPLE: jdbc:db2://localhost:50000/SAMPLE)

Data Web Services without Programming

Turn frequently used database operations as Web services for reuse

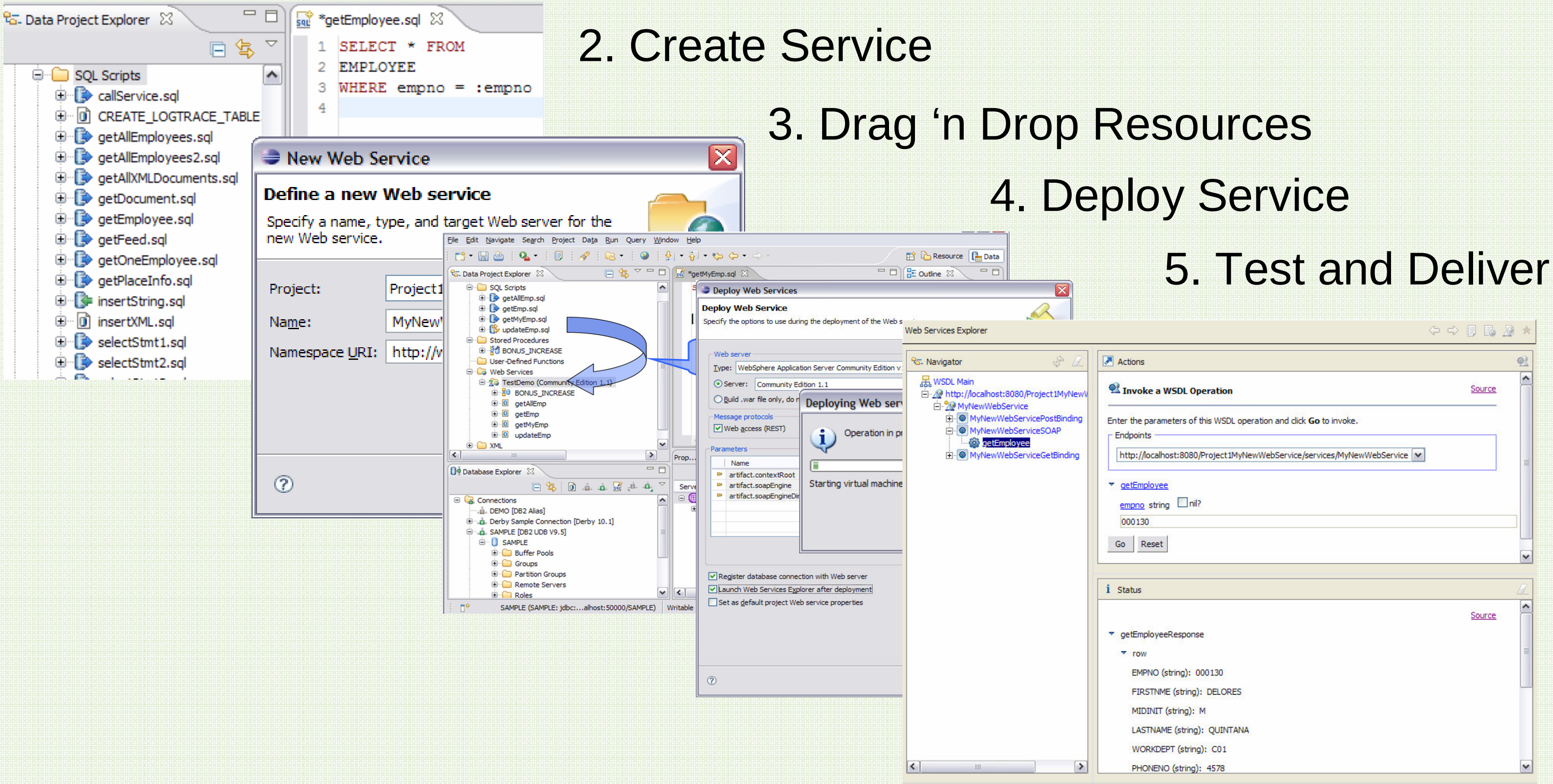
1. Create and Test Queries or Stored Procedures

2. Create Service

3. Drag ‘n Drop Resources

4. Deploy Service

5. Test and Deliver



Data Project Explorer

- SQL Scripts
  - callService.sql
  - CREATE_LOGTRACE_TABLE
  - getAllEmployees.sql
  - getAllEmployees2.sql
  - getAllXMLDocuments.sql
  - getDocument.sql
  - getEmployee.sql
  - getFeed.sql
  - getOneEmployee.sql
  - getPlaceInfo.sql
  - insertString.sql
  - insertXML.sql
  - selectStmt1.sql
  - selectStmt2.sql

*getEmployee.sql

```
SELECT * FROM
EMPLOYEE
WHERE empno = :empno
```

New Web Service

Define a new Web service

Specify a name, type, and target Web server for the new Web service.

Project: Project1

Name: MyNew'

Namespace URI: http://w

File Edit Navigate Search Project Data Run Query Window Help

Data Project Explorer

- SQL Scripts
  - getAllEmp.sql
  - getEmp.sql
  - getMyEmp.sql
  - updateEmp.sql
- Stored Procedures
  - BONUS_INCREASE
- User-Defined Functions
- Web Services
  - TestDemo (Community Edition 1.1)
    - BONUS_INCREASE
    - getAllEmp
    - getEmp
    - getMyEmp
    - updateEmp
- XML

Database Explorer

- Connections
  - DEMO [DB2 Alias]
  - Derby Sample Connection [Derby 10.1]
  - SAMPLE [DB2 UDB V9.5]
    - SAMPLE
      - Buffer Pools
      - Groups
      - Partition Groups
      - Remote Servers
      - Roles

SAMPLE (SAMPLE: jdbc:...alhost:50000/SAMPLE) Writable

*getMyEmp.sql Outline Resource Data

Deploy Web Services

Deploy Web Service

Specify the options to use during the deployment of the Web s

Web server

Type: WebSphere Application Server Community Edition v

● Server: Community Edition 1.1

○ Build .war file only, do n

Message protocols

☑ Web access (REST)

Parameters

Name
artifact.contextRoot
artifact.soapEngine
artifact.soapEngineDi

☑ Register database connection with Web server

☑ Launch Web Services Explorer after deployment

☐ Set as default project Web service properties

Deploying Web ser

Operation in pr

Starting virtual machine

Web Services Explorer

Navigator

- WSDL Main
  - http://localhost:8080/Project1MyNewW
    - MyNewWebService
      - MyNewWebServicePostBinding
      - MyNewWebServiceSOAP
        - getEmployee
      - MyNewWebServiceGetBinding

Actions

Invoke a WSDL Operation Source

Enter the parameters of this WSDL operation and click **Go** to invoke.

Endpoints: http://localhost:8080/Project1MyNewWebService/services/MyNewWebService

getEmployee

empno string ☐ nil?

000130

Go Reset

Status Source

getEmployeeResponse

row

EMPNO (string): 000130

FIRSTNME (string): DELORES

MIDINIT (string): M

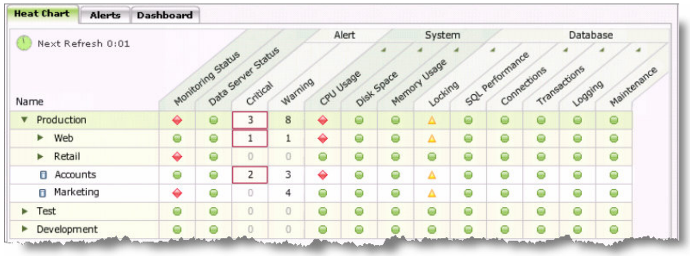
LASTNAME (string): QUINTANA

WORKDEPT (string): C01

PHONENO (string): 4578

Quick & Easy
Problem Determination

Heatchart – Overall Health Status



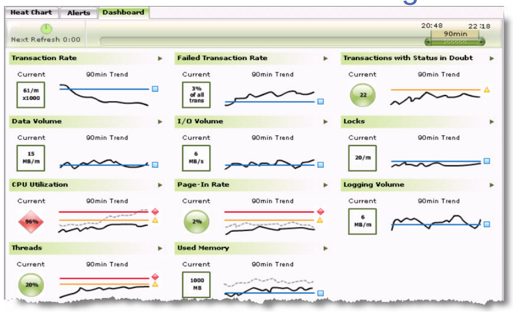
Heat Chart | Alerts | Dashboard

Next Refresh 0:01

			Alert		System				Database				
Name	Monitoring Status	Data Server Status	Critical	Warning	CPU Usage	Disk Space	Memory Usage	Locking	SQL Performance	Connections	Transactions	Logging	Maintenance
▼ Production	◆	●	3	8	◆	●	●	△	●	●	●	●	●
▶ Web	●	●	1	1	◆	●	●	△	●	●	●	●	●
▶ Retail	◆	●	0	0	●	●	●	●	●	●	●	●	●
Accounts	●	●	2	3	◆	●	●	△	●	●	●	●	●
Marketing	◆	●	0	4	●	●	●	△	●	●	●	●	●
▶ Test	●	●	0	0	●	●	●	●	●	●	●	●	●
▶ Development	●	●	0	0	●	●	●	●	●	●	●	●	●

Where are the most important hotspots that need my attention?

