

DB2 und Anwendungsoptimierung mit Optim

Heidrun Wietzorek, wietzo@de.ibm.com

Christian Daser, chris.daser@de.ibm.com



Agenda: DB2 Anwendungsoptimierung mit Optim



Optim Query Workload Tuner

DB2 z/OS Themen

Query Tuning / Optimizer / DataStudio

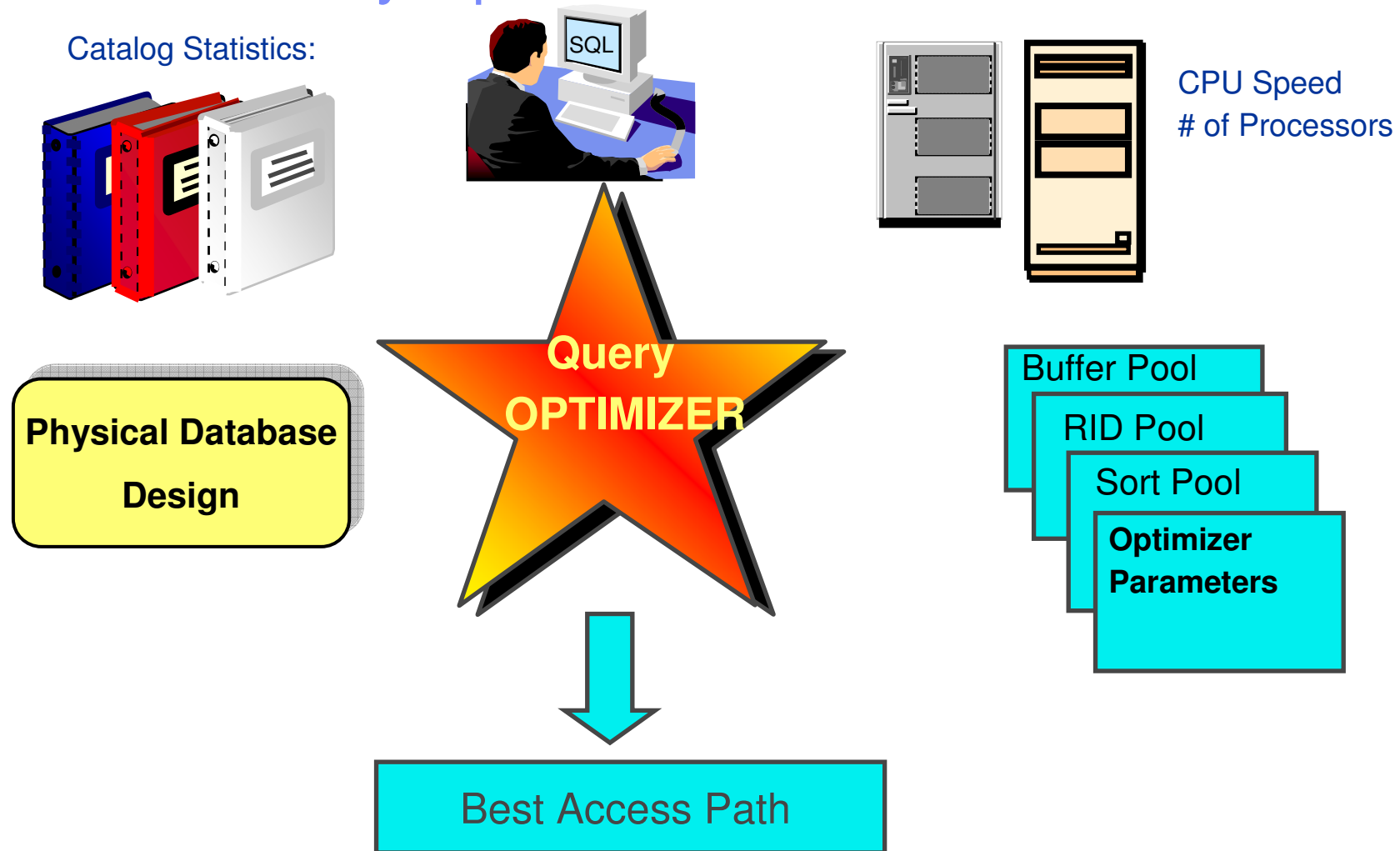
Acceleratoren

Java Zugriffe: Optim pureQuery Runtime

Motivation for Query Tuning

- Application Performance
- SLAs & Batch Windows
- System Resources
 - Software Licenses
 - Capacity Upgrades
- High Manual effort
 - Access Path Analysis
 - Dynamic SQL
 - Application Design & DDL for Portability
 - Differences between LPARs and DB2 Versions and PTFs

