# *Why* pureQuery for Java/.Net and DB2 for z/OS *work so well together…*

Information Management software

By: Paul Wirth wirthp@us.ibm.com

# Disclaimer/Trademarks

Information concerning non-IBM products w as obtained from the suppliers of those products, their published announcements, or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

**All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.**

This information may contain examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious, and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

**Trademarks**

The follow ing terms are trademarks or registered trademarks of other companies and have been used in at least one of the pages of the presentation:

The follow ing terms are trademarks of International Business Machines Corporation in the United States, other countries, or both: AIX, AS/400, DataJoiner, DataPropagator, DB2, DB2 Connect, DB2 Extenders, DB2 OLAP Server, DB2 Universal Database, Distributed Relational Database Architecture, DRDA, eServer, IBM, IMS, iSeries, MVS, Net.Data, OS/390, OS/400, Pow erPC, pSeries, RS/6000, SQL/400, SQL/DS, Tivoli, VisualAge, VM/ESA, VSE/ESA, WebSphere, z/OS, zSeries

Microsoft, Window s, Window s NT, and the Window s logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
Intel and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.
UNIX is a registered trademark of The Open Group in the United States and other countries.
Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
Other company, product, or service names may be trademarks or service marks of others.

# *Capture: SQL Literal Replacement - 2.2 Highlight*


- **Application programs sometimes use literals in SQL excessively**

```
SELECT * FROM CUST_ORDER_HEADER WHERE CUST_CODE = 18107
SELECT * FROM CUST_ORDER_HEADER WHERE CUST_CODE = 414
SELECT * FROM CUST_ORDER_HEADER WHERE CUST_CODE = 351973
SELECT * FROM CUST_ORDER_HEADER WHERE CUST_CODE = 42097
SELECT * FROM CUST_ORDER_HEADER WHERE CUST_CODE = 684
SELECT * FROM CUST_ORDER_HEADER WHERE CUST_CODE = 34098
SELECT * FROM CUST_ORDER_HEADER WHERE CUST_CODE = 95123
SELECT * FROM CUST_ORDER_HEADER WHERE CUST_CODE = 19732804
SELECT * FROM CUST_ORDER_HEADER WHERE CUST_CODE = 40386
SELECT * FROM CUST_ORDER_HEADER WHERE CUST_CODE = 345
SELECT * FROM CUST_ORDER_HEADER WHERE CUST_CODE = 22307
SELECT * FROM CUST_ORDER_HEADER WHERE CUST_CODE = 121201
SELECT * FROM CUST_ORDER_HEADER WHERE CUST_CODE = 3244
SELECT * FROM CUST_ORDER_HEADER WHERE CUST_CODE = 69624
SELECT * FROM CUST_ORDER_HEADER WHERE CUST_CODE = 95141
SELECT * FROM CUST_ORDER_HEADER WHERE CUST_CODE = 77309
```

- **Consolidated SQL can benefit static and dynamic SQL execution**

```
SELECT * FROM CUST_ORDER_HEADER WHERE CUST_CODE = ?
```

- **Benefits:**
  - Enables more SQL for static SQL execution (broader coverage)
  - Improves performance (e.g. lower overhead – no prepare activity)
  - Reduces resource consumption

# *pureQuery and DB2 z/OS*:

**Who will take advantage of pureQuery? LOTS of different folks!**

*pureQuery*

## Java & .Net Programs

- Generate best-in-class Java code and provides SQL coding assist
- pureQuery supports Microsoft .Net Programs C# and VB programs too

## Web Server

- Bind existing Java applications SQL in without access to programs
- pureQuery Runtime located with JDBC Universal Driver and .Net Provider

## DB2 for z/OS

- Reduces Optimizer Processing
- Helps in problem indemnification
- Package Security
- Accounting Reports (Capacity Planning)
- Only thing installed in DB2 z/OS is a good old SQL package

## z/OS Workload Manager(WLM)

- Thread priority management using WLM
- RMF Service Class & Reporting Class management (Capacity Planning)



** Key areas where pureQuery helps

# *pureQuery and DB2 for z/OS*

***"Why use pureQuery with DB2 for z/OS? "***

1) pureQuery can create DB2 packages for Java and .Net programs
   - A package is an object containing a set of SQL statements that is stored in the DB2 Catalog
   - DB2 determines the access plan for the SQL statements in a package when you perform the **bind** process, the access path is **static** and unchanging until a bind is preformed again – providing consistence repeatable performance unlike dynamic SQL which is subject to change
   - Packages as provide **Improved security** to pureQuery, because data access is granted to the package and not tables.
   - pureQuery's packages provide **"*open systems*"** with ***mainframe discipline***, similar to CICS/COBOL
     - pureQuery provides predetermined access paths for consistent operation
     - pureQuery identifies application programs
     - pureQuery access security is granted to a package and not to tables

# *pureQuery and DB2 for z/OS (Continued)*

***"Why use pureQuery with DB2 for z/OS?"***

2) DB2 package names are **unique** to an application program, which will **identify** pureQuery application programs helping in problem determination and monitoring
   - pureQuery provides .Net with **unique static** package names, instead of the normal dynamic SQL package names, like "SYSLN200", which helps in problem determination
   - Unique package names can also be used by **Workload Manager** (WLM) to assign Service Classes to packages, allowing thread transaction level **prioritization**, much like CICS and IMS/TM does
   - Online reporting tools can identify threads by package names improving problem determination of applications
   - RMF Reporting Classes can be created for package names to monitor z/OS hardware resources in capacity planning
   - DB2 accounting record reporting tools can summarize packages for activity monitoring