Dashboard – Adhoc Investigation



Heat Chart | Alerts | Dashboard

Next Refresh 0:00

20:48 22 :18 90min

Transaction Rate ▶ — Current: 61/m x1000 — 90min Trend

Failed Transaction Rate ▶ — Current: 3% of all trans — 90min Trend

Transactions with Status in Doubt ▶ — Current: 22 — 90min Trend

Data Volume ▶ — Current: 15 MB/m — 90min Trend

I/O Volume ▶ — Current: 6 MB/s — 90min Trend

Locks ▶ — Current: 20/m — 90min Trend

CPU Utilization ▶ — Current: 96% — 90min Trend

Page-In Rate ▶ — Current: 2% — 90min Trend

Logging Volume ▶ — Current: 6 MB/m — 90min Trend

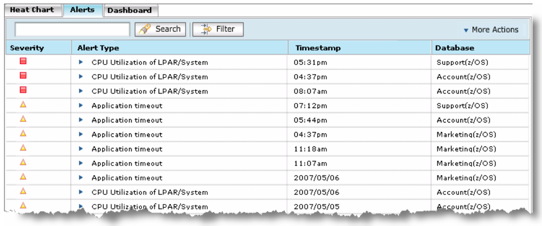
Threads ▶ — Current: 20% — 90min Trend

Used Memory ▶ — Current: 1000 MB — 90min Trend

Something doesn‘t seem quite right.
I wonder what‘s happening?

Database Administrator

Alert List – Historical Investigation



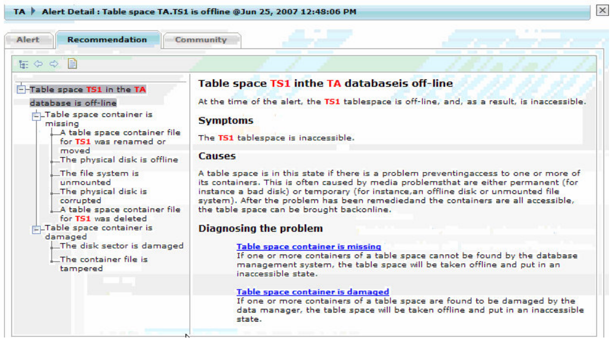
Heat Chart | Alerts | Dashboard

Search | Filter | ▾ More Actions

Severity	Alert Type	Timestamp	Database
■	▶ CPU Utilization of LPAR/System	05:31pm	Support(z/OS)
■	▶ CPU Utilization of LPAR/System	04:37pm	Account(z/OS)
■	▶ CPU Utilization of LPAR/System	08:07am	Account(z/OS)
△	▶ Application timeout	07:12pm	Support(z/OS)
△	▶ Application timeout	05:44pm	Account(z/OS)
△	▶ Application timeout	04:37pm	Marketing(z/OS)
△	▶ Application timeout	11:18am	Marketing(z/OS)
△	▶ Application timeout	11:07am	Marketing(z/OS)
△	▶ Application timeout	2007/05/06	Marketing(z/OS)
△	▶ CPU Utilization of LPAR/System	2007/05/06	Account(z/OS)
△	▶ CPU Utilization of LPAR/System	2007/05/05	Account(z/OS)

What happened when I was out for lunch? ... Away for weekend?

Recommendations – Root Cause Analysis



TA ▶ Alert Detail : Table space TA.TS1 is offline @Jun 25, 2007 12:48:06 PM

Alert | Recommendation | Community

- Table space TS1 in the TA database is off-line
  - Table space container is missing
    - A table space container file for TS1 was renamed or moved
    - The physical disk is offline
    - The file system is unmounted
    - The physical disk is corrupted
    - A table space container file for TS1 was deleted
  - Table space container is damaged
    - The disk sector is damaged
    - The container file is tampered

Table space TS1 inthe TA databaseis off-line

At the time of the alert, the TS1 tablespace is off-line, and, as a result, is inaccessible.

Symptoms

The TS1 tablespace is inaccessible.

Causes

A table space is in this state if there is a problem preventingaccess to one or more of its containers. This is often caused by media problemsthat are either permanent (for instance a bad disk) or temporary (for instance,an offline disk or unmounted file system). After the problem has been remediedand the containers are all accessible, the table space can be brought backonline.

Diagnosing the problem

Table space container is missing
If one or more containers of a table space cannot be found by the database management system, the table space will be taken offline and put in an inaccessible state.

Table space container is damaged
If one or more containers of a table space are found to be damaged by the data manager, the table space will be taken offline and put in an inaccessible state.

Guide me to the root cause and help me fix it properly; I need to know all the revelant info to make best decision.

Comparison - Developer Workbench vs. IBM Data Studio

2006 2007

IBM DB2 Developer Workbench V9.1

- SQL Query Editor
- SQLJ Editor
- SQL Builder
- XQuery Builder
- SQL Routine Debugger
- Java Routine Debugger
- XML Editor
- XML Schema Editor
- Data Management
- Visual Explain
- Project Management

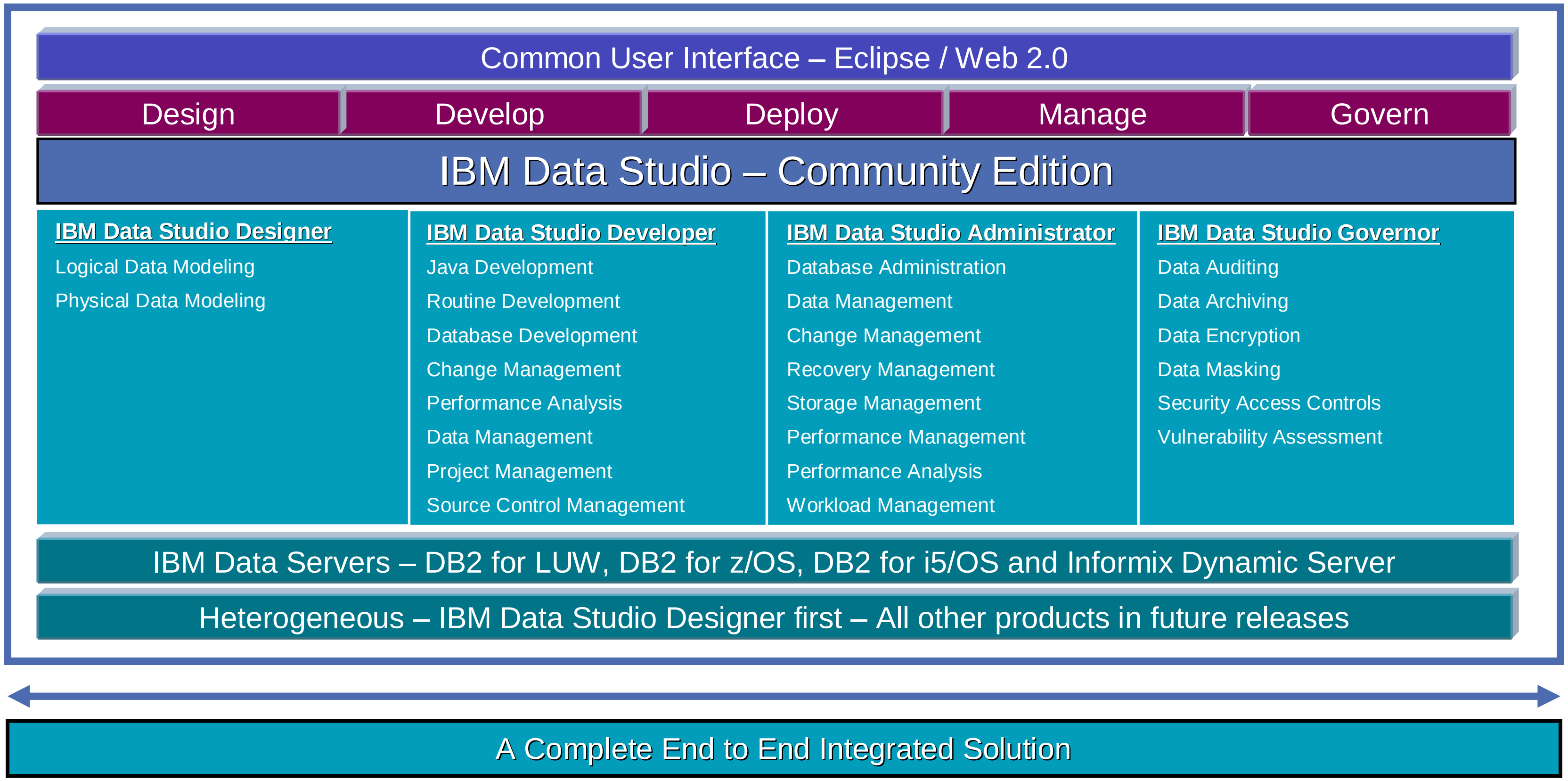
IBM Data Studio

- Integrated Query Editor
- SQLJ Editor
- SQL Builder
- Integrated Query Editor
- SQL Routine Debugger
- Java Routine Debugger
- XML Editor
- XML Schema Editor
- Data Management
- Visual Explain
- Project Management