Overview Query Optimization



Single Query and Workload

Development Environment



InfoSphere Optim Query Tuner

Tune SQL pre-production while costs and impact are low

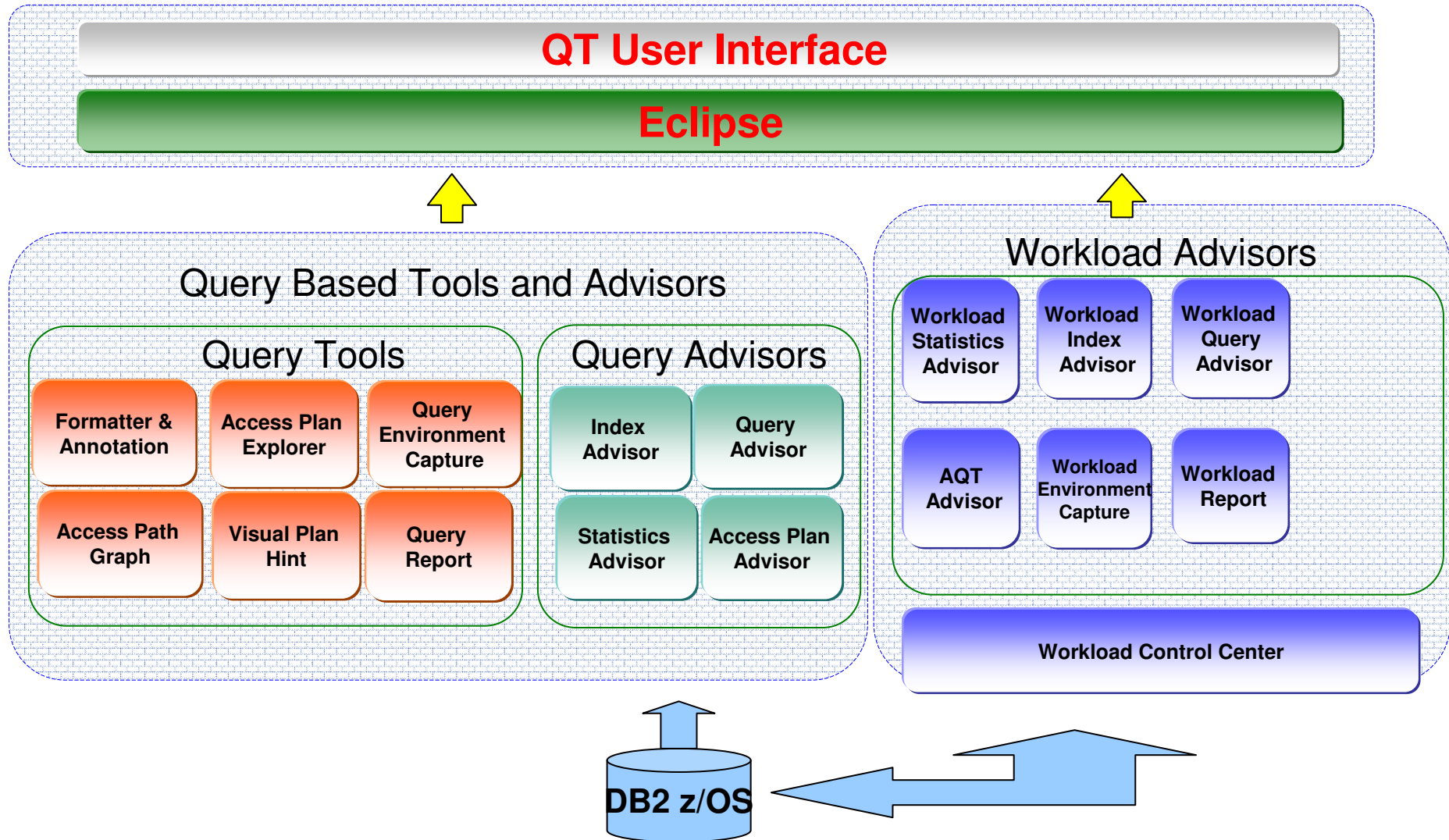
Production Environment



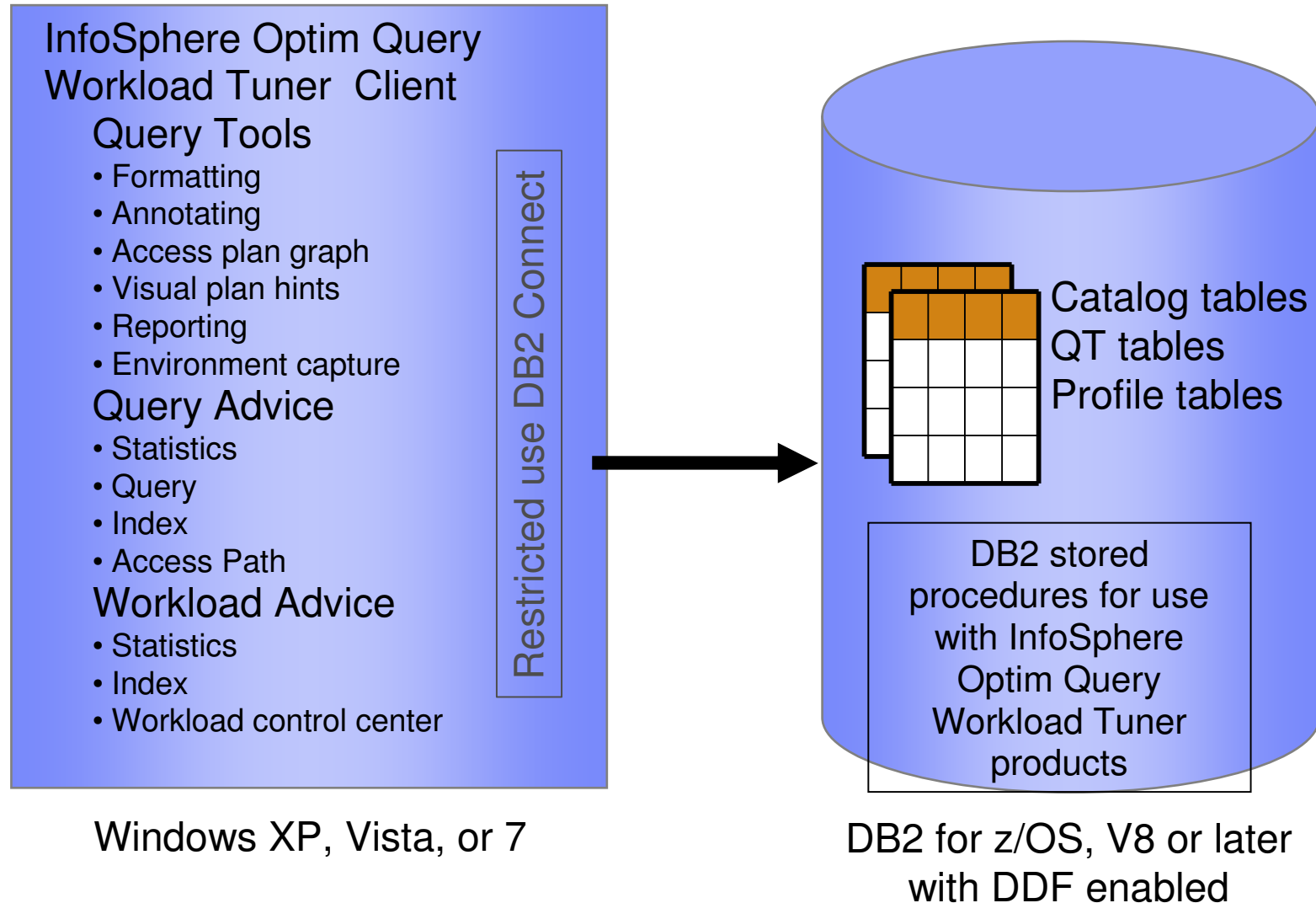
Optim Query Workload Tuner

Optimize workload for peak performance

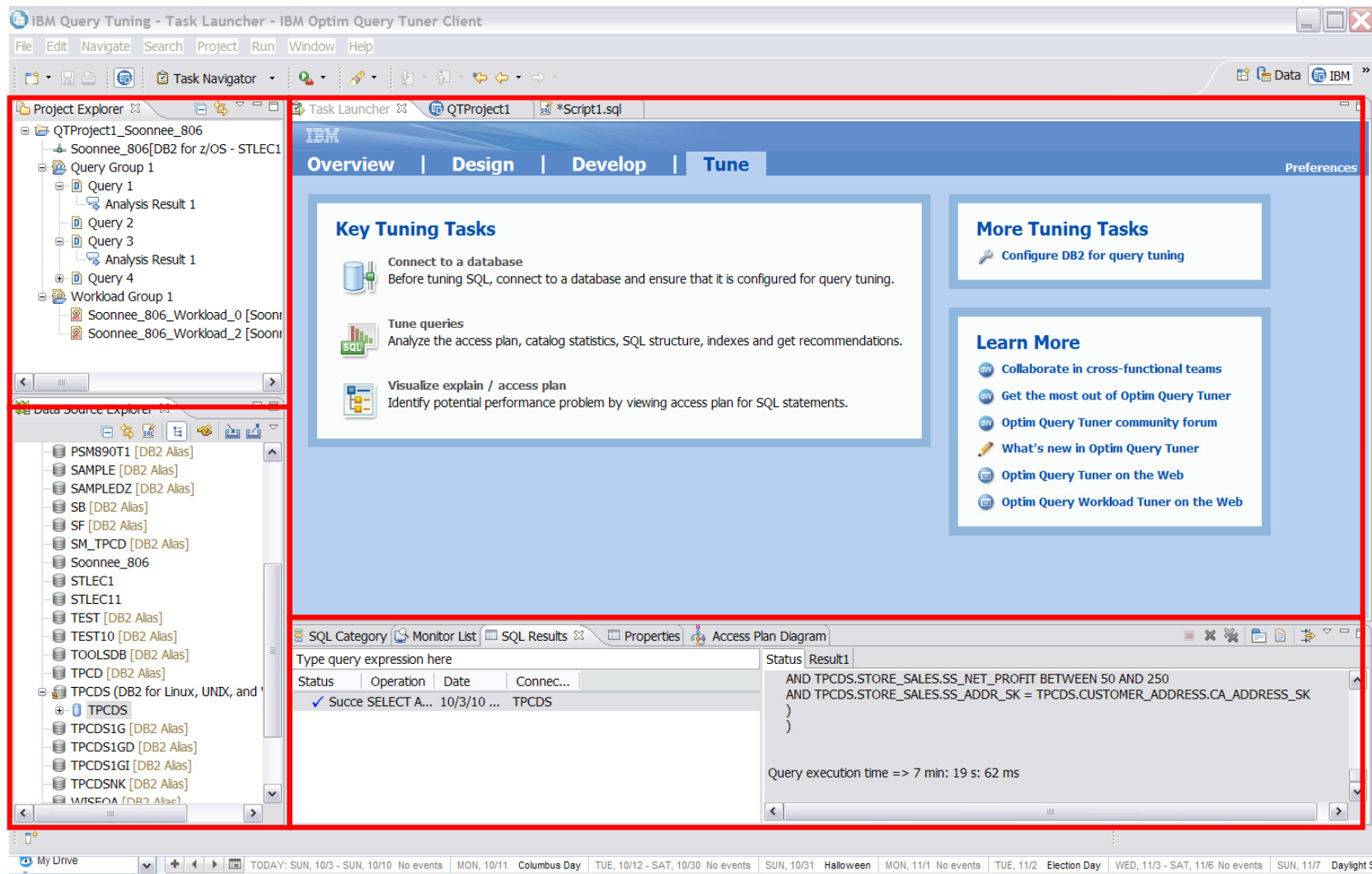
QQWT: Features for DB2 z/OS



Query Tuner Architecture



‘Getting Started’ with OQT/OQWT



Expert-Enabled Query Tuning Out of the Box

- **Identify query candidates from numerous sources**

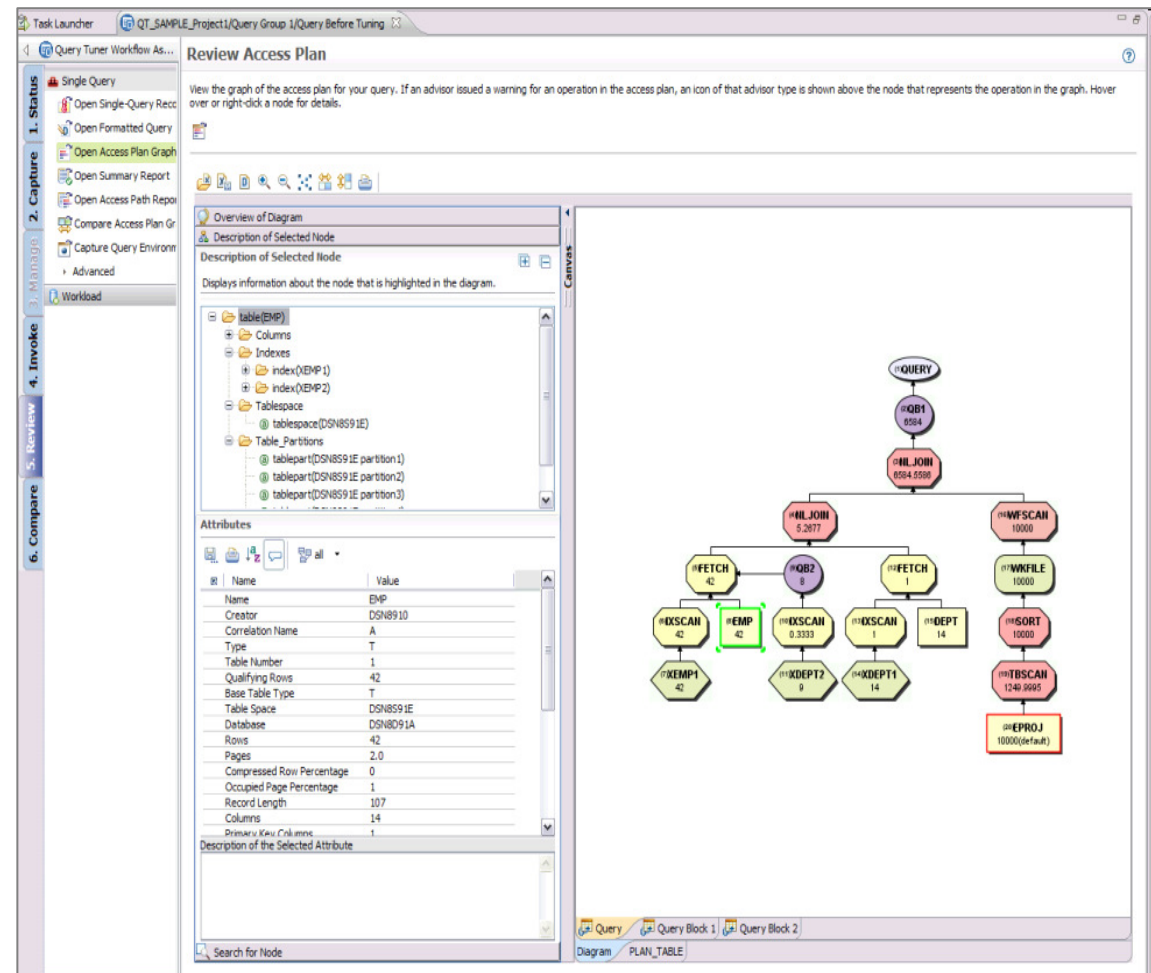
- DB2 catalog
- Dynamic statement cache
- Development Studio hot spots
- Query or performance monitors

- **Facilitate analysis**

- Query formatting
- Query annotation
- Access path visualization and annotation

- **Get expert tuning advice**

- Improve query design
- Improve statistics quality
- Improve database design



Streamlined Analysis with workflow assistant

Define or select a workload

Execute Advisors

Drill Down into advice

Recommendations - 1 In

Advisor	Number	Priority	Description
Statistics Advisor	1	HIGH	Repair statistics problems for this query. Gather missing statistics. Recollect conflicting st...
Query Advisor	2	MEDIUM	Provide a join predicate based on the referential constraint between tables DSN8910.EM...
Query Advisor	3	MEDIUM	Provide a predicate on column WORKDEPT.
Access Path Advisor	4	LOW	Avoid reading all index keys on an index scan (QBLOCKNO = 1, PLANNO = 1).
Index Advisor	5	LOW	Index recommendations found.

Validate improvement

Name	Summary Status	Owner	Execution Time
WorkloadWithTypicalStats	ANALYZING	B3OSC12	CPU time: 97.32 (second...
WorkloadTunedWithStatsAdvisor	ANALYZING	B3OSC12	CPU time: 53.19 (second...
WorkloadTunedWithIndexAdvisor	ANALYZING	B3OSC07	CPU time: 40.67 (second...
AbsoluteCPUTimeExceptionMonitor	ENABLED/STARTED	SYSADM	N/A
NormalMonitor	ENABLED/STARTED	SYSADM	N/A

Gather High Cost Queries and Workloads

DB2

- Plan table
- Statement table
- Catalog plan or package
- Statement cache

Optim Development Studio

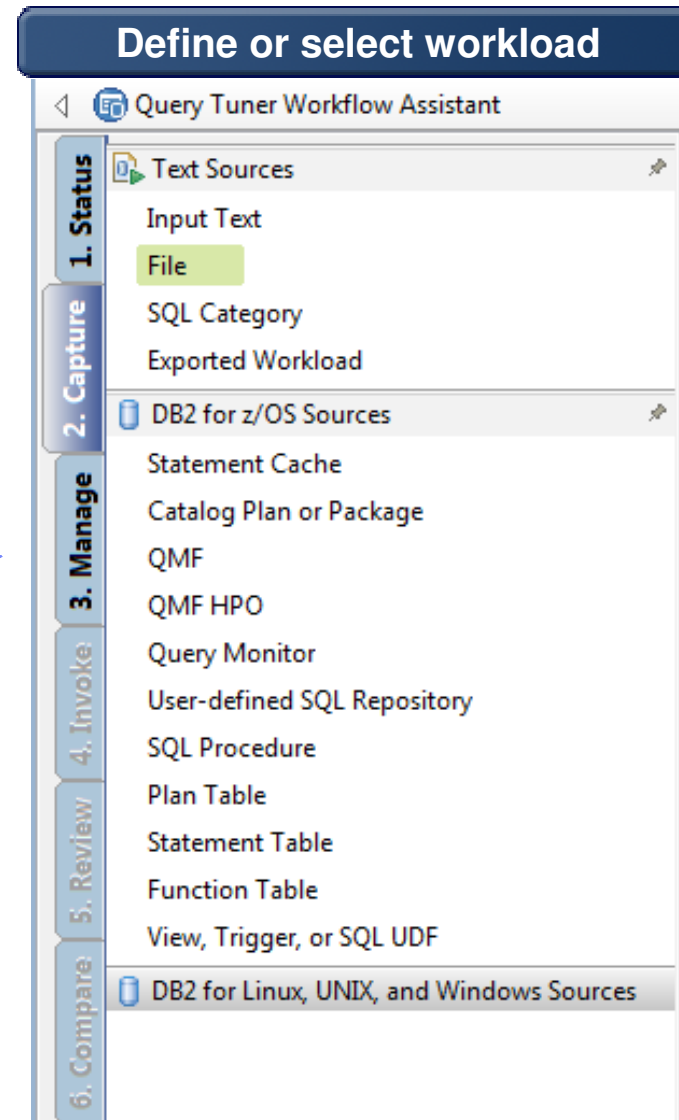
QMF and QMF HPO

DB2 Query Monitor

OMEGAMON XE for DB2

File, text, or exported
workload

User-defined SQL
Repository



Capture Workload

Task Launcher *QTProject1_Soonnee_806

Query Tuner Workflow Assistant

1. Status
2. Capture
 3. Manage
 4. Invoke
 5. Review
 6. Compare

Text Sources
 Input Text
 File
 SQL Category
 Exported Workload
 DB2 for z/OS Sources
 Statement Cache
 Catalog Plan or Package
 QMF
 QMF HPO
 SQL Procedure
 Plan Table
 Statement Table
 Function Table
 View, Trigger, or SQL UDF
 DB2 for Linux, UNIX, and Windows Sources

Capture SQL from Statement Cache

Create or select a filter for capturing SQL statements from the dynamic statement cache on the subsystem. You can then capture statements into them as a new workload.

Database connection: Soonnee_806 (DB2 for z/OS V9.1 (New-Function Mode))

Filter

Filter name: ACCUM_CPU_DESC Capture Schedule

☐ Disable cache trace after capturing

Captured Statements

The number of captured statements is 32. Right-click a statement and select Invoke Advisors and Tools. If workload tuning is enabled, you can capture statements.