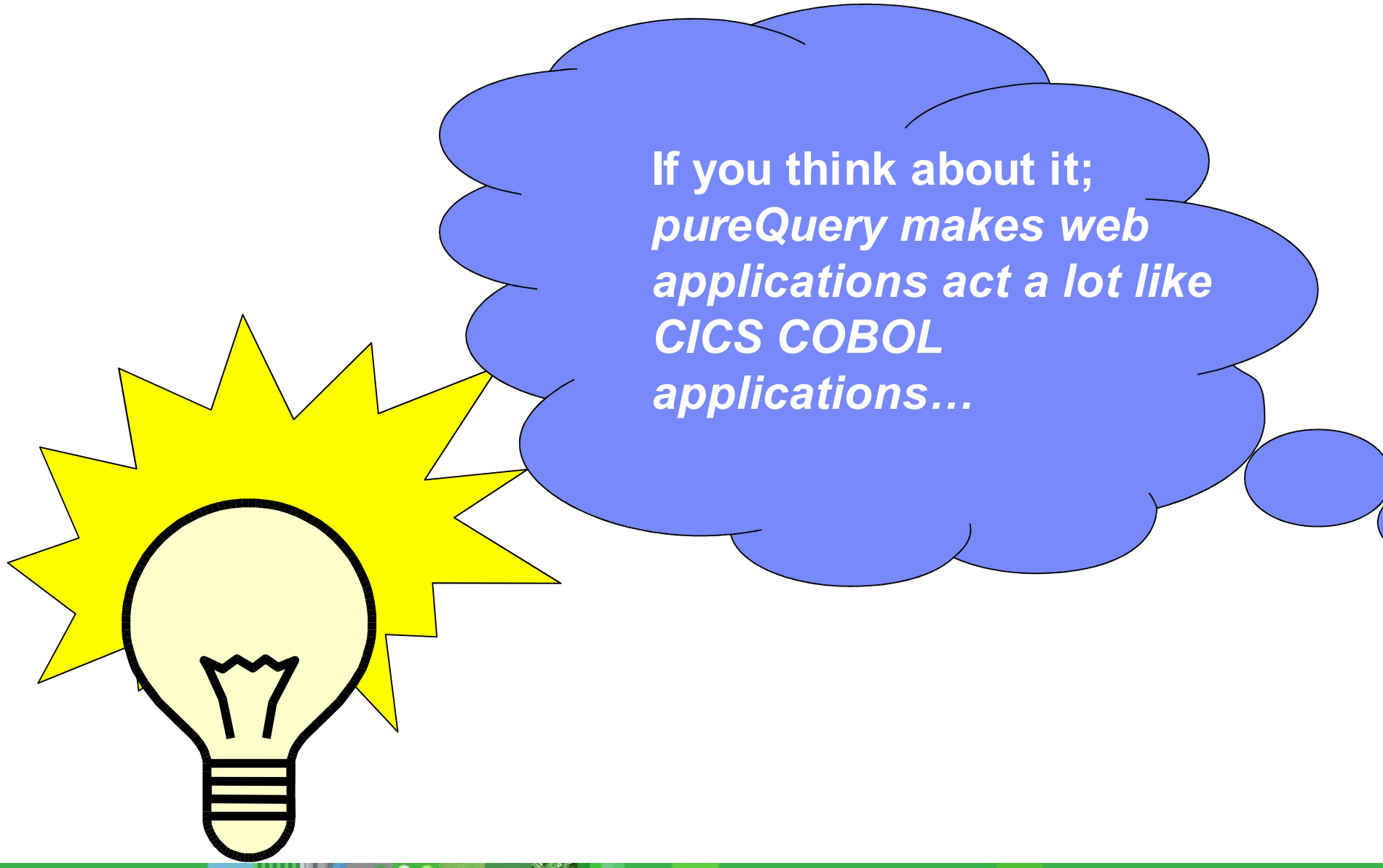
# *pureQuery and DB2 for z/OS (Continued)*

***"Why use pureQuery with DB2 for z/OS?"***

3) pureQuery and stored procedures (SPs)

- SPs have static execution plans, but pureQuery's unique package names can still help with problem determination
- Many customers create SPs without business logic (including only create cursor, open cursor and return code statements) just to provide static execution plans, using pureQuery could avoid the use of stored procedures (SPs) which are used only to facilitate static SQL and avoiding unneeded external Workload Manager (WLM) SPs:
  - Reduces CPU utilization, external SPs use more CPU than threads
  - Increases zIIP eligible workload for DRDA threads

    **Note: DB2 9 for z/OS Native SPs are zIIP eligible for DRDA threads and "can" save CPU over External SPs**
  - Simplifies application development and change management when SQL is included in the program and not using SPs
  - Reduces DB2 z/OS Total Cost of Ownership (TCO)


If you think about it;
pureQuery makes web applications act a lot like CICS COBOL applications…