- Physical Data Diagramming
- Data Distribution Viewer
- Object Management
- Browse & Update Statistics
- Security Access Control
- Connection Management integration with Kerberos and LDAP
- Data Web Services
- IDS Servers Support
- pureQuery for Java*

Data Studio is a full replacement of DB2 Developer Workbench, plus more

IBM Data Studio – 2008 and Beyond



Common User Interface – Eclipse / Web 2.0

Design | Develop | Deploy | Manage | Govern

IBM Data Studio – Community Edition

IBM Data Studio Designer	IBM Data Studio Developer	IBM Data Studio Administrator	IBM Data Studio Governor
Logical Data Modeling	Java Development	Database Administration	Data Auditing
Physical Data Modeling	Routine Development	Data Management	Data Archiving
	Database Development	Change Management	Data Encryption
	Change Management	Recovery Management	Data Masking
	Performance Analysis	Storage Management	Security Access Controls
	Data Management	Performance Management	Vulnerability Assessment
	Project Management	Performance Analysis	
	Source Control Management	Workload Management	

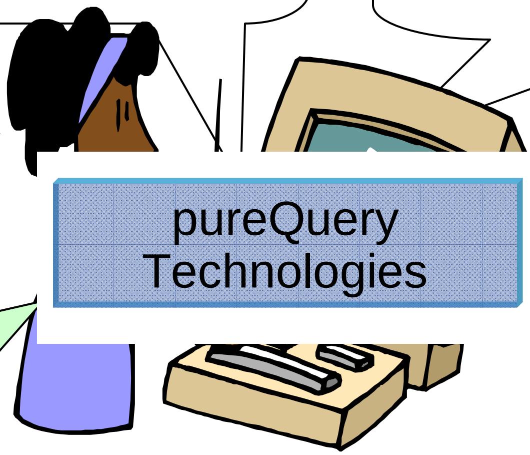
IBM Data Servers – DB2 for LUW, DB2 for z/OS, DB2 for i5/OS and Informix Dynamic Server

Heterogeneous – IBM Data Studio Designer first – All other products in future releases

A Complete End to End Integrated Solution

- A single consistent, flexible and intuitive user interface
- Common code based shared by all products
- Products that can be plugged-in as needed to satisfy customer needs and requirements and meet changing market demands
- Products that can be sold individually, packaged as part of an edition or customized in any combination ala carte style
- Products packaged with value added components that satisfy the daily workflow requirements of specific end user roles
- Products that support all IBM Data Servers
- Products positioned to support Non-IBM Data Servers

Java access to relational – pureQuery on ramp for all cases



pureQuery Technologies

I don’t want to have any SQL in my program.

I need to run on J2SE.

I want a full persistence layer with transaction management.

I just need a light-weight interface that makes SQL coding easy.

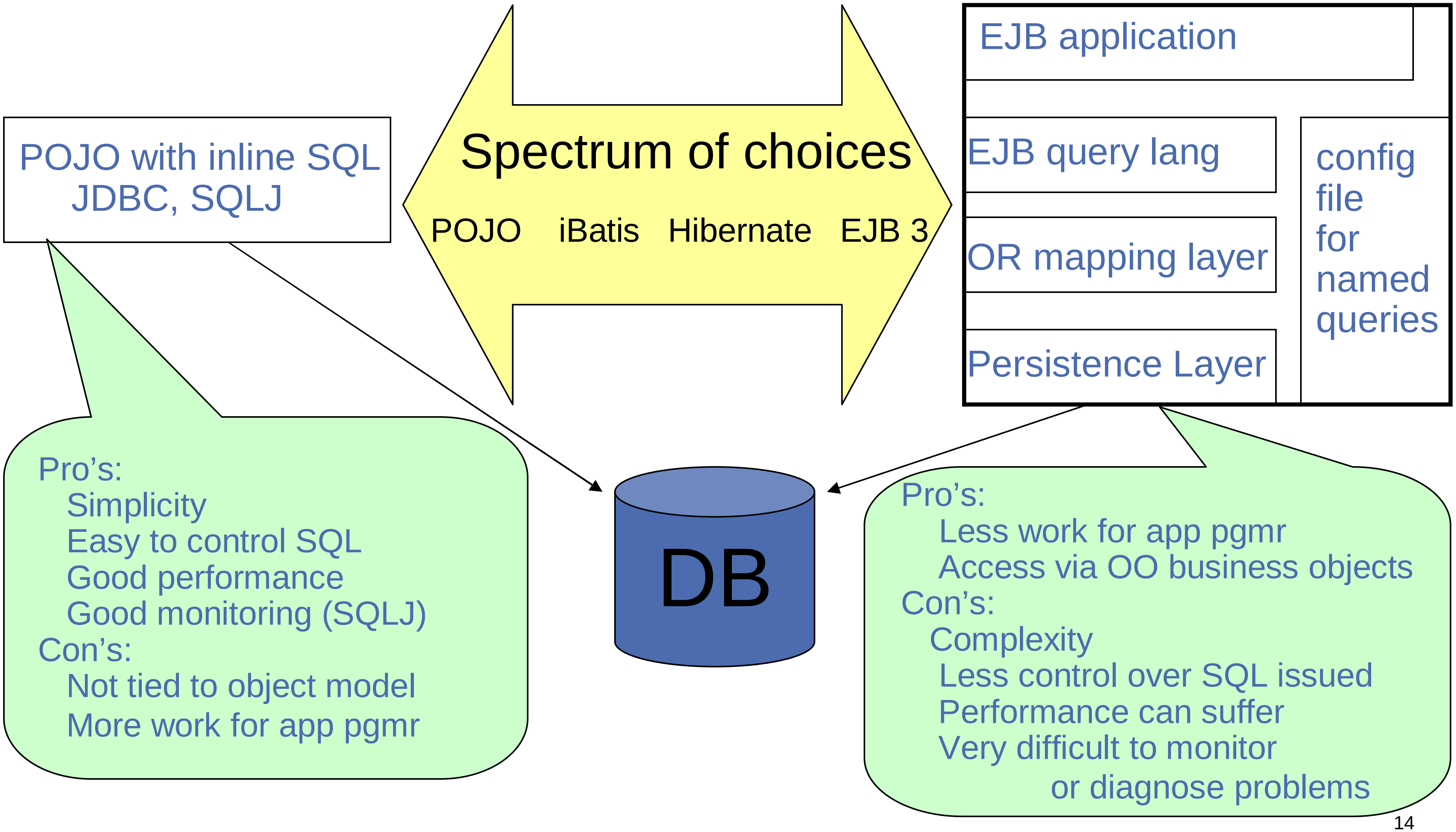
I need a solution that locks in database access path for reliable performance.

I need OR mapping between relational and my object model.

I want business objects returned to my program – not individual columns.

I need an interface that allows efficient SQL access and detailed monitoring for problem diagnosis.

Java Data Access – many forms



POJO with inline SQL
JDBC, SQLJ

Spectrum of choices

POJO iBatis Hibernate EJB 3

EJB application

EJB query lang

OR mapping layer

Persistence Layer

config file for named queries

DB

Pro's:
- Simplicity
- Easy to control SQL
- Good performance
- Good monitoring (SQLJ)

Con's:
- Not tied to object model
- More work for app pgmr

Pro's:
- Less work for app pgmr
- Access via OO business objects

Con's:
- Complexity
- Less control over SQL issued
- Performance can suffer
- Very difficult to monitor or diagnose problems

What performance/diagnosis challenges?

EJB Query Language:

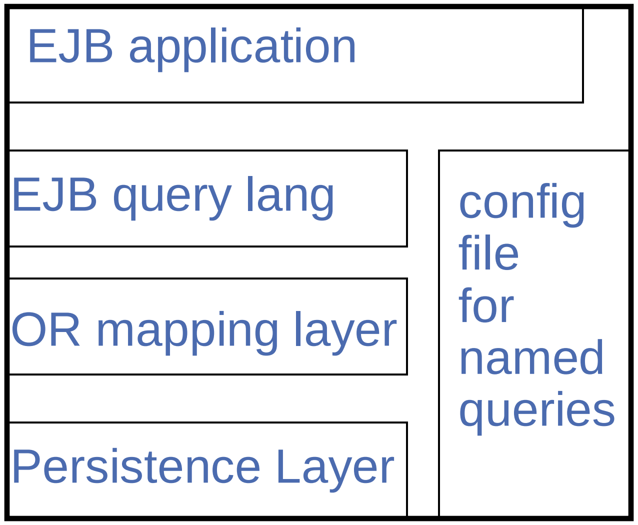
```
SELECT object(e) FROM Employee e
   WHERE e.dept=?1 AND e.salary>=?2
```

Query language is a subset of SQL. Doesn't have all the SQL features you want.

OR mapping, Transform to SQL

App query syntax is different from SQL query. How do you track problem SQL queries back to the app that issued the original query???