STMT_ID	STAT_EXEC	STAT_CPU	STAT_ELAP	STAT_GPAG	STMT_TEXT
12	0	0	0	0	select QBLOCKNO ,TIMES ,ROWCOUNT ,ATOPEN ,CONTE
2	0	0	0	0	SELECT FOREIGNKEY FROM SYSIBM.SYSCOLUMNS WHERE NAME = 'TEXT
14	0	0	0	0	select P.QBLOCKNO as PQBLOCKNO ,P.PREDNO as PPREDNO ,P
24	0	0	0	0	DELETE FROM DSN_STATEMENT_CACHE_TABLE
1	0	0	0	0	SELECT FOREIGNKEY FROM SYSIBM.SYSCOLUMNS WHERE NAME = 'TEXT
31	0	0	0	0	SELECT NAME,TYPE,REMARKS,LABEL,EDPROC,VALPROC,AUDITING, DATA
13	0	0	0	0	select QBLOCKNO ,TABNO ,RANGE ,FIRSTPART ,LASTPAR
11	0	0	0	0	select STMT_TYPE , COST_CATEGORY , PROCMS , PROCSU
19	0	0	0	0	SELECT QUERY_STAGE, SEQNO, NODE_DATA FROM ADMF001.DSN_QUE
22	0	0	0	0	SELECT DB2OE.QT_LIC() AS LIC FROM SYSIBM.SYSDUMMY1
33	0	0	0	0	SELECT V.SRCTYPE AS SRCTYPE, V.QUALIFIER, V.TEXT, V.STMT_TEXT_LI
20	0	0	0	0	SELECT QUERY_STAGE, QBLOCKNO,PARENT,TIMES,ROWCOUNT,ATOPEN
29	0	0	0	0	SELECT NAME,TYPE,REMARKS,LABEL,EDPROC,VALPROC,AUDITING, DATA
9	0	0	0	0	SELECT CURRENT MEMBER FROM SYSIBM.SYSDUMMY1
8	0	0	0	0	select CURRENT SCHEMA AS SCHEMA,CURRENT SQLID AS SQLID,CURRE
4	0	0	0	0	SELECT 1 FROM SYSIBM.SYSDUMMY1 WHERE 0 = 1
25	0	0	0	0	SELECT CURRENT SQLID FROM SYSIBM.SYSDUMMY1

Visualize Queries and Costs to Speed Analysis

Formatted Query	Annotation	Additional Information
<pre> SELECT A.EMPNO , A.FIRSTNAME , A.LASTNAME , A.JOB , A.SALARY , A.BONUS , A.COMM , B.LOCATION , C.PROJNAME FROM DSN8910.DEPT AS B , DSN8910.EMP AS A , DSN8910.EPROJ AS C WHERE (A.EMPNO IN (SELECT DSN8910.DEPT.MGRNO FROM DSN8910.DEPT WHERE DSN8910.DEPT.MGRNO IS NOT NULL) AND A.WORKDEPT = B.DEPTNO AND B.DEPTNO = C.DEPTNO) ORDER BY A.EMPNO ASC , A.FIRSTNAME ASC , A.LASTNAME ASC </pre>	<pre> CARDF=14 QUALIFIED_ROWS= CARDF=42 QUALIFIED_ROWS= CARDF=(missing) QUALIFIED_ROWS= COLCARDF=42 MAX_FREQ=(r CARDF=14 QUALIFIED_ROWS= COLCARDF=9 MAX_FREQ=42 COLCARDF=8/14 MAX_FREQ= COLCARDF=14/(missing) MAX_FREQ= </pre>	<p>DSN8910.DEPT.MGRNO contain(s) skewed data</p> <p>DSN8910.EMP.WORKDEPT contain(s) skewed data</p>

- Accelerate analysis, reduce downtime
 - Spot human errors
 - Identify where filtering should occur

View Optimizer Transformations

Original	Transformed
Annotation to display: All	
Expand All Collapse All Customize Save Print	
Formatted Query	
<pre> SELECT A.EMPNO , A.FIRSTNAME , A.LASTNAME , A.JOB , A.SALARY , A.BONUS , A.COMM , B.LOCATION , C.PROJNAME FROM DSN8910.DEPT AS B , DSN8910.EMP AS A , DSN8910.EPROJ AS C WHERE (A.EMPNO IN (SELECT DSN8910.DEPT.MGRNO FROM DSN8910.DEPT WHERE DSN8910.DEPT.MGRNO IS NOT NULL) AND A.WORKDEPT = B.DEPTNO AND B.DEPTNO = C.DEPTNO) ORDER BY A.EMPNO ASC , A.FIRSTNAME ASC , A.LASTNAME ASC </pre>	

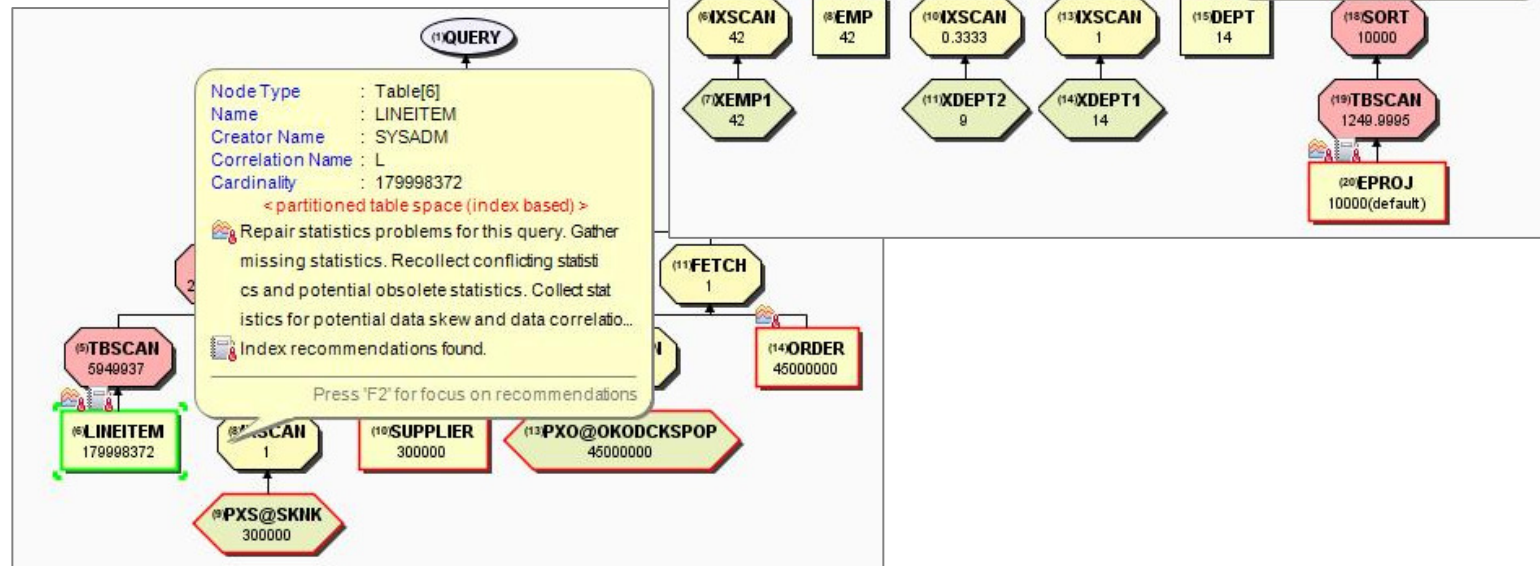


Original	Transformed
Annotation to display: All	
Expand All Collapse All Customize Save Print Clear Highlights	
Formatted Query	
<pre> SELECT A.EMPNO FROM DSN8910.DEPT AS B , DSN8910.EMP AS A , DSN8910.EPROJ AS C WHERE (A.WORKDEPT = B.DEPTNO AND A.WORKDEPT = C.DEPTNO AND B.DEPTNO = C.DEPTNO AND A.EMPNO = SYSADM."DSNWFQB(02)".MGRNO DB2 creates a virtual table, SYSADM.DSNWFQB(02), to process the following correlated subquery SELECT DSN8910.DEPT.MGRNO FROM DSN8910.DEPT WHERE (DSN8910.DEPT.MGRNO IS NOT NULL AND A.EMPNO = DSN8910.DEPT.MGRNO) End of the subquery for the SYSADM.DSNWFQB(02) virtual table.) ORDER BY A.EMPNO ASC , A.FIRSTNAME ASC , A.LASTNAME ASC </pre>	

- Accelerate analysis, reduce downtime
 - Spot transformations which will occur

Analyze Access Plans

- Visualize access path
 - See flow of query processing
 - See indexes and operations
 - See optimizer rationale



Execute Advisors

Run Workload Advisors

Run advisors to get recommendations for tuning the workload, or right-click a statement and run single-query advisors on it.

Workload name: WorkloadTunedWithStatsAdvisor
 Number of statements: 22
 Workload owner: B3OSC12
 Status:
 Description: contains a set of SQL statements with the updated catalog statistics. You can invoke workload advisors to get further

Run Default Advisors Select What To Run...

< Previous 1-8 rows out of 22 are displayed. Next >

STAT_EXEC	SOURCE	STAT_ELAP	AVG_STAT_ELAP	STAT_CPU	AVG_STAT_CPU	STAT_GPAG	AVG_STAT_GPAG
5	CACHE	116.755714	23.351143	2.278993	0.455799	1480	296.0
5	CACHE	498.326721	99.665344	0.59746	0.119492	1020	204.0
11	CACHE	248.512375	22.592033	5.055559	0.459596	3256	296.0
11	CACHE	208.310974	18.937361	0.084992	0.007727	506	46.0
13	CACHE	353.33551	27.179655	6.443507	0.495654	13860	1066.1538
16	CACHE	11.281075	0.705067	0.789257	0.049329	5120	320.0
16	CACHE	299.994202	18.749638	0.085831	0.005364	0	0.0
16	CACHE	0.504447	0.031528	0.170163	0.010635	0	0.0

Statistics

- Get recommendations on the best statistics to capture to influence access path selection

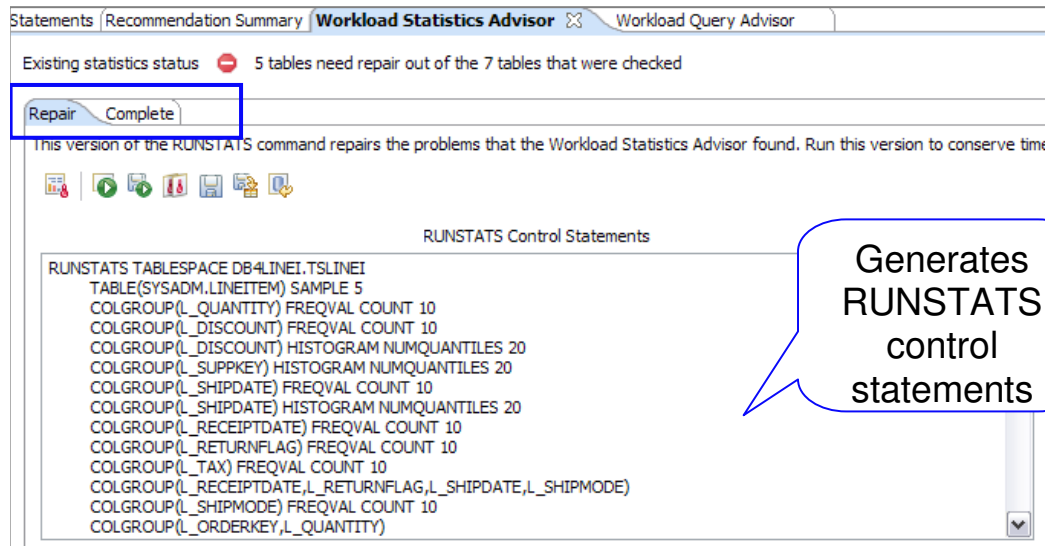
Query

- Get recommendations regarding how to rewrite the query for better efficiency

Index

- Get recommendations on additional indexes that can reduce database scans

Improve Statistics Quality and Collection



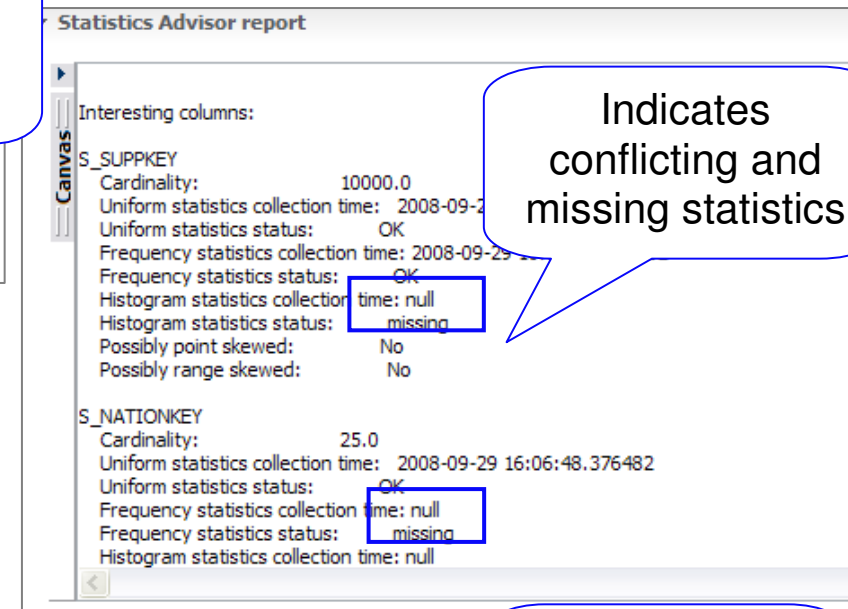
Generates
RUNSTATS
control
statements

Results

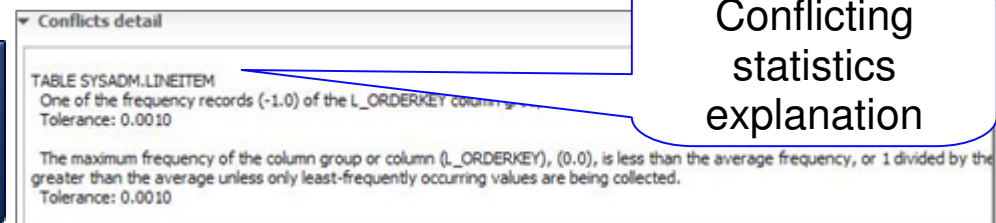
- Accurate estimated costs
- Better query performance
- Less CPU consumption
- Improved maintenance window throughput

“Half of access path PMRs could be resolved by statistics advisor before calling IBM support.” – IBM Support

- Provides advice on
 - Missing statistics
 - Conflicting statistics
 - Out-of-date statistics



Indicates
conflicting and
missing statistics



Conflicting
statistics
explanation

Improve Query Design

Statements Recommendation Summary Workload Query Advisor Workload Index Advisor Workload Query Advis

The following is a summary of the queries analyzed in the workload. Use this criteria to filter

Statements Sorted by	Number
Statements Analyzed Successfully	22
Statements with Warnings	4
Number of High Severity Warnings	0
Number of Medium Severity Warnings	0
Number of Low Severity Warnings	7
Statements with High Severity Warnings	0
Statements with Medium Severity Warnings	0
Statements with Low Severity Warnings	4

View analysis summary

Filter recommendations by severity

Query Advisor checks for

- Missing join predicate for referential constraint
- Predicates that can be rewritten as indexable
- Stage 2 predicates that can be rewritten as stage 1 predicates

Guard against errors and oversights:

Further constrain query, increase index utilization, and reduce data reads

Highlights relevant components of the query

SQL Text

```
SELECT A.EMPNO
, A.FIRSTNAME
, A.LASTNAME
, A.JOB
, A.SALARY
, A.BONUS
, A.COMM
, B.LOCATION
, C.PROJNAME
FROM DSN8910.DEPT AS B
, DSN8910.EMP AS A
, DSN8910.EPROJ AS C
WHERE ( A.EMPNO IN ( SELECT DSN8910.DEPT.MGRNO
FROM DSN8910.DEPT
WHERE DSN8910.DEPT.MGRNO IS NOT NULL
```

Recommendation and rationale

Selected Recommendation:

Recommendation Details

Provide a join predicate based on the referential constraint between tables DSN8910.EMP and DSN8910.DEPT.