# Having unique Package names improves application program problem identification…

*With almost all of the dynamic SQL applications using packages like "SYSLNx00" identifying specific programs* ***is difficult****. Unique package names link SQL to Java Class Names, Client Optimization root package names or .Net programs similar to CICS transaction names to programs.*

**Static *pureQuery* Java SQL**

**Dynamic Java SQL**

```
                  ZALLU    VTM     02       V410./C DB1S 09/12/08 11:29:22
> Help PF1      Back PF3       Up PF7      Down PF8     Sort PF10    Zoom F
> T.A                          OMEGAVIEW PA2
>         THREAD ACTIVITY:  Enter a selection letter on the top line.

> *-ALL         B-TSO      C-CICS      D-IMS     E-BACKGROUND   F-DIST ALLI
> G-DIST DBAC   H-UTIL     I-INACT     J-FILTER  K-FUNCTIONS    L-STORED PR
> M-TRIGGERS    N-SYSPLEX  O-ENCLAVES  P-WORKSTA
================================================================================
>                        ALL THREADS CONNECTED TO DB2
PTHDA                                                                      FLT
+                 *
+ Elapsed         Package    CPU    Status       GetPg  Update Commit CORRID
+ ----------      --------   -----  -----------  -----  -----  ------ --------
+ 00:00:13.6      PAW_OR_O   00.0%  IN-DB2         25       0       0 db2jcc_ap
+ 00:02:27.3      SYSLN200   00.0%  IN-DB2        897       0       0 db2jcc_ap
+ 00:02:52.3      SYSLN200   00.0%  IN-DB2       1025       0       0 db2jcc_ap
+ 00:03:05.8      SYSLN200   00.0%  IN-DB2       1324       0       0 db2jcc_ap
+ 00:02:32.7      SYSLN200   00.0%  IN-DB2        961       0       0 db2jcc_ap
+ 00:02:59.2      SYSLN200   00.0%  IN-DB2       1046       0       0 db2jcc_ap
================================================================================
```

# *Tip: Adding Java clientProgramName will set Correlation Id.*

```
ZALLU    VTM    02    V410./C DB1S 09/22/08 17:04:53   2
```

```
java.util.Properties connectionProperties = new java.util.Properties ();
connectionProperties.put ("user", user);
connectionProperties.put ("password", password);
// Setting Corr. Id
  connectionProperties.put("clientProgramName","ME_PAW");
if (!url.contains("emulateParameterMetaDataForZCalls"))
  connectionProperties.put ("emulateParameterMetaDataForZCalls", "1");
if (!url.contains("retrieveMessagesFromServerOnGetMessage"))
  connectionProperties.put ("retrieveMessagesFromServerOnGetMessage", "true");
// connectionProperties.put ("traceFile", "/temp/trace.txt")

try {
  Connection connection = DriverManager.getConnection (url, connectionProperties);
  System.out.println ("Successfully Connected to " + connection.getMetaData ().getDatabasePro
  return connection;
  }
```

| Elapsed | Package | CPU | Status | GetPg | Update | Commit | CORRID |
|---|---|---|---|---|---|---|---|
| 00:02:46.5 | SYSLN200 | 00.0% | IN-DB2 | 1805 | 0 | 0 | db2jcc_appli |
| 00:02:32.4 | SYSLN200 | 00.0% | IN-DB2 | 1615 | 0 | 0 | db2jcc_appli |
| 00:02:23.1 | SYSLN200 | 00.0% | IN-DB2 | 1572 | 0 | 0 | db2jcc_appli |
| 00:00:40.9 | SYSLN200 | 00.0% | IN-DB2 | 286 | 0 | 0 | db2jcc_appli |
| 00:00:21.3 | SYSLN200 | 00.0% | IN-DB2 | 151 | 0 | 0 | ME_PAW |
| 00:00:03.9 | SYSLN200 | 00.0% | IN-DB2 | 66 | 0 | 0 | db2jcc_appli |

# *Having unique Package names improves application program problem identification…*

*With almost all of the dynamic SQL applications using packages like "SYSxxx00" identifying specific programs* ***is difficult****. Unique package names link SQL to .Net programs similar to CICS transaction names to programs.*

**Dynamic .Net SQL**

```
> *-All-Idle    B-TSO      C-CICS      D-IMS     E-Background   F-Dist Allied
> G-Dist DBAC   H-Util     I-Inact     J-Filter  K-Functions    L-Stored Proc
> M-Triggers    N-Sysplex  O-Enclaves  P-Worksta Q-All+Idle
================================================================================
>                    Threads Summary Excluding Idle Threads
PTHDA                                                                  FLTR ON
+ *
+ Elapsed       Package   CPU    Status       GetPg  Update Commit CORRID/JOBN
+ ----------    --------  -----  -----------  ------ ------ ------ ------------
+ 00:02:12.6    SYSSH200  00.0%  WAIT-REMREQ      28      0      0 db2Commands.
================================================================================
```

**Static *pureQuery* .Net SQL**

```
================================================================================
>                    Threads Summary Excluding Idle Threads
PTHDA                                                                  FLTR ON
+ *
+ Elapsed       Package   CPU    Status       GetPg  Update Commit CORRID/JOBN
+ ----------    --------  -----  -----------  ------ ------ ------ ------------
+ 00:00:07.0    PAWTST1   00.0%  WAIT-REMREQ       3      0      0 db2Commands.
================================================================================
```