In most cases, queries map to JDBC. No ability to lock in access path at program deployment. No ability to search catalog to see which queries are issued by a given program.


EJB application

EJB query lang

OR mapping layer

Persistence Layer

config file for named queries

SQL issued to database:

```
SELECT * FROM PROD.EMP
   WHERE DEPT=? AND SALARY>?
```

Often, app query is intercepted by persistence layer, and the resulting SQL query looks nothing like the app query.

-- Resulting query might perform badly.

-- Changing app query might not result in a similar change in the SQL query…

pureQuery

pureQuery is a high-performance Java data access platform focused on simplifying the tasks of developing and managing applications that access data.

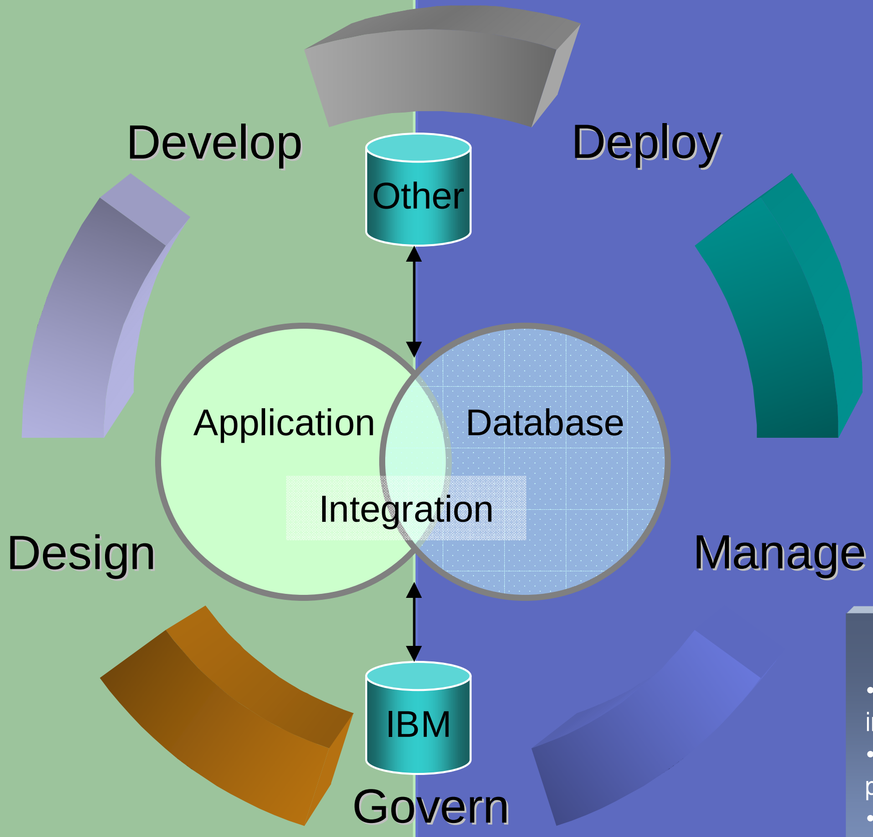
Improves the Java data access life cycle

- Tools, APIs, and runtime environment
- Single API to query databases and in-memory Java objects
- Embraces SQL as the common query language
- Simplify SQL Data Access

pureQuery across the Life Cycle

Develop

- Java and SQL development integrated
- Simple SQL APIs, easy to write to and extend
- Multiple API “styles” to address popular Java frameworks
- Query Database, Cache, Collections, XML with 1 API
- Query with Standard SQL and XQuery
- Best performance



Management and Problem Determination

- Find source application quickly
- Policy driven
- Prevent resource leaks
- Data Management
- Change Management
- Performance Mgmt
- Performance Tuning
- End-to-End Test

Design

- Connect logical, physical design with query, application objects, and model
- Tracability and Impact analysis
- Analyze static and runtime of application and database together

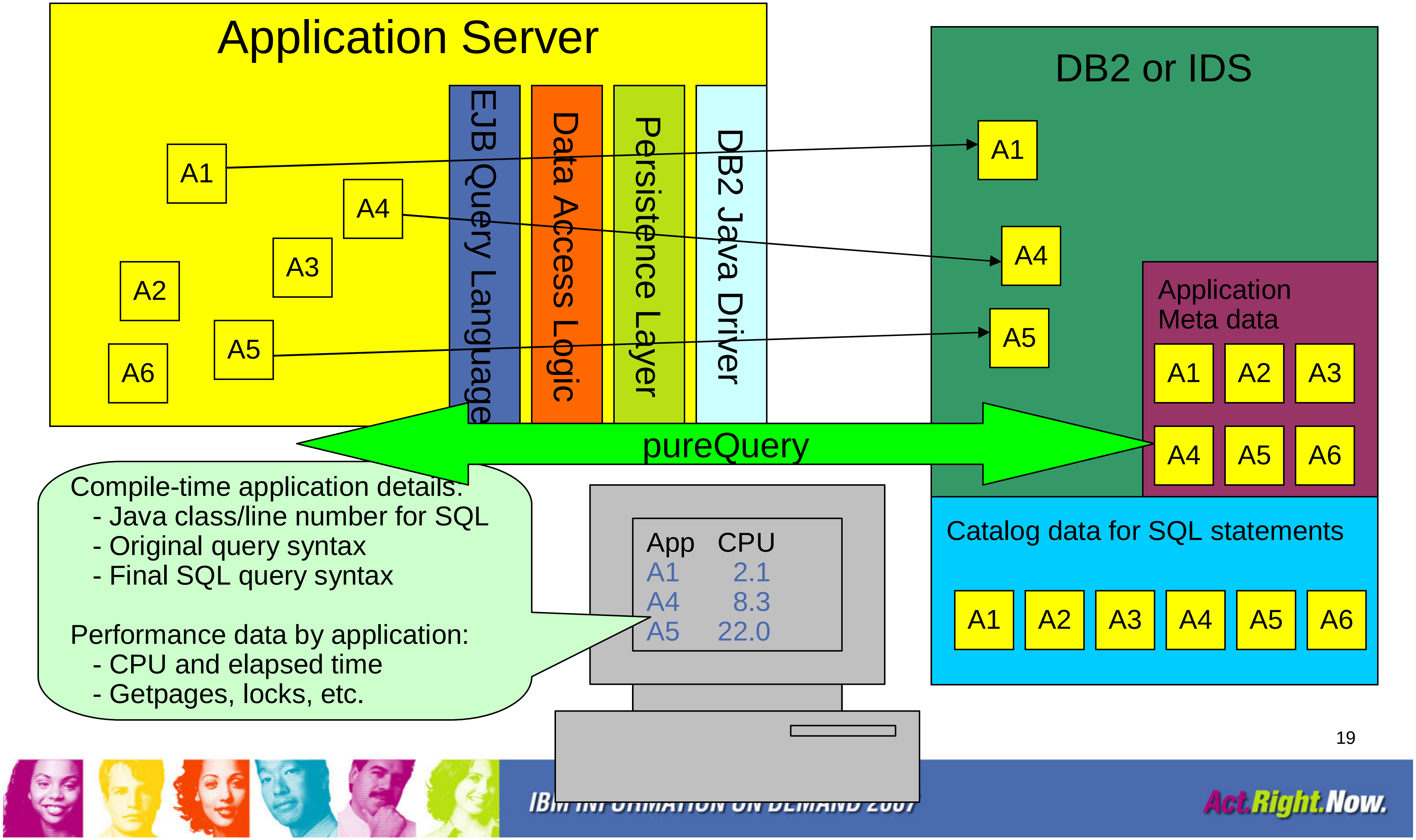
Governance / Management

- Track SQL back to individual apps, lock in access paths with static SQL
- Align with customer change control processes
- Know which programs access which parts of DB and the reverse
- Govern enterprise DB access

pureQuery – Across the Life Cycle

- Development of applications
  - Tools to assist SQL development in .java source file
  - Simple SQL APIs, easy to write to and extend
  - Multiple API “styles” to align with popular Java frameworks, including JPA/EJB3
- Query important data sources simply
  - Database, Cache, Collections, XML
- Problem Determination
  - When problems occur, find source quickly across the tiers
- Governance / Management
  - Track SQL back to individual apps, lock in access paths with **static SQL** packages, align with customer change control processes
- High performance/scalability
  - Application: short path length, coding over metadata, optional code gen, JDBC and **static SQL** runtime optimizations
  - Database: **static SQL**, batching, pass app SQL directly to database