There is a referential constraint between these two tables, but there is no join predicate that uses the foreign keys and their corresponding primary keys. Consider adding join predicates between columns EMPNO in table DSN8910.EMP and columns MGRNO in table DSN8910.DEPT to avoid fetching redundant data in the result set.

Explanation

If a referential constraint is defined between two tables, the queries that join the two tables generally have corresponding join predicates that map to the referential constraint.

Indexing Advice to Improve Query Efficiency

- Improve query efficiency
 - Indexing foreign keys in queries that do not have indexes defined
 - Identifying index filtering and screening
 - Support for index only access
 - Indexing to avoid sorts
- Simplify use
 - Consolidate indexes and provide a single recommendation
 - Enables what-if analysis
 - Provides DDL to create indexes
 - Run immediately or save
- Test before deployment
 - Utilize virtual index capabilities built into the DB2 engine

The screenshot displays the 'Workload Index Advisor' window. At the top, it shows the 'Recommendation Summary' tab. Key metrics are listed:

- Estimated performance improvement: 68.52 %
- Disk space required (DASD space): 673.08 MB
- Total IUD Cost: 0.0 MS

A callout bubble points to the 'Estimated performance improvement' value, stating 'Estimated performance improvement'.

Below the summary, the 'Recommendation' tab is active, showing a table of recommended indexes:

Feature Details	Action	Object N...	New Index Columns	Old Inde
[-] Create Indexes				
[-] CUSTOMER				
[-] Index	Create	CUSTOME...	C_MKTSEGMENT(ASC), C_...	
[-] Index	Create	CUSTOME...	C_MKTSEGMENT(ASC), C_...	
[-] ORDER				
[-] Index	Create	ORDER_V...	O_CUSTKEY(ASC), O_ORD...	
[-] Index	Create	ORDER_V...	O_ORDERKEY(ASC), O_O...	
[-] Index	Create	ORDER_V...	O_ORDERSTATUS(ASC), O...	
[-] LINEITEM				
[-] Index				
[-] Index				

At the bottom, a 'DDL for Selected Indexes' window is open, showing the SQL commands to create the recommended indexes:

```
CREATE INDEX DB2OE.LINEITEM_VIRT_IDX_1159535563777 ON SYSADM.LINEITEM (
  L_SUPPKEY ASC, L_EXTENDEDPRICE ASC, L_TAX ASC) FREEPAGE 0 PCTFREE 10;

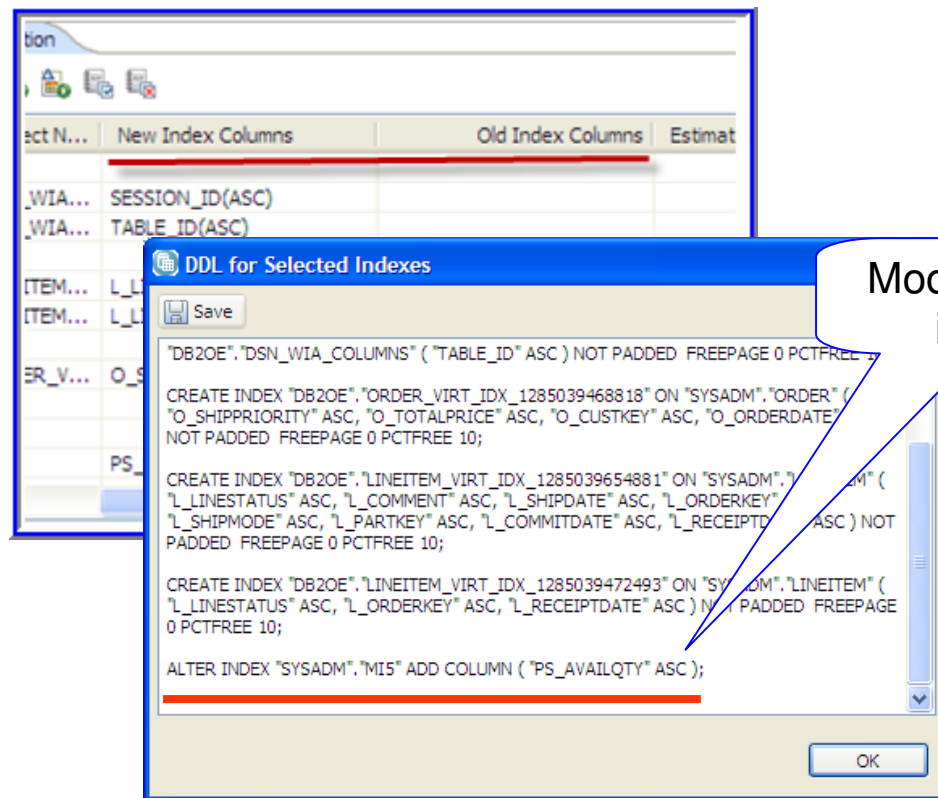
CREATE INDEX DB2OE.ORDER_VIRT_IDX_1159535563657 ON SYSADM.ORDER ( O_CUSTKEY
  ASC, O_ORDERDATE ASC, O_SHIPPRIORITY ASC, O_ORDERKEY ASC) FREEPAGE 0 PCTFREE
  10;

CREATE INDEX DB2OE.CUSTOMER_VIRT_IDX_1159535564238 ON SYSADM.CUSTOMER (
  C_MKTSEGMENT ASC, C_NATIONKEY ASC, C_ACCTBAL ASC) FREEPAGE 0 PCTFREE 10;

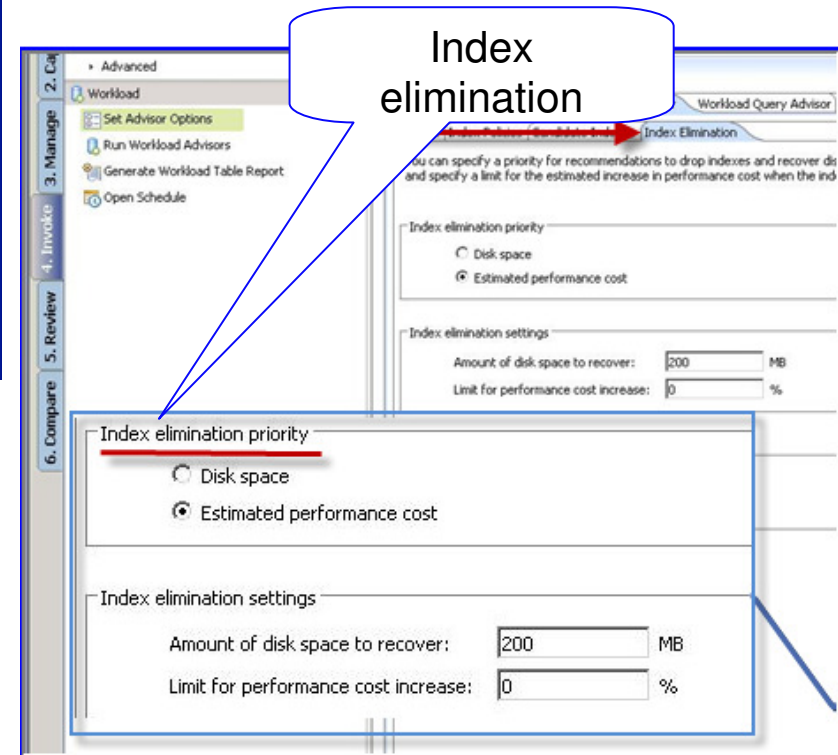
CREATE INDEX DB2OE.ORDER_VIRT_IDX_1159535554165 ON SYSADM.ORDER (
  O_ORDERKEY ASC, O_ORDERDATE ASC) FREEPAGE 0 PCTFREE 10;

CREATE INDEX DB2OE.LINEITEM_VIRT_IDX_1159535563807 ON SYSADM.LINEITEM (
  L_QUANTITY ASC, L_EXTENDEDPRICE ASC, L_TAX ASC) FREEPAGE 0 PCTFREE 10;

CREATE INDEX DB2OE.PARTSUPP_VIRT_IDX_1159535562295 ON SYSADM.PARTSUPP (
```



Modify existing
indexes

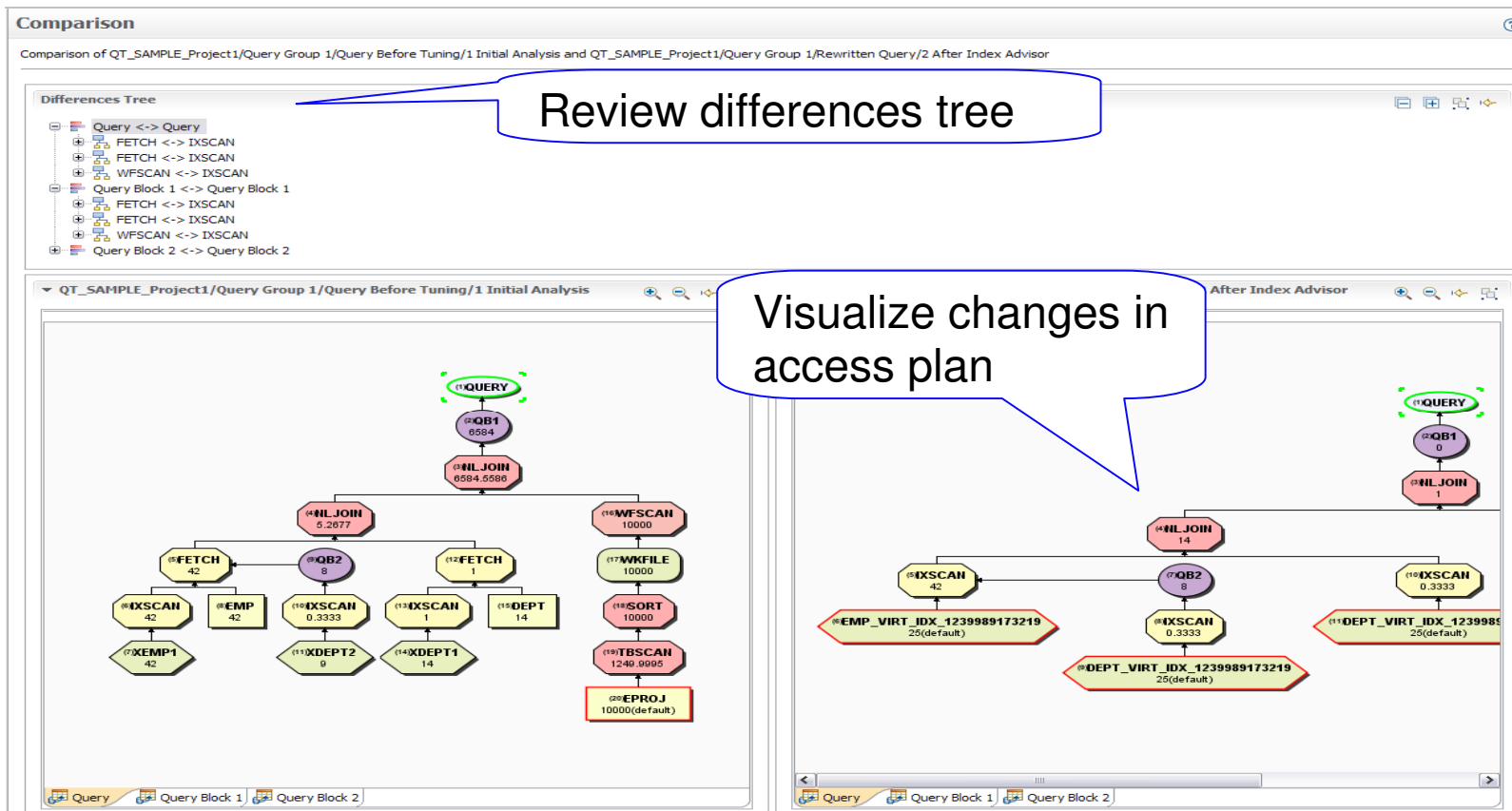


Index
elimination

Validate Improvement

Capture workloads repeatedly
to see tuning effects

Name	Summary Status	Owner	Execution Time
WorkloadWithTypicalStats	ANALYZING	B3OSC12	CPU time: 97.32 (second...
WorkloadTunedWithStatsAdvisor	ANALYZING	B3OSC12	CPU time: 53.19 (second...
WorkloadTunedWithIndexAdvisor	ANALYZING	B3OSC07	CPU time: 40.67 (second...
AbsoluteCPUTimeExceptionMonitor	ENABLED/STARTED	SYSADM	N/A
NormalMonitor	ENABLED/STARTED	SYSADM	N/A



Environment Capture Facilitates Collaboration

- Enable environment reproduction
- Speed up service process

Collect Data about the Workload Environment

This page generates a report that you can send to IBM support after you opened a problem management record. The report contains information about the environment in which your workload runs. After you generate the report, use this page to send the report to an FTP server for IBM support.