# *Interesting comparison of .Net dynamic and static SQL CPU utilization*

***Note: You mileage may very!***

**Dynamic .Net SQL**

Initial Execution

```
                                THREAD PACKAGE SUMMARY
 PLAN
+ Thread:  Plan=DISTSERV  Connid=SERVER   Corrid=db2Commands. Authid=DBA031
+ Dist   :  Type=DATABASE ACCESS, Luwid=G9274493.H546.C3A4177E4D37=569
+ Location  :  NDCDB205
 pkl

 pkg
+
+ Package/            SQL        In-DB2        In-DB2                  In-DB2
+     DBRM         Requests  Elapsed Time      CPU Time     Waits     Wait Time
+ --------      ----------  ------------  ------------  --------  ------------
+ SYSSH200   *          4  00:00:00.224  00:00:00.005        26  00:00:00.160
```

Subsequent Executions

```
+
+ Package/            SQL        In-DB2        In-DB2                  In-DB2
+     DBRM         Requests  Elapsed Time      CPU Time     Waits     Wait Time
+ --------      ----------  ------------  ------------  --------  ------------
+ SYSSH200   *          4  00:00:00.007  00:00:00.000         4  00:00:00.005
```

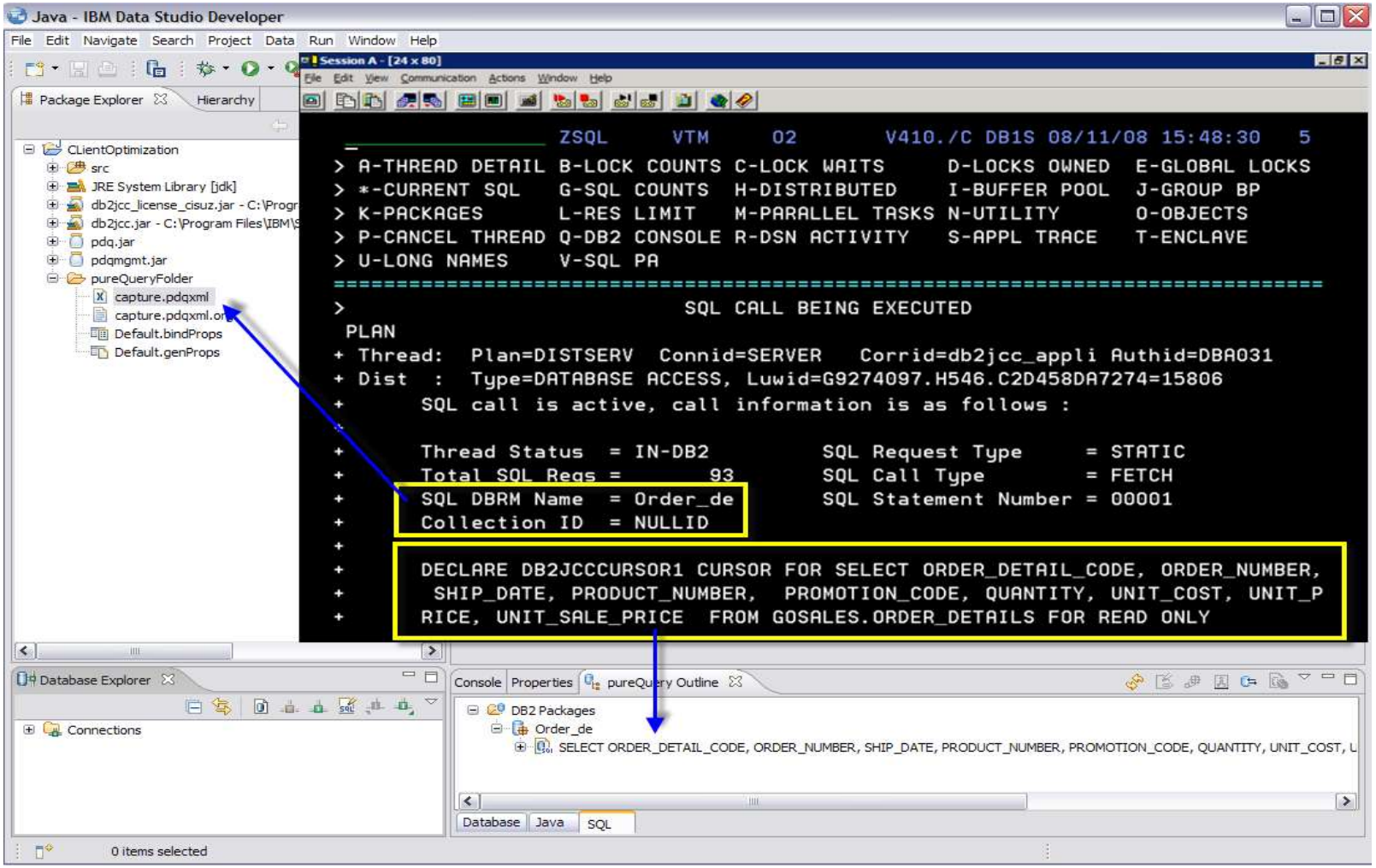
**Static *pureQuery* .Net SQL**

```
  pkl

  pkg
 +
 + Package/            SQL        In-DB2        In-DB2                  In-DB2
 +     DBRM         Requests  Elapsed Time      CPU Time     Waits     Wait Time
 + --------      ----------  ------------  ------------  --------  ------------
 + PAWTST1   *          3  00:00:00.002  00:00:00.000       N/A           N/A
```