pureQuery with IBM Runtime/Tooling

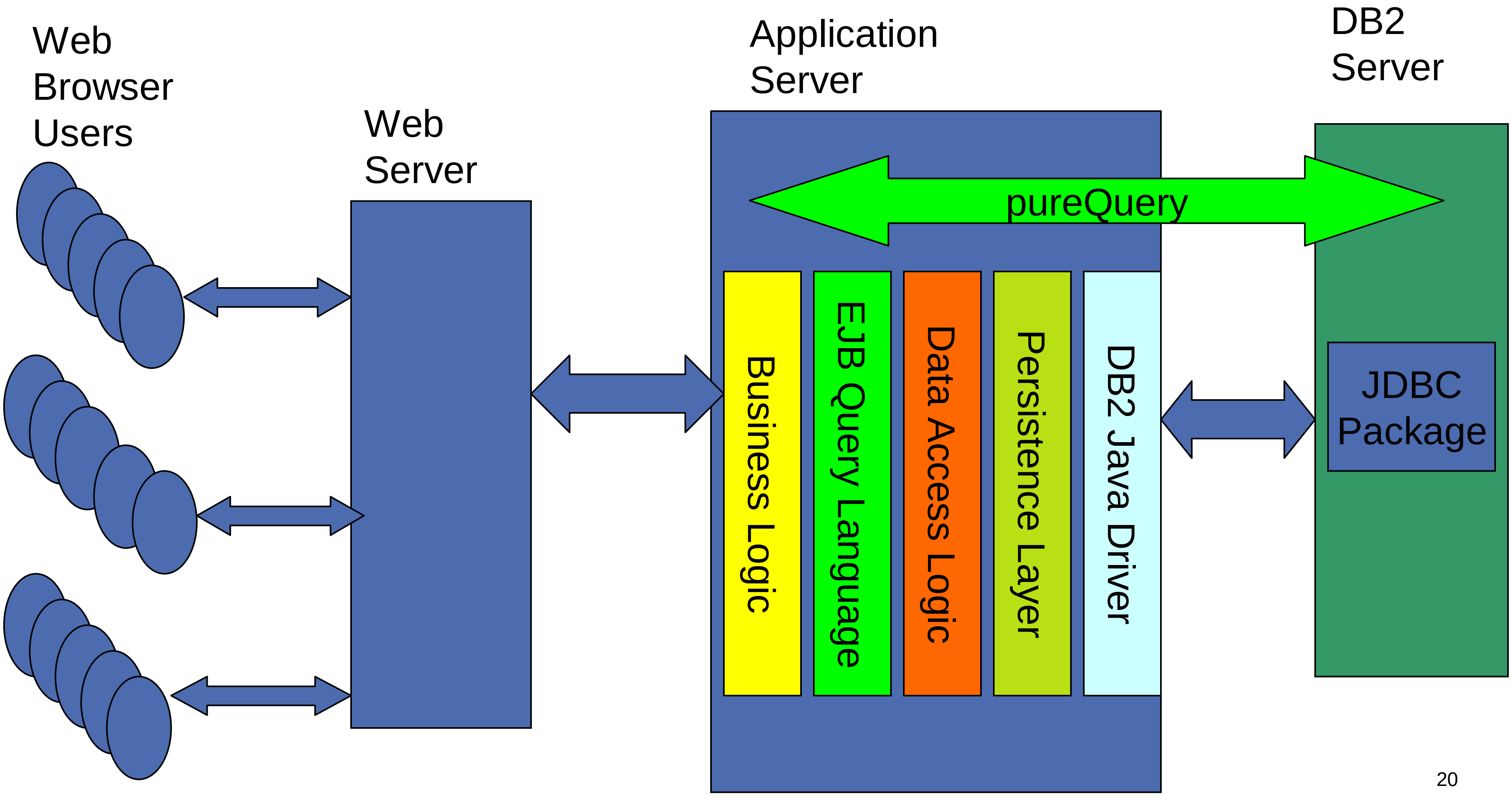


Compile-time application details:
- Java class/line number for SQL
- Original query syntax
- Final SQL query syntax

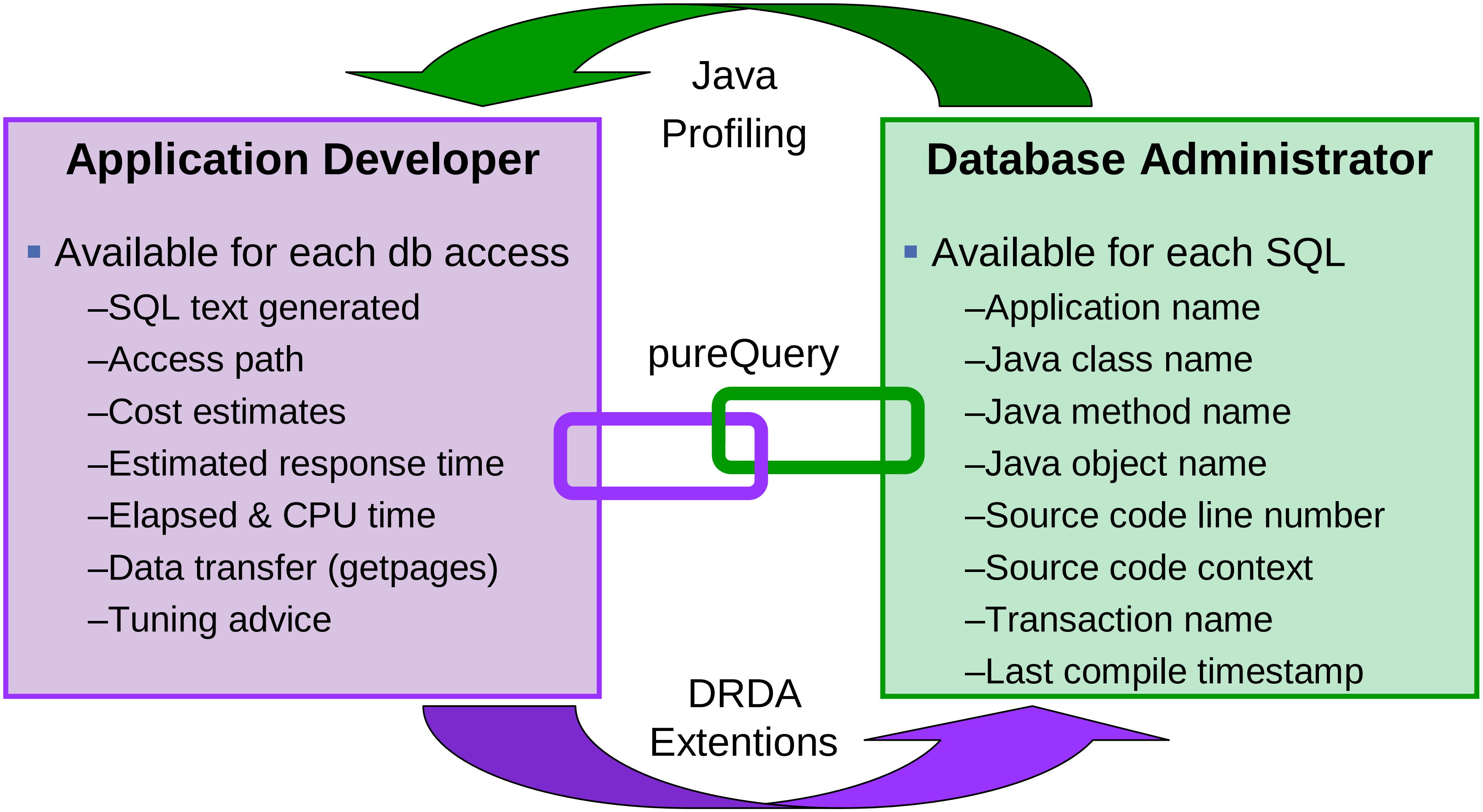
Performance data by application:
- CPU and elapsed time
- Getpages, locks, etc.

App	CPU
A1	2.1
A4	8.3
A5	22.0

Toughest issue for Web applications – Problem diagnosis and resolution



Simplifying Problem Determination Scenario



Java

Profiling

Application Developer

- Available for each db access
  - –SQL text generated
  - –Access path
  - –Cost estimates
  - –Estimated response time
  - –Elapsed & CPU time
  - –Data transfer (getpages)
  - –Tuning advice

pureQuery

Database Administrator

- Available for each SQL
  - –Application name
  - –Java class name
  - –Java method name
  - –Java object name
  - –Source code line number
  - –Source code context
  - –Transaction name
  - –Last compile timestamp