Report Options

☐ Include information about parallel processing ☒ Change values for storage group name, PRIQTY, and SECQTY

☒ Send the files to IBM Support

PMR or ETR Number (xxxxx,yyy,zzz): , , Version:

☐ Send the files to another subsystem or save them to your workstation only

Unique ID

Path:

FTP Server Settings

Server name: Port:

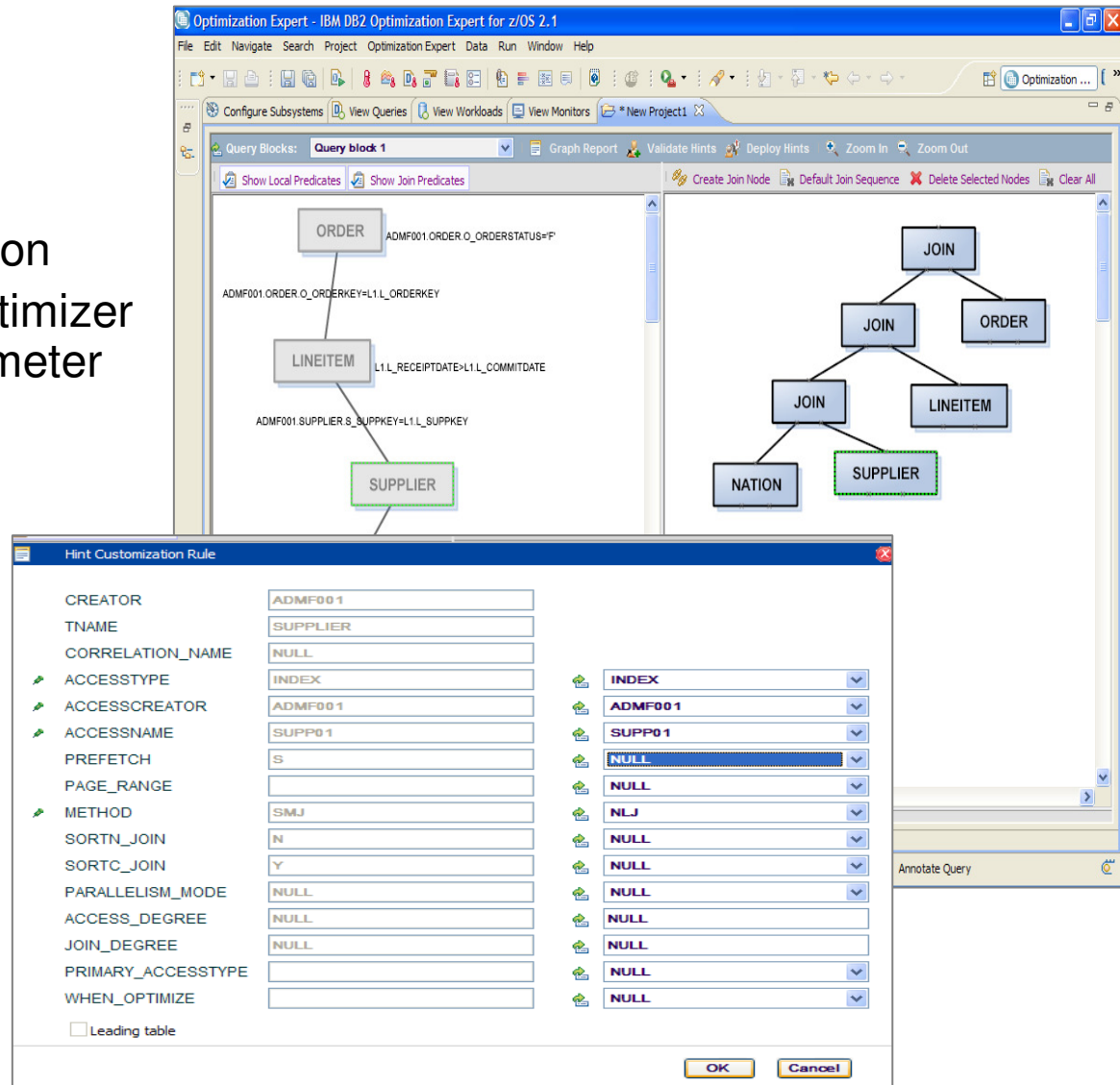
User: Password:

Directory:

Files to upload:

Visual Plan Hints for Experienced DBAs

- Why hints?
 - Version to version migration
 - Assumptions made by optimizer when SQL contains parameter markers or host variables
- Reduce errors
 - Hint editor
 - Hint validation
 - Hint deployment



Foster cross-role collaboration

- Challenge
 - Disconnect between application and data group
- Solution- verbose HTML file
 - Overview
 - Recommended actions
 - Input sql statement
 - Access Plan Summary
 - Transformed/optimized query
 - Predicates
 - Table access
 - Join operation
 - Table and index catalog information: table, column and index

IBM Query Tuner Report

This report contains a summary of the recommendations from the Query Tuner advisors and tools. Examine the recommendations and corresponding DDL scripts, appropriate actions to tune your query. You can also examine the formatted query and access plan summary, and cross reference the recommendations generated table, column, and index information to do further analysis and tuning. Navigate to the different sections using the action buttons and then return to the top of the report.

[View Query](#)
[Recommended Action](#)
[Access Plan](#)
[DB Catalog Info](#)

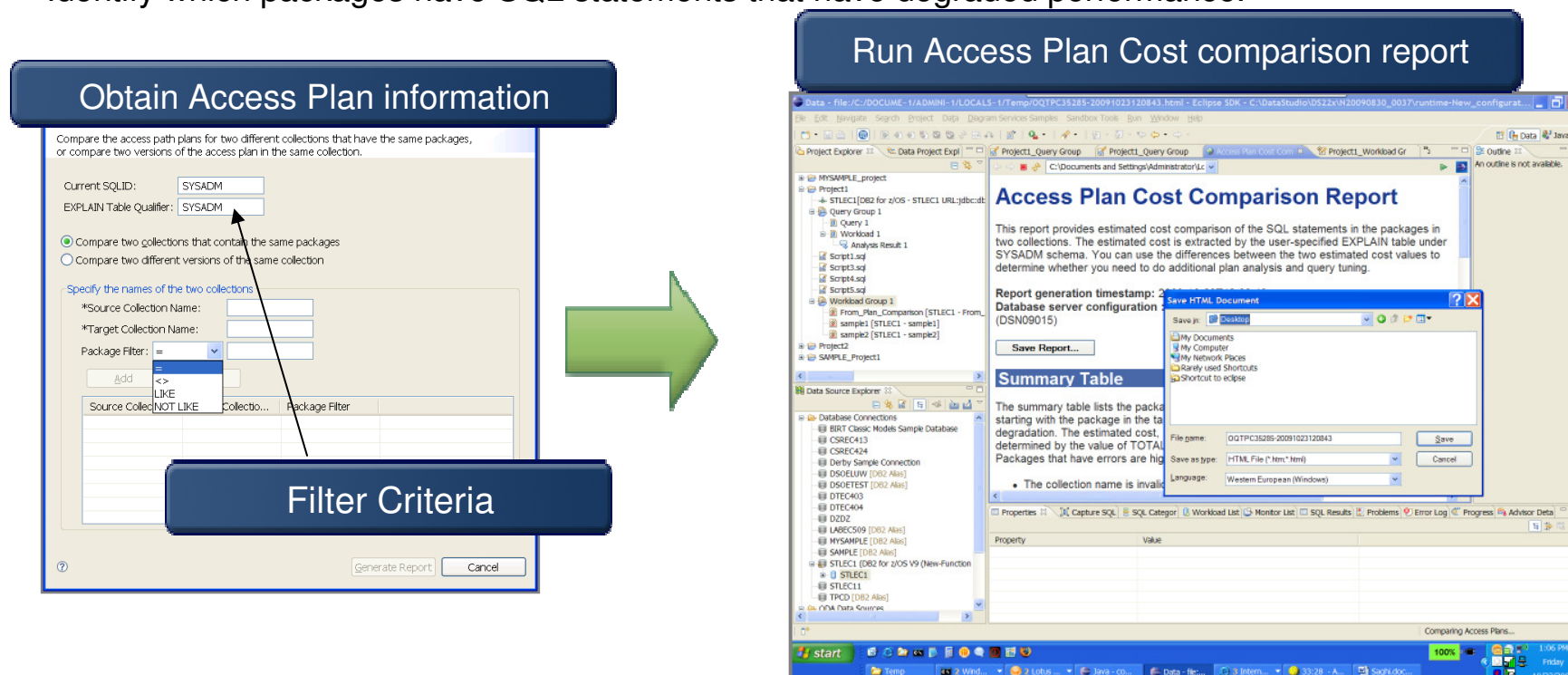
Overview

Recommendation generation timestamp: 2009-10-14T20:47:03
Database server configuration: jdbc:db2://dtec630.svl.ibm.com:446/STLEC1 (DSN09015)
Estimated plan cost: 351.787 units
Critical problems: 0 statistics recommendations, 0 index recommendations, 0 query recommendations, and 0 access path warnings.
Best practices: 1 statistics recommendations, 1 index recommendations, 1 query recommendations, and 1 access path warnings.

Advice Number	Advice Type	Tuning Recommendation Description
1	Statistics	Consolidate statistics. Use RUNSTATS to recollect all the relevant statistics for this query for an accurate evaluation. Important: if statistics are missing, Query tuner estimates subsequent recommendations based on database default statistics. Click here to review the recommended RUNSTATS script.
2	Index	Index recommendation is found for the following tables: SYSIBM.SYSCOLUMNS. The total estimated disk space required is 0.066 MB and the estimated performance improvement is 1.704%. This index recommendation will help improve query performance for join processing. Click here to review the recommended CREATE INDEX DDL script.
3	Query	Avoid an asterisk (*) or a long column list in the SELECT clause. Consider replacing the asterisk (*) or the long column list for table SYSIBM.SYSTABLES in the SELECT list with just the names of the required columns. Click here to see the affected query text.
4	Access path	Avoid table space scan (QBLOCKNO = 1, PLANNO = 1) on table SYSIBM.SYSTABLES. The table is accessed by a table space scan (table scan). When a large number of records are returned, DB2 might be using an inefficient access path. Consider running the statistics advisor or running the index advisor to determine whether creating an index might improve the access path. Click here to see the access plan operator.

Reduce Risk of Performance Regression After Migration

- Compare access path cost of SQL statements in a package via HTML
 - Version to version migrations of DB2
 - Cases where a mass-rebind is needed
- Optimize beyond the prior level of service.
 - Determine whether the later version of the collection has degraded performance.
 - Determine whether any packages have errors.
 - Identify which packages have SQL statements that have degraded performance.



Comparison result – package and SQL level

QTProject1

Query Tuner Workflow Assistant

1. Status
Single Query Comparison
Compare Access Plan Graphs

2. Capture
Workload Plan Comparison
Compare Package Access Plan
Compare Workload Access Plan

3. Manage
STLEC1_Workload_0 Comparison Result
Comparison History
2010-08-23 02:51:12.543404
Package Summary

4. Invoke

5. Review

6. Compare

View Comparison Result -Package Summary

Database connection: csrec791 (DB2 for z/OS V9.1 (New-Function Mode))

Workload name: COMPARISON_WL1

Number of filters: 2

Start Time: 2010-08-23 02:51:12.543404

Stop Time: 2010-08-23 04:51:18.524538

Package Summary

Package (Collection)	Version	Estimated Cost	# of SQL regressed/total	# of SQL improved/total	# of SQL added
AOC5OAPC			12/24	2/24	2
DB2OSC	20100803	22255252.2676			
DB2OSC1	20100803	23562565.1872			
AOC5IAOC			10/17	0/17	0
DB2OSC	20100803	13452565.9872			
DB2OSC1	20100803	13457878.6735			
AOC5TUID			12/24	2/24	2
DB2OSC	20100803	22255252.2676			
DB2OSC1	20100803	23562565.1872			
AOC5TVIP			10/17	0/17	0
DB2OSC	20100803	13452565.9872			
DB2OSC1	20100803	13457878.6735			

Package and SQL Level

DB2 for z/OS V9.1 (New-Function Mode))

2010-08-23 02:51:12.543404

Package Summary

Package and SQL

Collection ID	Version	Estimated Cost	# of SQL regressed/total	# of SQL improved/total	# of SQL added
DB2OSC	20100803	22255252.2676	12/24	2/24	2
DB2OSC1	20100803	23562565.1872	12/24	2/24	2