# *Problem Determination*

## Correlate Package and SQL With Captured Metadata



# Comparison of CPU usage with pureQuery statically bound Stored Procedures

*SYSSTAT is charged for CPU time outside of a stored procedure, when dynamically executed. This CPU time includes the short prepare of the CALL statement in the SYSSTAT package, any authorizations for the SYSSTAT and procedure package, any scheduling CPU associated with WLM, and any result set data that's returned. Basically, any CPU time outside the actual stored procedure itself that is used for preparation or data retrieval is being charged to SYSSTAT. In the case of very simple stored procedures (especially those that return result sets), the CPU for the SYSSTAT package can be a significant percentage of the total CPU time.*

*pureQuery can help reduce the overhead of short prepares and authorization processing with stored procedures, when executed statically. The percent of of SYSSTAT overhead will differ, so your mileage will be different. Note below that "SYSSTAT" is not present in Static Packages, and the overhead is recorded to the static package name of "PAW_Empl" for the stored procedure "EMPL_DEP"*

**Dynamic Execution**

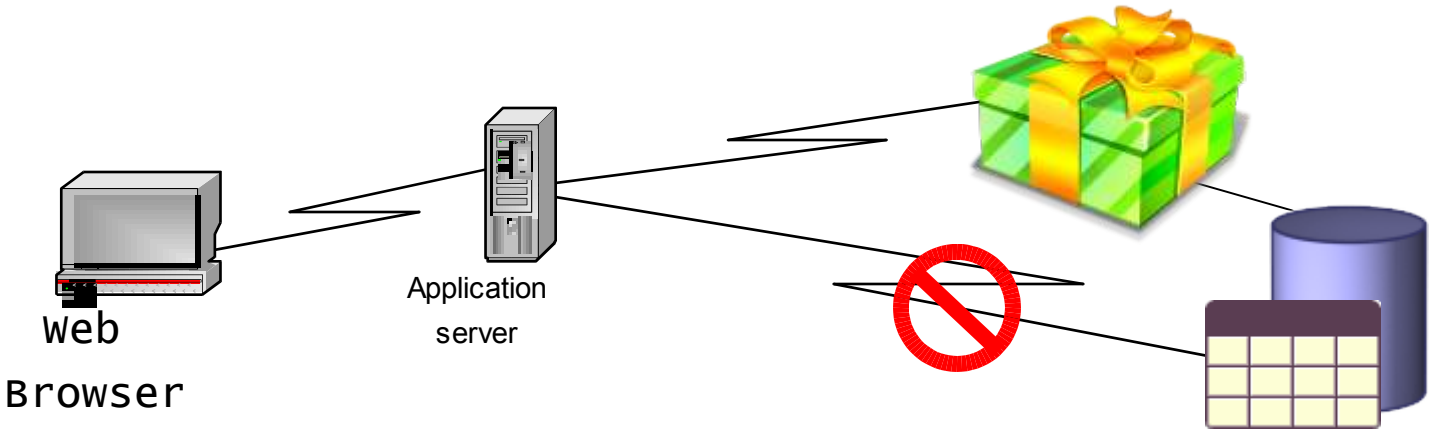
```
PLAN
+ Thread:  Plan=DISTSERV  Connid=SERVER   Corrid=db2jcc_appli Authid=DBA031
+ Dist   :  Type=DATABASE ACCESS, Luwid=G9274493.H546.C4EFDD7D7DF4=14394
+ Location  :  9.73.90.175
 pkl
 pkg
+
+ Package/            SQL          In-DB2          In-DB2                    In-DB2
+     DBRM         Requests   Elapsed Time      CPU Time     Waits      Wait Time
+ --------      ----------  -------------  -------------  --------   -------------
+ SYSSTAT             1000   00:00:00.152   00:00:00.065         1   00:00:00.000
+ EMPL_DEP   *        2000   00:00:00.295   00:00:00.120         1   00:00:00.003
```

**Static Execution**

```
 pkg
+
+ Package/            SQL          In-DB2          In-DB2                    In-DB2
+     DBRM         Requests   Elapsed Time      CPU Time     Waits      Wait Time
+ --------      ----------  -------------  -------------  --------   -------------
+ PAW_Empl             500   00:00:00.061   00:00:00.029        N/A            N/A
+ EMPL_DEP   *        2000   00:00:00.277   00:00:00.122        N/A            N/A
```

# *pureQuery using Packages for Security*

- Granting user access to pureQuery packages instead of tables avoids direct access to data, similar to CICS and COBOL programs
- Restricting generalize access to tables limits exposure to unplanned access to data
- Data access is restricted to the SQL statements in the package data
- Data locking can also be controlled using packages