DRDA

Extentions

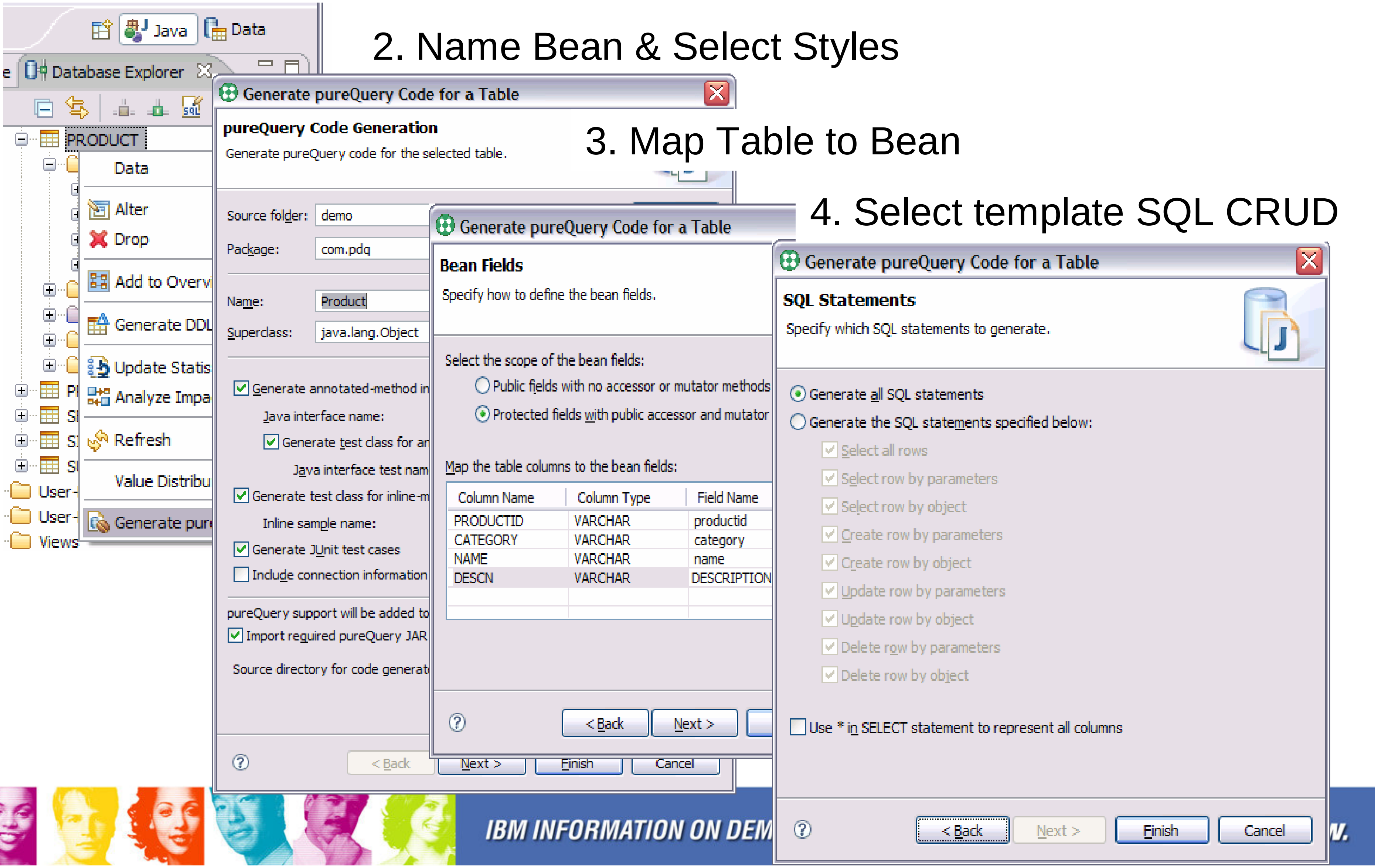
Scenario: Java Data Access in 4 simple steps

1. Select Table

2. Name Bean & Select Styles

3. Map Table to Bean

4. Select template SQL CRUD



Generate pureQuery Code for a Table

pureQuery Code Generation

Generate pureQuery code for the selected table.

Source folder: demo

Package: com.pdq

Name: Product

Superclass: java.lang.Object

☑ Generate annotated-method in

Java interface name:

☑ Generate test class for an

Java interface test nam

☑ Generate test class for inline-m

Inline sample name:

☑ Generate JUnit test cases

☐ Include connection information

pureQuery support will be added to

☑ Import required pureQuery JAR

Source directory for code generat

Generate pureQuery Code for a Table

Bean Fields

Specify how to define the bean fields.

Select the scope of the bean fields:

○ Public fields with no accessor or mutator methods

● Protected fields with public accessor and mutator

Map the table columns to the bean fields:

Column Name	Column Type	Field Name
PRODUCTID	VARCHAR	productid
CATEGORY	VARCHAR	category
NAME	VARCHAR	name
DESCN	VARCHAR	DESCRIPTION

Generate pureQuery Code for a Table

SQL Statements

Specify which SQL statements to generate.

● Generate all SQL statements

○ Generate the SQL statements specified below:

☑ Select all rows

☑ Select row by parameters

☑ Select row by object

☑ Create row by parameters

☑ Create row by object

☑ Update row by parameters

☑ Update row by object

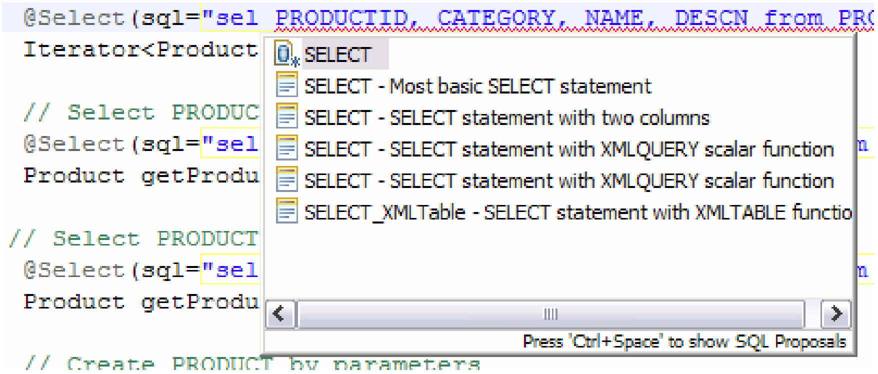
☑ Delete row by parameters

☑ Delete row by object

☐ Use * in SELECT statement to represent all columns

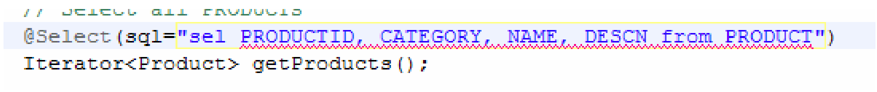
pureQuery Tools

- Generate basic Java data access objects and JUnit
- Integrated Java & SQL editor simplifies development
    - SQL content assist
    - SQL validation
    - SQL execution
    - SQL open definition

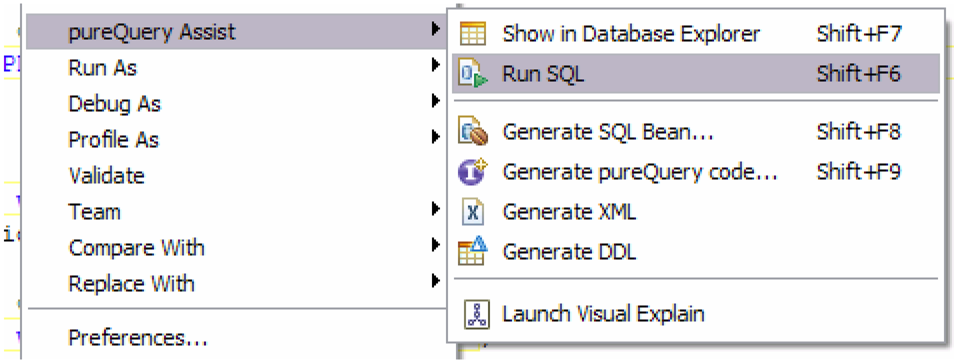


```
@Select(sql="sel PRODUCTID, CATEGORY, NAME, DESCN from PRO
Iterator<Product

// Select PRODUC
@Select(sql="sel
Product getProdu

// Select PRODUCT
@Select(sql="sel
Product getProdu

// Create PRODUCT by parameters
```