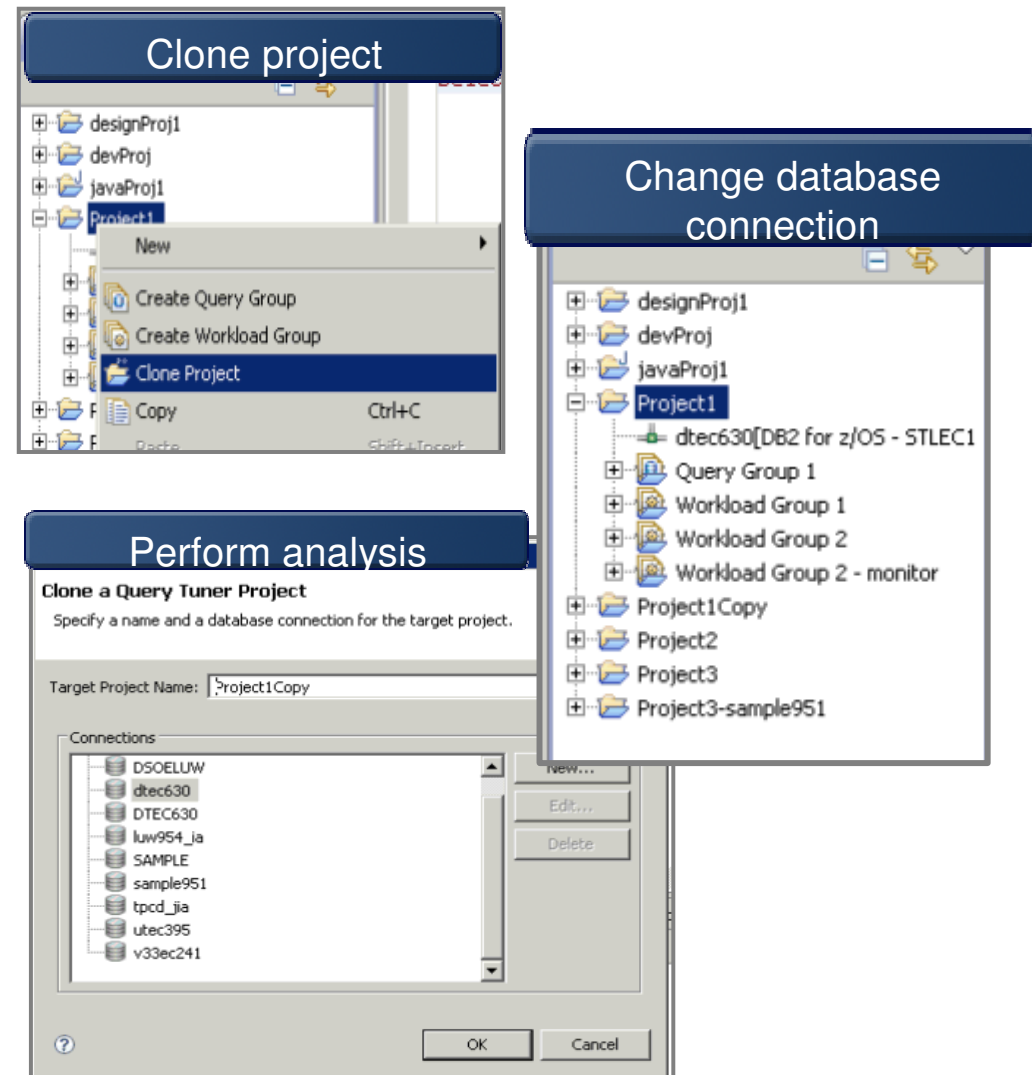
Statements

Package: AOC5OAPC

STMT_ID	Regressed	Access Path Changed	Regressed Ratio	Old Cost	New Cost	STMT Chan	SQL Text
1011	Y	Y	-19.18	14563.12	17127.38	N	SELECT *
1012	Y	N	-10.49	7563.12	8356.23	N	SELECT *
1014	N	N	N/A	NA	14563.12	A	update id

Validation Against a Production System

- **Accelerate development**
 - Develop, test, and run queries, procedures, and functions.
- **Enhance productivity**
 - Analyze impact of changes
 - Enhance problem isolation
- **Be proactive**
 - Leverage best practices for new development
 - Test before deployment
 - Adapt ahead of impact
 - Minimize production slowdowns



Minimize Production Slowdowns, Tune in Development

Visualize query hot spots during development

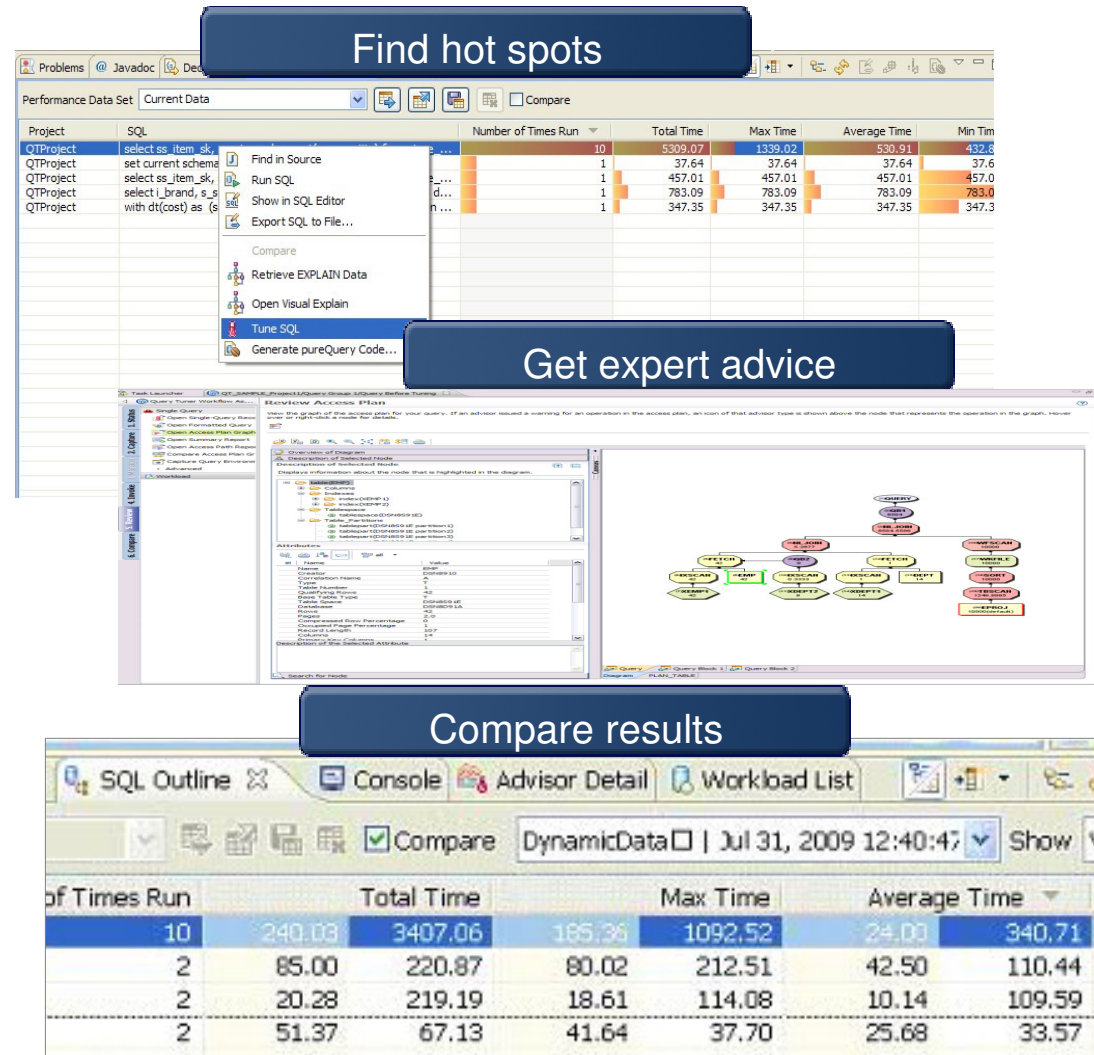
- Most frequently executed
- Longest running

Get expert advice

- View formatted queries with statistical annotations
- Improve query structure
- Improve statistics quality and collection efficiency
- Get index recommendations

Validate improvement

- Replace SQL
- Run application
- Compare performance results



Agenda: DB2 Anwendungsoptimierung mit Optim



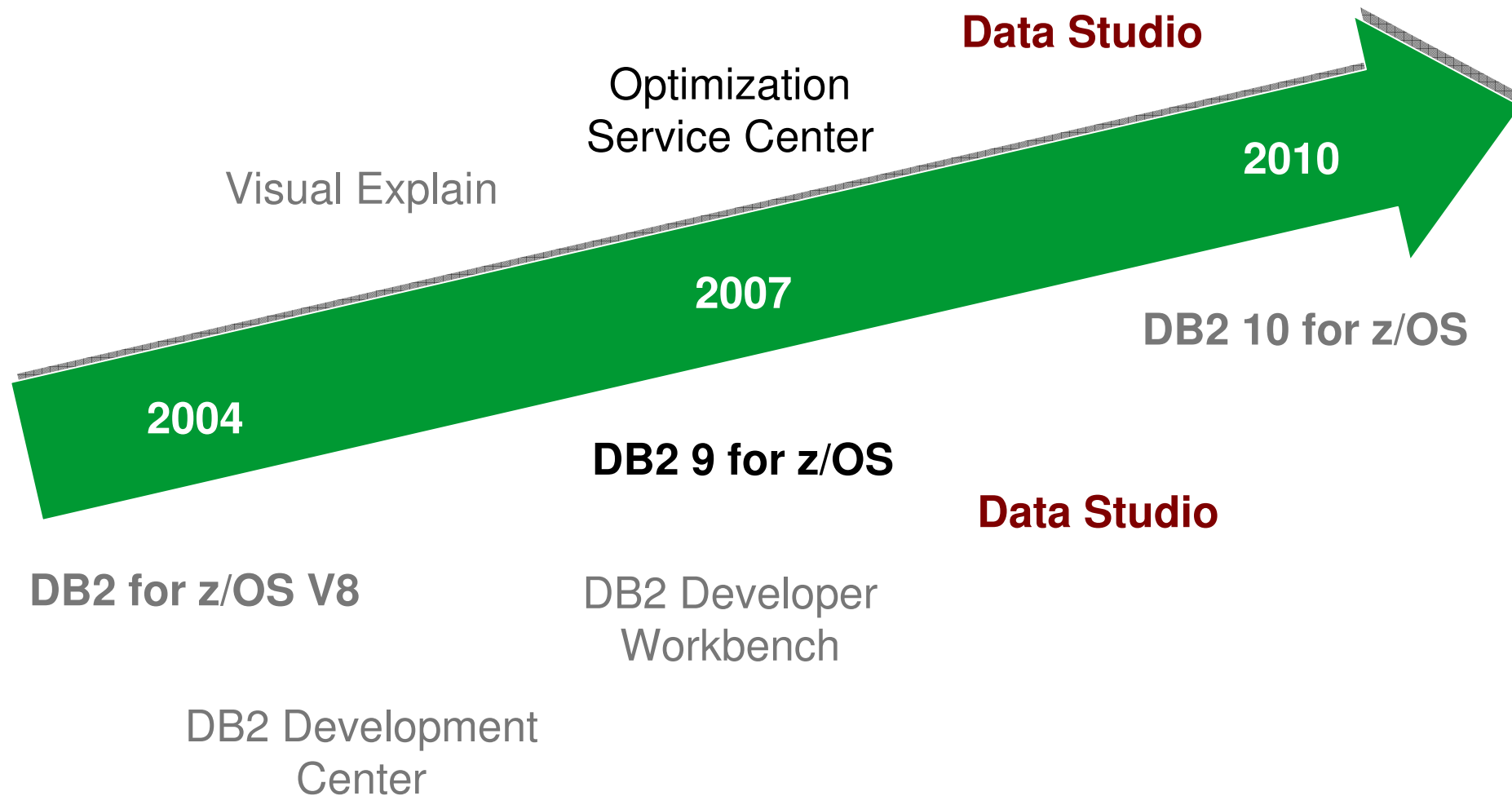
Optim Query Workload Tuner

DB2 z/OS Themen

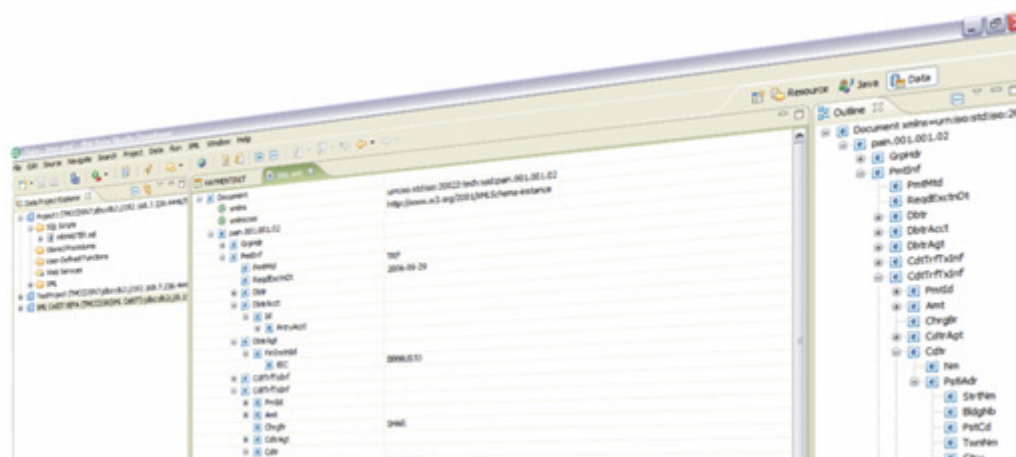
Query Tuning / Optimizer / DataStudio
Acceleratoren