# Workload Manger (WLM) Classification - Java

DB2's Type-4 Universal Driver for distributed Java applications provide properties for the connection class to help WLM classify work, but are not often used (*setClientUser, setClientApplicationInformation, setClientWorkStation and setClientAccountingInformation)*

pureQuery's unique package names can provide WLM classification without programmers needing to including properties

```
_____________ ZENCLD   VTM     02       V410./C DB1S 09/12/08 11:46:3
+
+ CLASSIFICATION WORK QUALIFIERS
+ Subsystem Type:    DDF                  Correlation:        DB2JCC_APPLI
+ Proc Name:                              Trans Program Name:
+ UserId:           DBA031                Transaction Class:
+ Network ID:                             Logical Unit Name:
+ Plan Name:        DISTSERV              Package Name:       PAW_OR_0
+ Connection:       SERVER                Collection:         NULLID
+ Function Name:    DB2_DRDA              Subsystem Name:     DB1S
+ Accounting Info:  JCC03520IBM-3D70      Subsystem Parm:     DBA031
+ Perform:                                Subsystem Priority: N/A
+ Scheduling Env:                         Subsys Coll Name:   DSNSG
+ Process Name:     DB2JCC_APPLICATION
+
+ Performance Index Input Data for Velocity Goal
+ Total Usings:          3930             Total Delays:           712
================================================================================
```

# *Workload Manger (WLM) Classification - .Net*

DB2's Driver for distributed applications provide properties for the connections to help WLM classify work, but are not often used. pureQuery's unique package names can provide WLM classification without programmers needing to including properties

```
                    ZENCLD   VTS     02       V410./C DB1S 03/11/09 13:09:3
+
+ CLASSIFICATION WORK QUALIFIERS
+ Subsystem Type:     DDF                    Correlation:         DB2COMMANDS.
+ Proc Name:                                 Trans Program Name:
+ UserId:            DBA031                 Transaction Class:
+ Network ID:                                Logical Unit Name:
+ Plan Name:         DISTSERV               Package Name:        PAWTST1
+ Connection:        SERVER                 Collection:          TESTA
+ Function Name:     DB2_DRDA               Subsystem Name:      DB1S
+ Accounting Info:   SQL09053NT             Subsystem Parm:      DBA031
+ Perform:                                   Subsystem Priority:  N/A
+ Scheduling Env:                           Subsys Coll Name:    DSNSG
+ Process Name:      DB2COMMANDS.VSHOST.E161C1754000
+
```

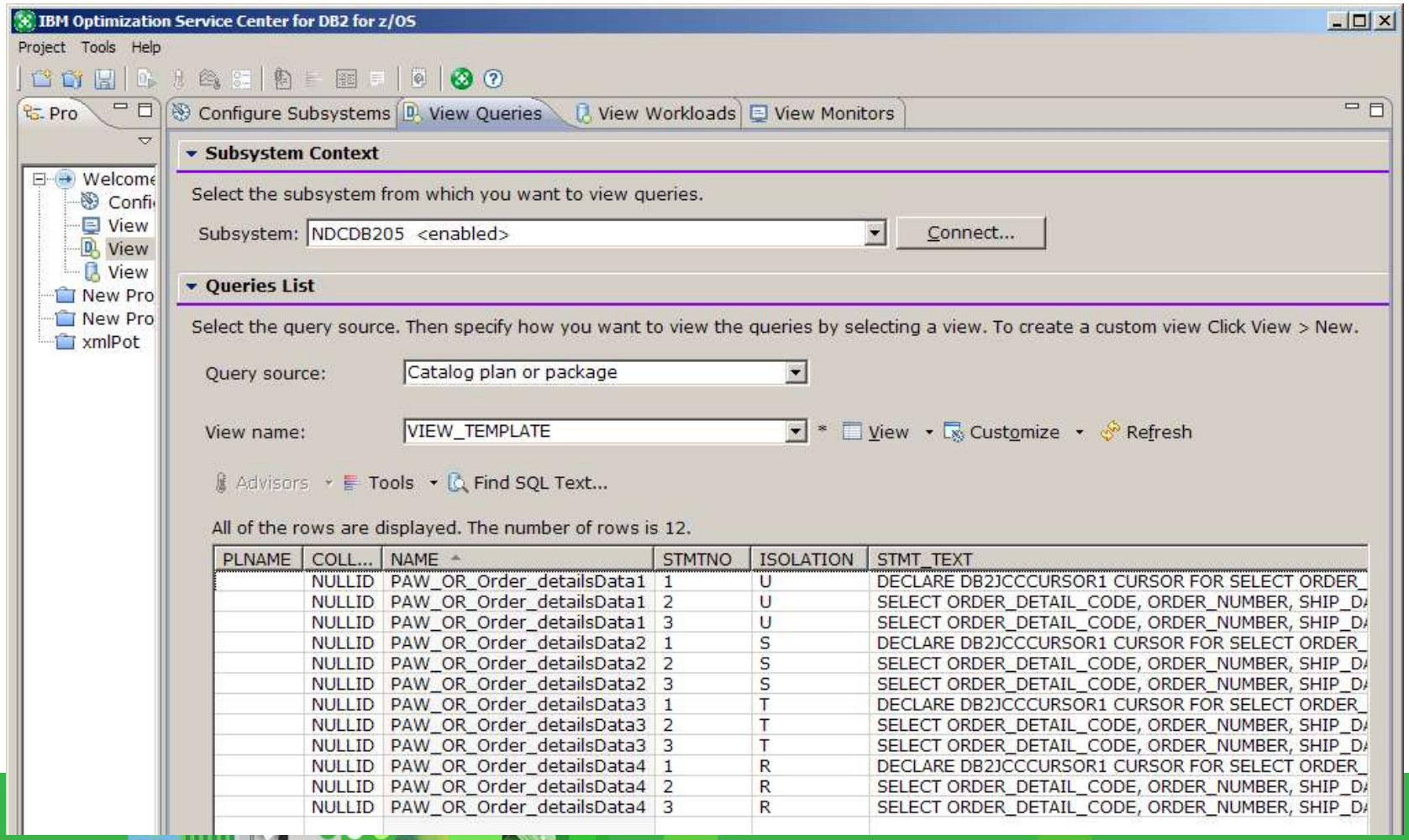
# *Example: Without proper WLM classification package PAW_OR_O is using the default DDF service class*

```
                  ZENCLD   VTM     02       V410./C DB1S 09/12/08 11:46:39  18
+ ENCLAVE TOKEN:      500000E600          Enclave Type:        Original Indep
+ Owning System:      DEMOMVS             Owning Job:          DB1SDIST
+ WLM Mode:           Goal                Enclave CPU Time     00:00:01.374
+
+ SERVICE PERIOD INFORMATION
+ Period(s) for Service Class DDFDEF:     2
+ Current Period for This Thread:         2
+ Performance Index This Period:             .24
+
+ +----------------------------- Period Number
+ | +---------------------------- Importance
+ | | +-------------------------- Duration in Service Units
+ | | |      +------------------- Percentile
+ | | |      |     +------------- Response Time Goal or Velocity
+ | | |      |     |      +------ Response Time Unit
+ | | |      |     |      |      Goal Description
+ - - +----  +---  +----  +------- -------------------
+ 1 3   500   80%  2500 MilliSecs Percentile Response Time Goal
+ 2 4              .20            Velocity Goal
+
+ SERVICE CLASS INFORMATION
+ CPU Critical:       No                  Storage Protection: No
+
```