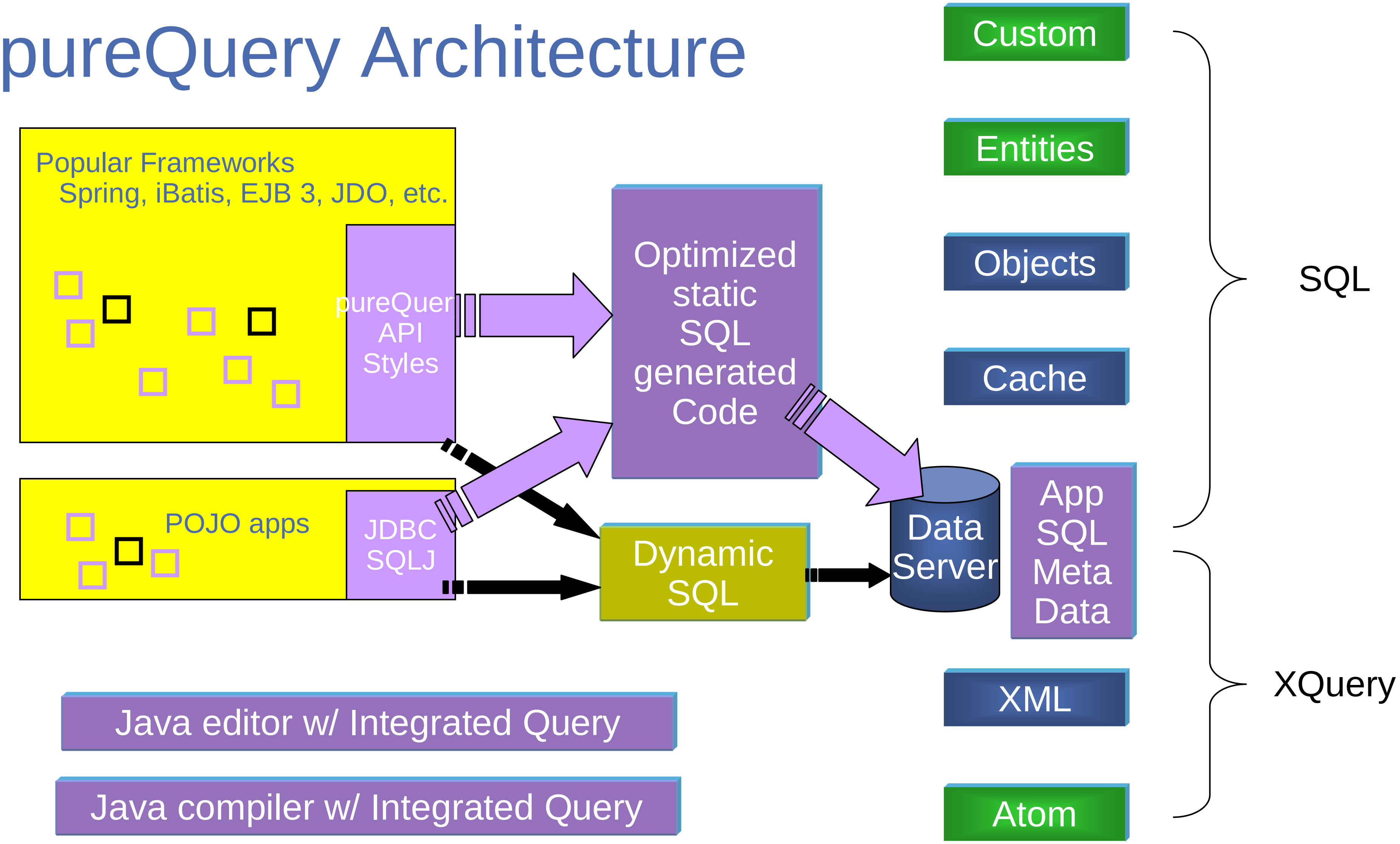
SELECT
SELECT - Most basic SELECT statement
SELECT - SELECT statement with two columns
SELECT - SELECT statement with XMLQUERY scalar function
SELECT - SELECT statement with XMLQUERY scalar function
SELECT_XMLTable - SELECT statement with XMLTABLE functio
Press 'Ctrl+Space' to show SQL Proposals



```
// Select all PRODUCTS
@Select(sql="sel PRODUCTID, CATEGORY, NAME, DESCN from PRODUCT")
Iterator<Product> getProducts();
```

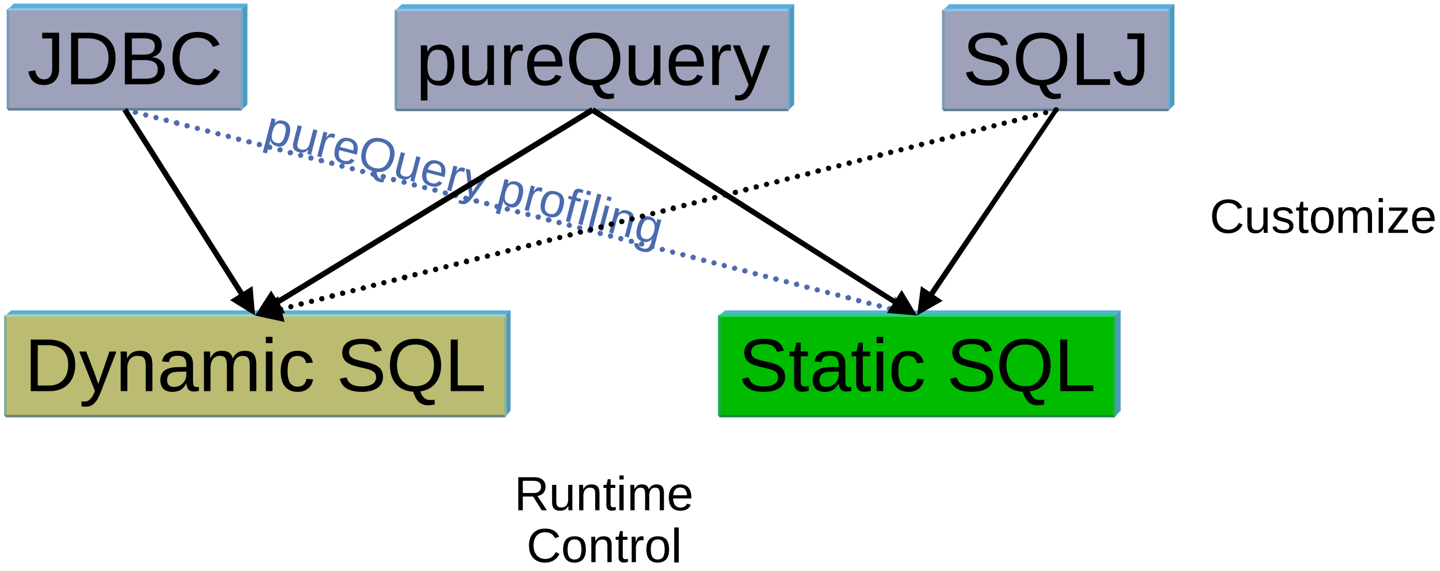


pureQuery Architecture



pureQuery enables wide use of Static SQL

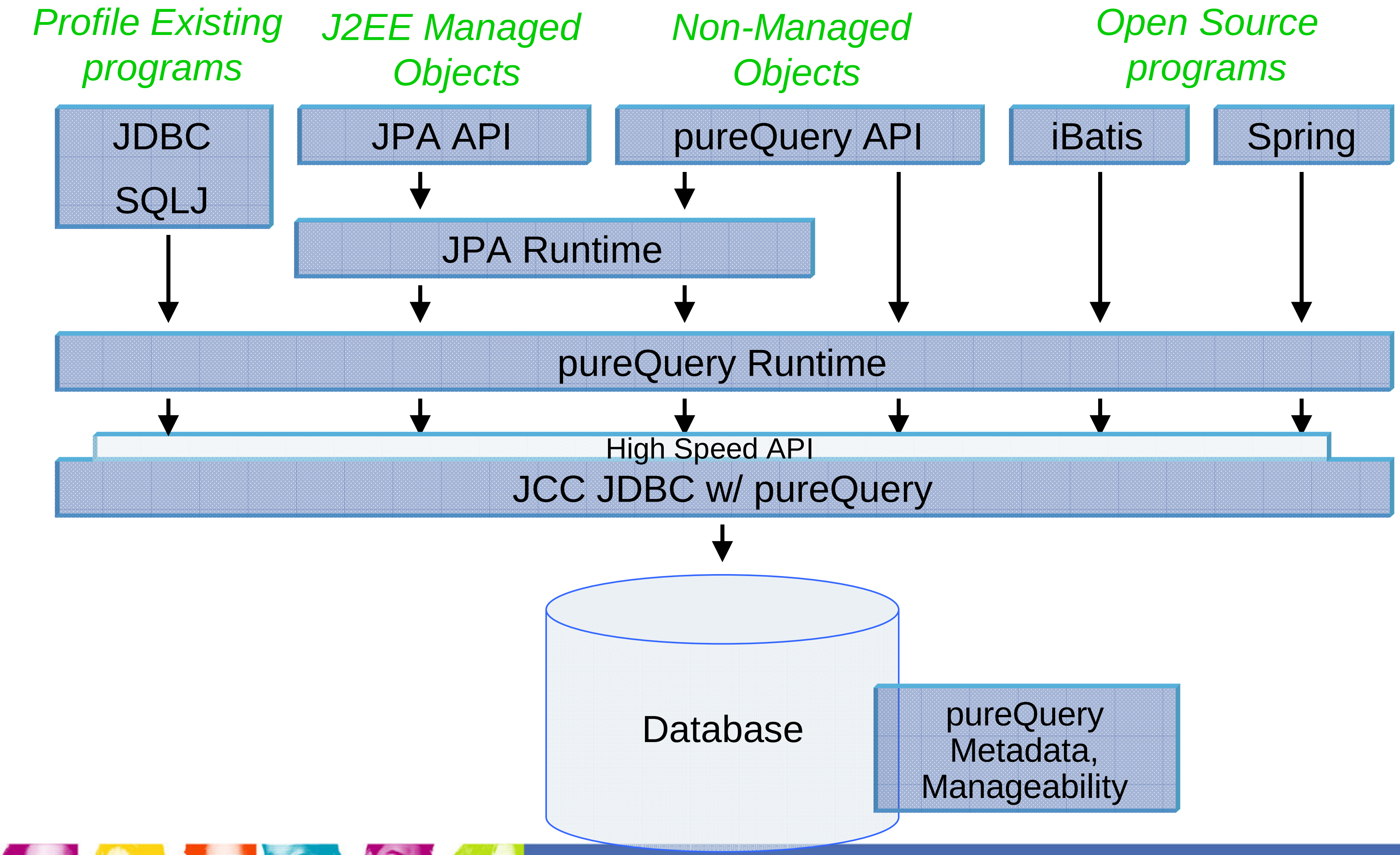
- Static SQL
  - Highest speed
  - Greatest reliability
- JDBC is basic access, uses Dynamic SQL
- SQLJ adds Static SQL
- pureQuery supports both Static SQL and Dynamic SQL
  - Code to dynamic SQL, turn on static SQL at deployment