Java Zugriffe: Optim pureQuery Runtime

Kostenfreie Tools



Data Studio Varianten ab Version 3.1.1



Data Studio V3.1.1 Administration Client

Zielgruppe: Datenbankadministratoren

**Reduzierter Funktionsumfang mit Fokus auf
DB2-/Administrations-Funktionalitäten**

Größe: ~ 330 MB

**Direktinstallation ohne Installation Manager
(bisher Stand-Alone Package)**

Data Studio V3.1.1 Full Client

Zielgruppe: Anwendungsentwickler

**Voller Funktionsumfang inkl. vollständigem
Eclipse und Entwicklungs-Funktionalitäten**

Größe: ~ 1.200 MB

**erfordert IBM Installation Manager
(bisher IDE Package)**

<http://www.ibm.com/developerworks/downloads/im/data/>

Contents of Eclipse-based Query Tuning offerings

	Data Studio	InfoSphere Optim Query Tuner for z/OS	Optim Query Workload Tuner for z/OS
Queries from all sources			
Reports			
Query Formatter			
Access Plan Graph			
Query Statistics Advisor			
Query Annotation			
Visual Plan Hint			
Query Index Advisor			
Query Advisor			
Access Path Advisor			
Workload Statistics Advisor			
Workload Index Advisor			
Workload Query Advisor			

Motivation für Query Tuning

- DB2 ist aufgrund der langjährigen Historie in diversen Einsatzszenarien vertreten
- Unterschiedliche Anforderungen an DB2
 - OLTP (CICS/IMS)
 - Batch Verarbeitung
 - Web (Java, ODBC,...)
 - DataWarehousing
 - Business Intelligence
 - Statistik
 - Geografische Datenverarbeitung



Kostendruck als ein Schwerpunkt

- IT Betrieb als Kostenfaktor
- Projekte zur Kostensenkung
- Mehr Workload mit den selben Strukturen/Kosten
- Kapazitätsupgrades vermeiden
- CPU als wichtiger Faktor auf Mainframe



→ QueryPerformance hat Potential zur CPU Reduktion

Häufig auftretende Punkte

- Cobol / PL1 in der Regel besser im Griff als .NET, Java etc.
- Aufgrund anderer Prioritäten wird immer weniger Focus auf QueryTuning in der AE gelegt
- Verständnis für DB2 Optimizer sinkt
- Sicherung von Zugriffspfaden wird als schwierig gesehen
- Kippen von Zugriffspfaden
- Tooling nur für Teilbereiche produktiv nutzbar
- Hoher manueller Aufwand, der Ressourcen bindet
- Softwareprobleme mit DB2 Optimizer
- Wo anfangen? Zuviel Workload
-



Lösungskomponenten

- DB2 z/OS
 - Plantable, VisualExplain, StatsAdvisor, DataStudio
 - OptHints
 - Bind Parameter (Degree, Reopt, ...)
 - Materialized Query Tables
- V9
 - Plan Stability
 - Reopt Auto
 - Optimizer Erweiterungen (Global Opt, Index ANDing, Histogram)
 - Virtual Index
 - Index on Expression
 - XML
 - Spatial und Omnifind
- V10
 - Optimizer Erweiterungen (Safe Query Optimization, Inlist Optimization, SQL Pagination, Degree Any)
 - Index Include Column
 - Plan Stability
 - DSC Literal Replacement
 - CurrentExplainMode
 - Package last used
 - Hash Access



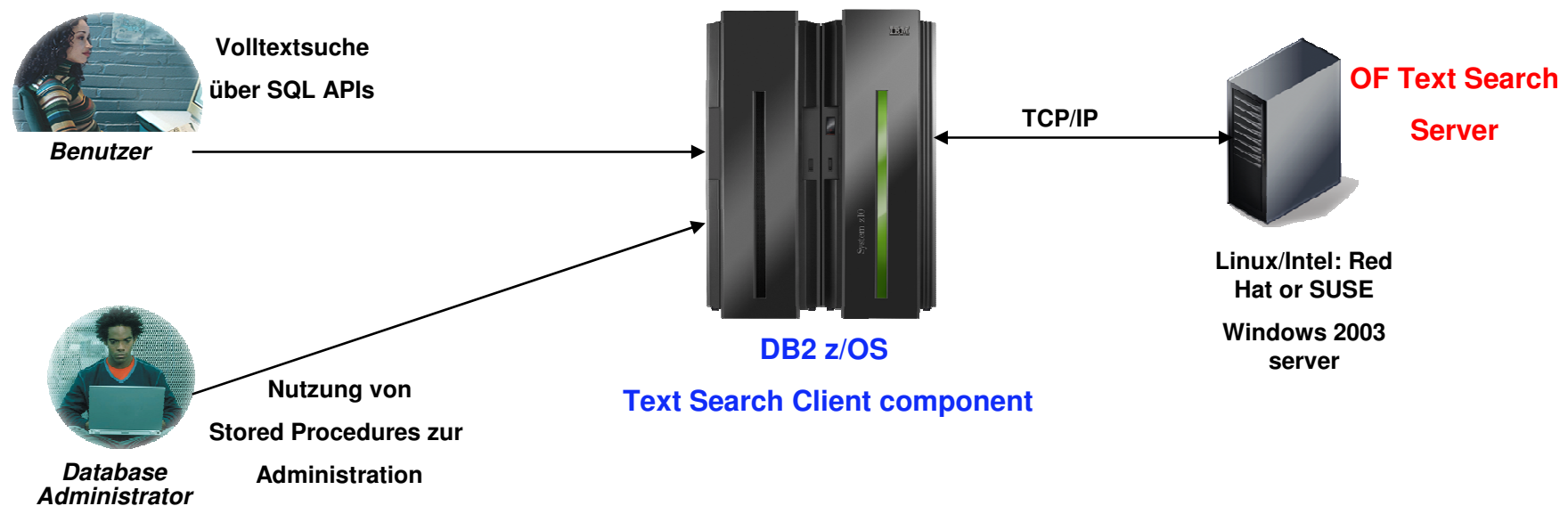
Mögliche Werkzeuge

- **Monitore**
 - Omegamon XE for DB2 Performance Expert
 - Auffinden von Performanceproblemen
 - DB2 Query Monitor
 - Auffinden von SQL Problemen
- **Tuning Tools**
 - SQL Performance Analyzer
 - Analyse der SQL Statements (v.a. TSO, stat.SQL)
 - Optim Query Workload Tuner
 - Analyse von SQL Statements (v.a. GUI, stat+dyn SQL)
 - Data Studio
 - freies Werkzeug zur einfachen SQL Analyse
 - Bind Manager / Path Checker
 - Binds verhindern / Zugriffspfadverschlechterung verhindern
- **Infrastruktur**
 - Optim pureQuery Runtime
 - Optimierter Zugriff für Java, CLI
 - IBM DB2 Analytics Accelerator
 - Appliance für komplexe Queries
 - Omnifind
 - Volltextsuche über SQL



Omnifind Text Search Server für DB2 z/OS

- **OF Text Search Server for DB2 z/OS**
 - Omnifind stellt einen Text Index Server zur Indizierung von DB2 Tabellen zur Verfügung
- **IBM Text Search Client component on DB2 z/OS**
 - Ermöglicht die Volltextsuche und Administration des OF Text Search Server über SQL APIs
 - Kostenfreie Bereitstellung über DB2 Accessories Suite for z/OS

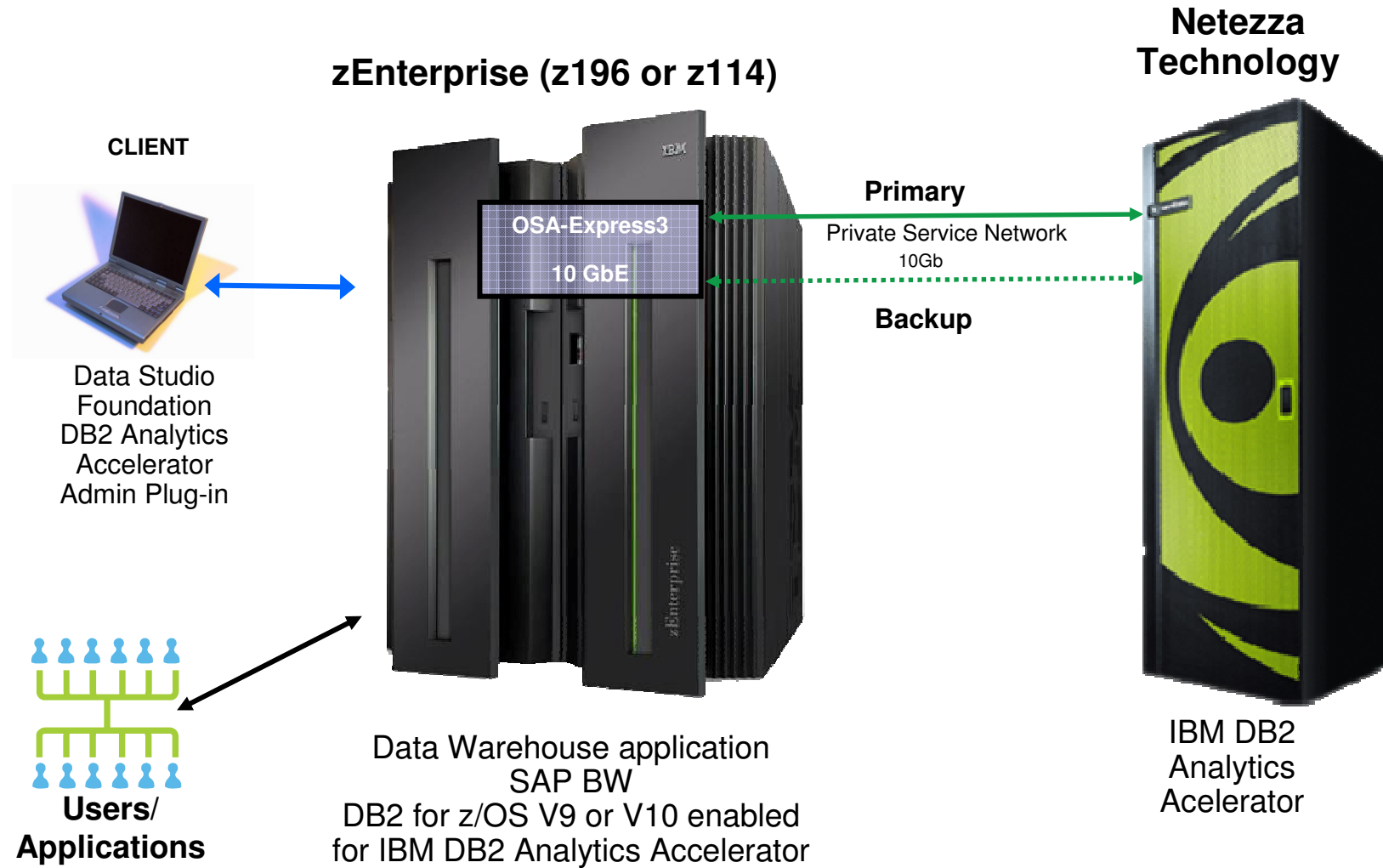


Sample queries

INT1	CLB
1	This is clob data
2	Herbs have been used
3	around the world for centuries.
4	For over 70 years
5	, Ricola has been making
6	pleasant tasting natural herbal
7	products for natural relief
8	of the discomfort due to sore throat
9	and minor irritation due to
10	coughs, hoarseness, dry throat.
11	The IBM OmniFind Text Search Server for DB2 is...

- `SELECT int1 FROM text_search WHERE CONTAINS(CLB, 'is, IBM') = 1`
→ '11'
- `SELECT int1 FROM text_search WHERE CONTAINS(CLB, 'has been making') = 1`
→ '5'
- `SELECT int1 FROM text_search WHERE CONTAINS(CLB, 'was making') = 1`
→ '5'
- `SELECT int1 FROM text_search WHERE CONTAINS(CLB, 'organic') = 1`
→ '6, 7'
- `SELECT int1 FROM text_search WHERE CONTAINS(CLB, 'is OR IBM') = 1`
order by score(CLB, 'is OR IBM')
→ '11, 1'

IBM DB2 Analytics Accelerator V2



Agenda: DB2 Anwendungsoptimierung mit Optim



Optim Query Workload Tuner

DB2 z/OS Themen

Query Tuning / Optimizer / DataStudio

Acceleratoren

Java Zugriffe: Optim pureQuery Runtime

Herausforderungen im Umfeld Java und DB2

Kapazitätsplanung

Ressourcenverbrauch

Zugriffspfadstabilität

Übergabeverfahren

Statisches SQL

Database Object Tracking

Leistungsfähigkeit

Authorisierung

Ganzheitliche Problemfeststellung



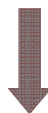
Statisches SQL hat oftmals nur einen geringen Stellenwert bei der AE



Anwendungsentwickler

Bevorzugt JDBC aufgrund der **einfachen Handhabung**, der **vorhandenen Tools** und den **eigenen Kenntnissen**.

JDBC



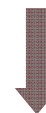
Dynamic SQL



Datenbankadministrator

Bevorzugt statisches SQL aufgrund der **überlegenen Leistungsfähigkeit**, **Sicherheit** und des **besseren Monitoring**.