***Ouch!***
*My importance is only 4 and I'm using the DDFDEF service class*

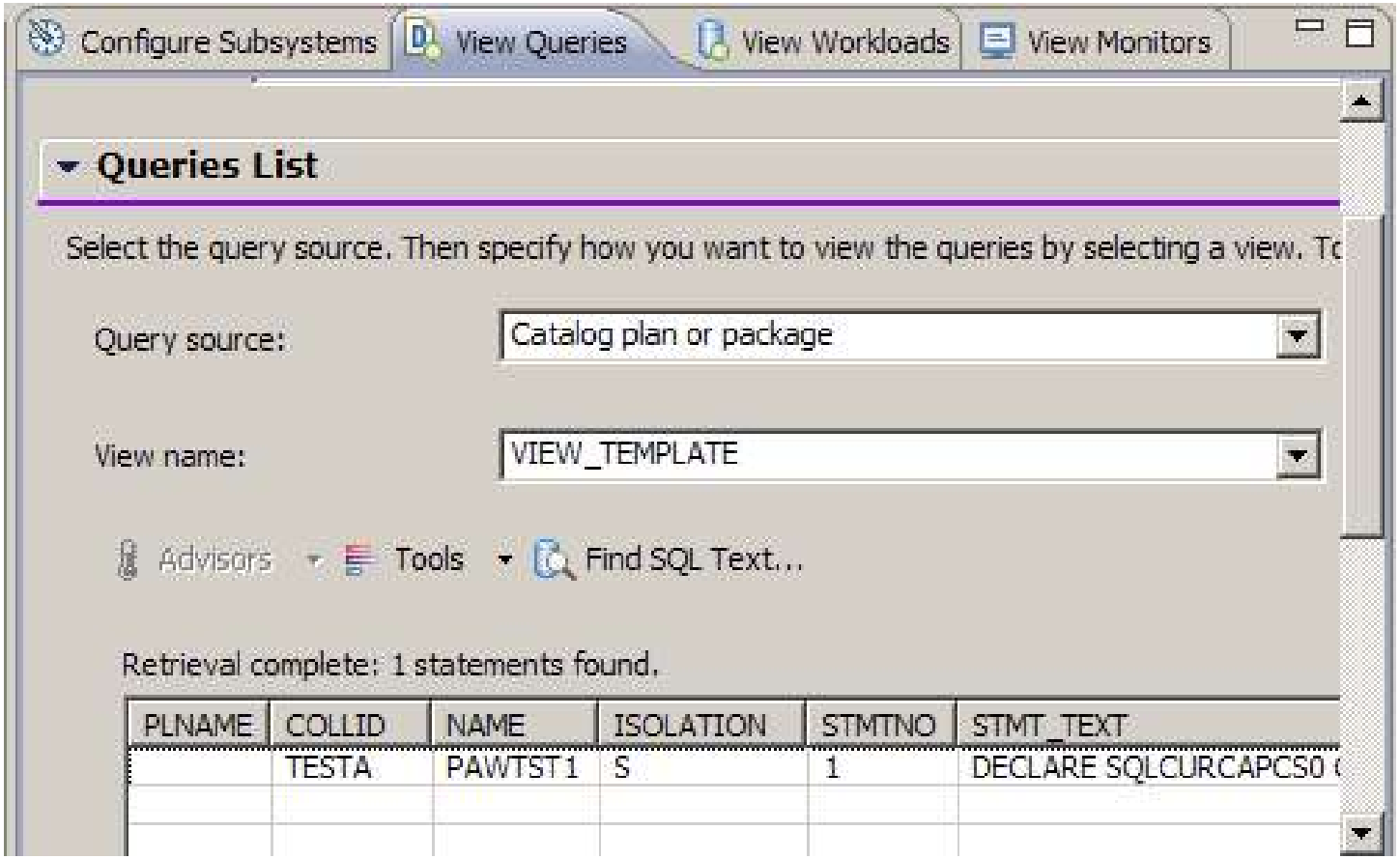
# Optimization Service Center and Visual Explain can access the pureQuery static packages in the DB2 catalog because they are just plain “old” packages