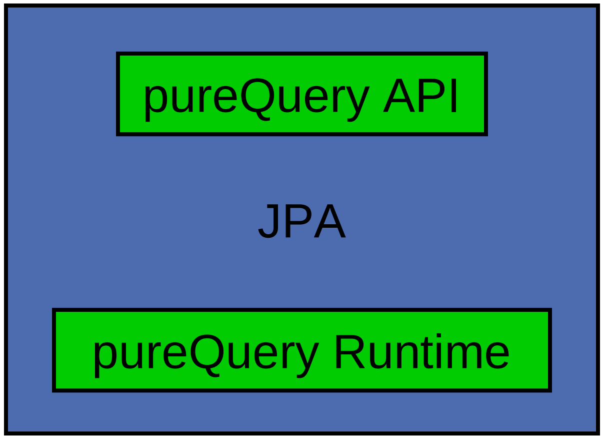
Static SQL Advantages – Comparison with Dynamic SQL

	Dynamic SQL (pureQuery, JDBC)	Static SQL (pureQuery, SQLJ)
Performance	Can approach static SQL performance with help from dynamic SQL caches. Cache misses are costly!	All SQL parsing, catalog access, done at BIND time. Fully optimized during execution.
Access Path Reliability	Unpredictable – Any prepare can get a new access path as statistics or host variables change	Guaranteed – locked in at BIND time All SQL available ahead of time for analysis by EXPLAIN.
Authorization	Pivileges handled at object level. All users or groups must have direct table privileges – Security exposure, and administrative burden	Privileges are package based. Only administrator needs table access. Users/Groups have execute authority. Prevent non-authorized SQL execution.
Montoring, Problem Determiation	Database View is of the JDBC or CLI package – No easy distinction of where any SQL statement came from.	Package View of applications makes it simple to track back to the SQL statement location in the application
Capacity Planning, Forecasting	Difficult to summarize performance data at program level.	Package Level Accounting gives program view of workload to aid accurate forecasting.
Tracking Dependent Objects	No record of which objects are referenced by a compiled SQL statement	Object dependencies registered in database catalog

Java On-Ramps to pureQuery

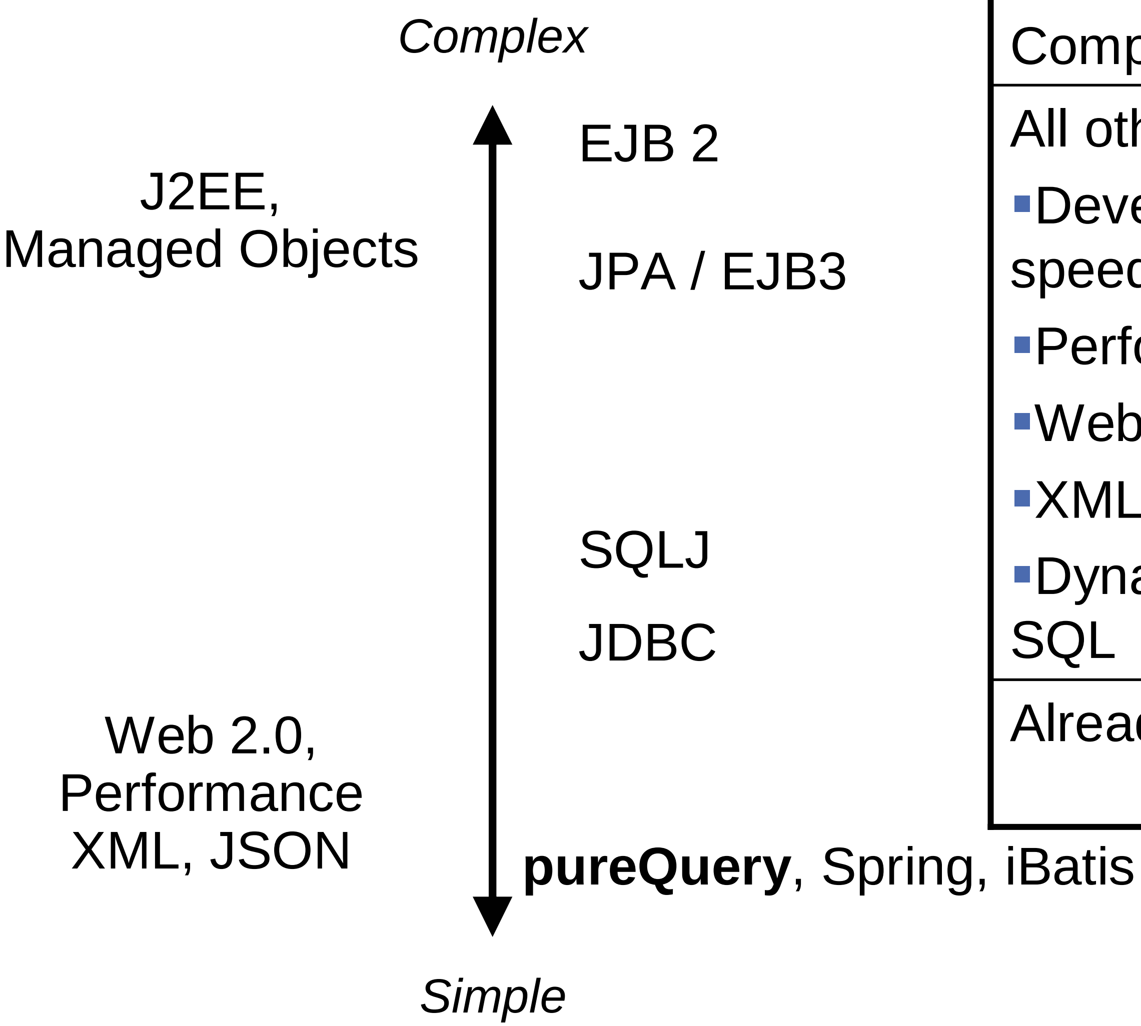


JPA and pureQuery together



- EJBQL and runtime SQL generation based on object manipulation make the value of pureQuery even more important in the JPA setting
- IBM is enhancing our JPA implementation with both pureQuery APIs and pureQuery runtime lifecycle benefits
- JPA w/pureQuery enables problem determination, optimization, and governance connecting the EJBQL and business logic to the actual SQL and database operation
- JPA / EJB3 is a J2EE5 standard
- WebSphere is delivering JPA
- Apache openJPA is the only JPA implementation supported by more than one major vendor: BEA and IBM
- Hibernate users should use the JPA standard APIs and migrate to the openJPA implementation.

Java Data API Space



J2EE Managed Objects Complex O/R	JPA **+ pureQuery**
All other cases ▪Development speed ▪Performance ▪Web 2.0 / AJAX ▪XML / JSON ▪Dynamic & Static SQL	pureQuery
Already using SQLJ	SQLJ **+ pureQuery**

pureQuery API “Styles”

- Support several API styles to fit well into all of the popular Java programming models/frameworks
  - –Inline style (familiar JDBC and SQLJ approach)
  - –Method style (similar to JDBC 4 ease of use enhancements)
  - –Named query style (similar to iBatis/JDO/Hibernate/JPA)

Retrieve a single row from Database

pureQuery:

Automatically Optimizes for 1 row

```
addr = db.queryFirst("SELECT ADDRESS FROM EMP
                WHERE NAME=?name", String.class, name);
-or-
addr = getAddress(name);
```

XML file or Java annotation
  SELECT ADDRESS FROM EMP
    WHERE NAME=?1;

SQLJ:

```
#sql [con] { SELECT ADDRESS INTO :addr FROM EMP
                WHERE NAME=:name };
```

JDBC:

```
java.sql.PreparedStatement ps = con.prepareStatement(
        "SELECT ADDRESS FROM EMP WHERE NAME=?");
ps.setString(1, name);
java.sql.ResultSet names = ps.executeQuery();
names.next();
addr = names.getString(1);
names.close();
```

pureQuery Technology SQL Query API

- Simple, straightforward programming model for data access
  - A fairly thin layer on top of JDBC that simplifies the most common tasks
  - Supports DB2, IDS, Oracle, SQL Server, etc.   (any JDBC database)
  - Out-of-the-box support for storing/retrieving Beans and Maps to/from the database
- Extensible framework
  - Pluggable custom result processing patterns
    - Use Java to implement the mapping behavior instead of a “mapping language”
    - Instantiate result types other than Beans and Maps
  - Framework itself uses the same extension points to provide the out-of-the-box behavior
    - Library of the most common patterns
- Full expressiveness of SQL available
  - In practice, even simple applications do “sophisticated” SQL
- SQL inlined in data access methods
  - Everything that is needed to understand a data access method is in the method