**(Note: By default pureQuery created packages for all 4 isolation levels)**




| PLNAME | COLL... | NAME | STMTNO | ISOLATION | STMT_TEXT |
|---|---|---|---|---|---|
| | NULLID | PAW_OR_Order_detailsData1 | 1 | U | DECLARE DB2JCCCURSOR1 CURSOR FOR SELECT ORDER_ |
| | NULLID | PAW_OR_Order_detailsData1 | 2 | U | SELECT ORDER_DETAIL_CODE, ORDER_NUMBER, SHIP_DA |
| | NULLID | PAW_OR_Order_detailsData1 | 3 | U | SELECT ORDER_DETAIL_CODE, ORDER_NUMBER, SHIP_DA |
| | NULLID | PAW_OR_Order_detailsData2 | 1 | S | DECLARE DB2JCCCURSOR1 CURSOR FOR SELECT ORDER_ |
| | NULLID | PAW_OR_Order_detailsData2 | 2 | S | SELECT ORDER_DETAIL_CODE, ORDER_NUMBER, SHIP_DA |
| | NULLID | PAW_OR_Order_detailsData2 | 3 | S | SELECT ORDER_DETAIL_CODE, ORDER_NUMBER, SHIP_DA |
| | NULLID | PAW_OR_Order_detailsData3 | 1 | T | DECLARE DB2JCCCURSOR1 CURSOR FOR SELECT ORDER_ |
| | NULLID | PAW_OR_Order_detailsData3 | 2 | T | SELECT ORDER_DETAIL_CODE, ORDER_NUMBER, SHIP_DA |
| | NULLID | PAW_OR_Order_detailsData3 | 3 | T | SELECT ORDER_DETAIL_CODE, ORDER_NUMBER, SHIP_DA |
| | NULLID | PAW_OR_Order_detailsData4 | 1 | R | DECLARE DB2JCCCURSOR1 CURSOR FOR SELECT ORDER_ |
| | NULLID | PAW_OR_Order_detailsData4 | 2 | R | SELECT ORDER_DETAIL_CODE, ORDER_NUMBER, SHIP_DA |
| | NULLID | PAW_OR_Order_detailsData4 | 3 | R | SELECT ORDER_DETAIL_CODE, ORDER_NUMBER, SHIP_DA |

# Optimization Service Center and Visual Explain can access the pureQuery static packages in the DB2 catalog because they are just plain “old” packages



# *pureQuery and DB2 z/OS are a great team:*

- In the past CICS, IMS/TM(DC) and JES provided transaction management and workload prioritization for DB2, but in today's distributed processing environment DB2 is doing transaction management.
- As a distributed transaction manager, DB2 uses CONDBAT and MAXDBAT to manage the number of connections and transactions/threads in the subsystem. And Workload Manager (WLM) provides DB2's transaction/thread workload prioritization. pureQuery complements DB2 distributed transaction management by:
  - Identifying transactions/threads helping WLM workload management
  - Helping in problem determination by linking SQL to a program
  - Improves security by restricting data access to a package
  - Providing many improvements in CPU usage (Static SQL vs Dynamic SQL, and may reduce number of stored procedures needed)
  - pureQuery also works with all versions of DB2 for z/OS, and no additional software will be installed in DB2 z/OS (just the package)
- pureQuery helps in workload consolidation, resource sharing, virtualization and can be implemented without changing existing applications
- Application development standards are hard to maintain today with frequent changes in staff, contract programming and outsourcing. pureQuery helps in the standardization of program coding, implementation and application management.

# *Summary:*

**pureQuery client optimization features for .NET applications enable you to improve Quality of Service and efficiency:**

- **Achieve improved and predictable SQL performance**
- **Enhance database security**
- **Simplify security administration**
- **Deliver better problem determination support**

**…. without changing a line of code**

# *Resources*

- **IBM Data Studio pureQuery Runtime web page**

  www.ibm.com/software/data/studio/purequery/

- **Articles and tutorials covering the Data Studio portfolio**

  **www.ibm.com/developerworks/**

  – Coming soon: Optimize your existing .NET application using pureQuery

- **IBM Data Studio community (links to articles, downloads, blogs, …)**

  www.ibm.com/developerworks/spaces/datastudio

- **For more information contact the following Data Studio Enablement team members:**

  – Patrick Titzler (ptitzler@us.ibm.com)

  – Vijay Bommireddipalli (vijayrb@us.ibm.com)

***Thank You***

***Questions?***