SQLJ

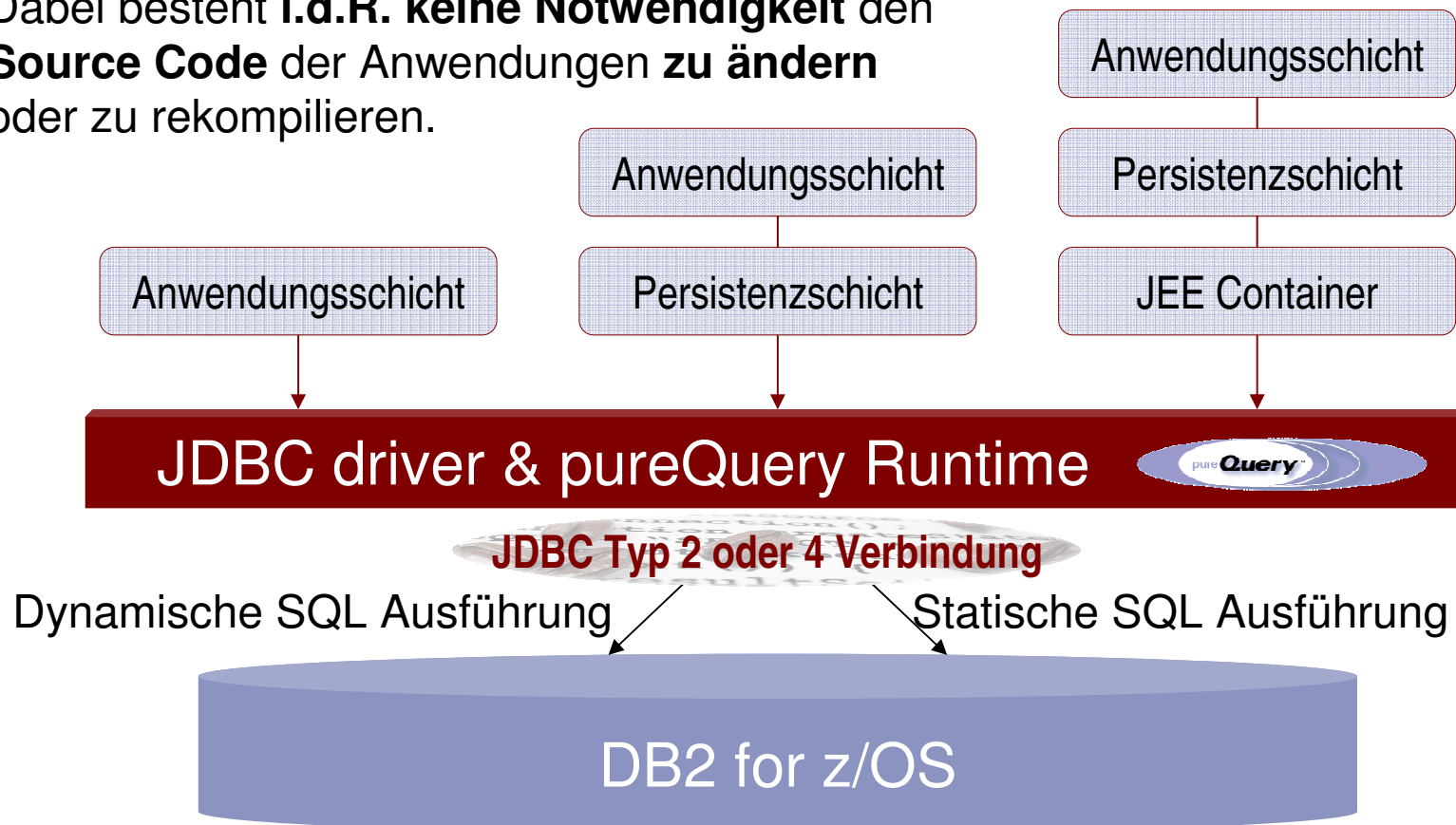


Static SQL



Diese Funktionalität steht für jegliche Art von JDBC Anwendung bereit

- Die pureQuery Runtime kann sowohl für **selbstgeschriebene** als auch für **Drittanbieter JDBC Anwendungen** verwendet werden, **unabhängig von** eventuellen **Persistenzframeworks** (z.B. Hibernate).
- Dabei besteht **i.d.R. keine Notwendigkeit** den **Source Code** der Anwendungen **zu ändern** oder zu rekompilieren.



Client Optimization

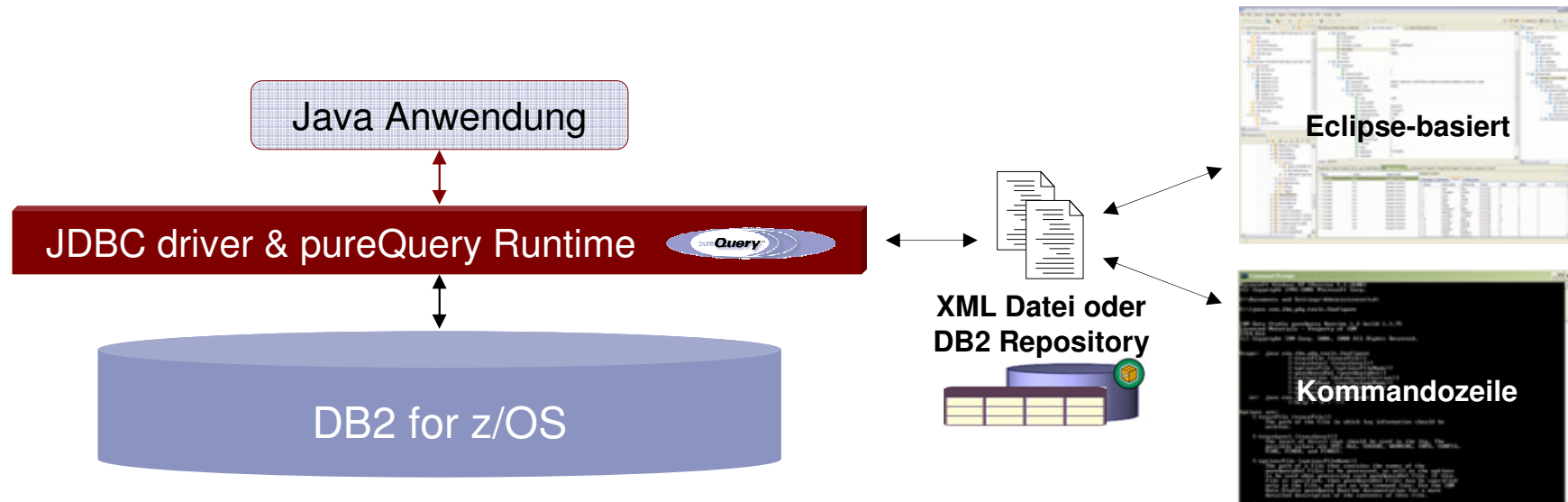
pureQuery Client Optimization bezeichnet den Prozess, herkömmliche JDBC Anwendungen automatisiert auf statisches SQL umzustellen ohne Eingriff in die Anwendung vorzunehmen.

Ein wesentlicher Bestandteil des Client Optimization ist der Capture Prozess. In diesem Prozess werden SQLs gesammelt und im DB2 gebunden.

Der JDBC Client Optimization Prozess als Grundlage für statisches SQL

Capture → Configure → Bind → Execute

- In der **Capture** Phase werden **sämtliche, erfolgreich ausgeführten SQL Statements** sowie **Metadaten** in einer XML Datei oder im DB2 erfasst.
- Der **Configuration** Schritt ermöglicht die **Anreicherung** und **Modifizierung** der erfassten Daten sowie eine Strukturierung der späteren DB2 Packages.
- Beim **Bind** Schritt werden die notwendigen **Packages** im DB2 erzeugt.
- Zur **Execution** Zeit kann **mittels Driver Parametern** der **Ausführungsmodus** eingestellt und somit die Statements **auf statisches SQL** umgestellt werden.

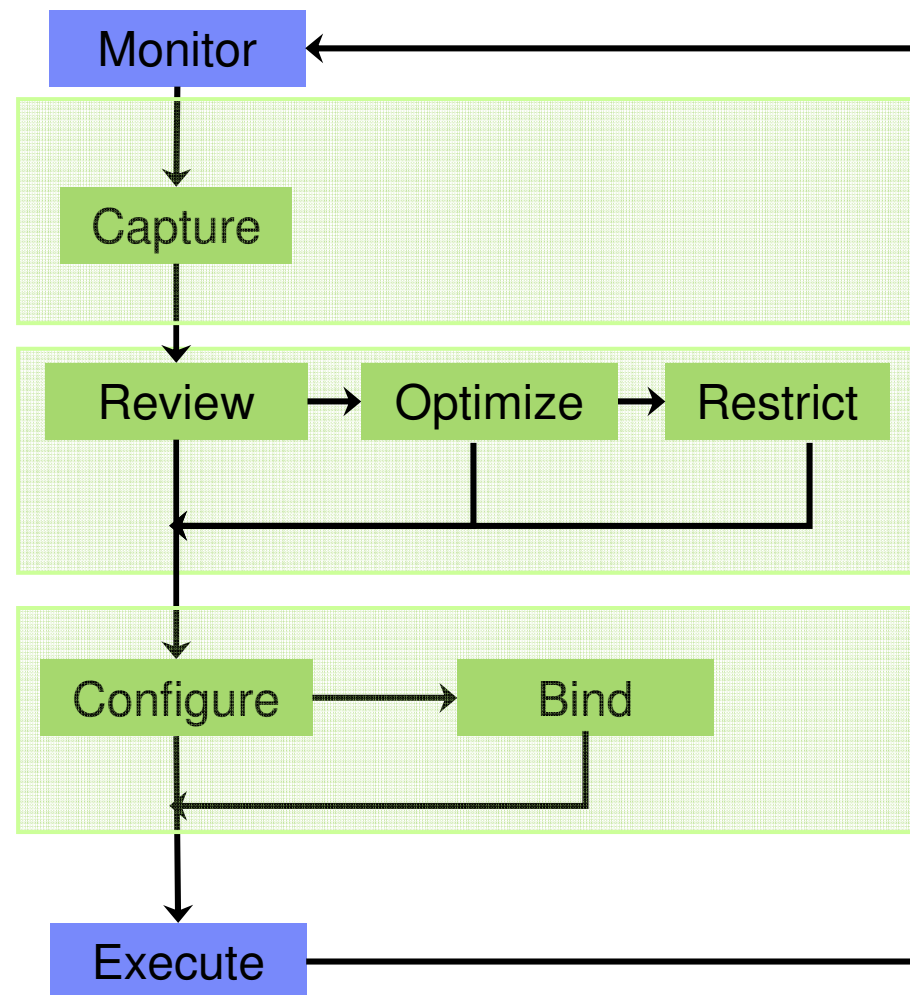


Zusätzliche Vorteile durch die Capture Phase

- Mehr Einblick
 - Welche SQLs werden überhaupt abgesetzt?



- Optimierung
 - Kontrolle der SQL Statements
 - Anzeige von HotSpots
 - Impact-Analysis für Schema Änderungen
 - SQL Tracing im Sourcecode
 - Optimierung und Änderung von SQLs
 - Erzeugung einer SQL Whitelist um z.b. SQLInjection zu vermeiden
- Deployment
 - Nutzung von Literal Replacement
 - Einrichtung verschiedener Profile
 - Optionale statische Ausführung



Mehr Einblick in die Anwendung

The screenshot displays the IBM Optim Developer interface with several callouts highlighting key features:

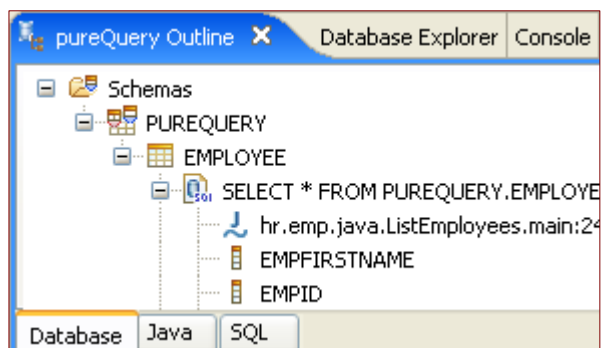
- Verbindung zum Sourcecode**: Points to the Java code editor showing the `ReportApplication.java` file.
- Darstellung des erfassten SQL**: Points to the **SQL Outline** tab, which lists SQL queries and their execution metrics.
- Vergleich von Ausführungsmetriken**: Points to the table of execution metrics in the **SQL Outline** tab.
- Ersetzen von SQL**: Points to the **Data Source Explorer** tab, which shows the database schema.
- Verbindung zur DB2 DataSource**: Points to the **Data Source Explorer** tab, which shows the database schema.
- Execute, tune, share, trace, export SQL**: Points to the context menu for the **SQL Outline** tab, which includes options like **Run SQL**, **Tune SQL**, and **Export SQL to File...**.

The **SQL Outline** table shows the following data:

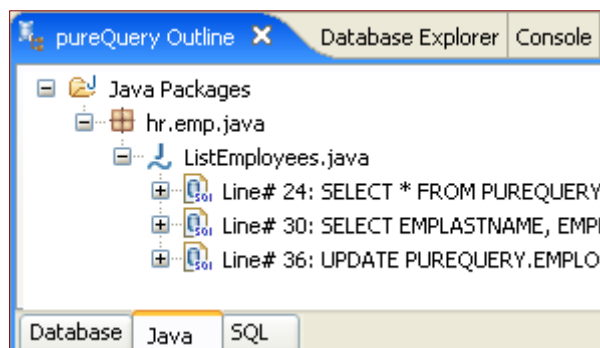
Schemas	Number of Times Run	Total Time	Max
GOSALESC			
CUST			
SELECT CUST_CODE, CUST_FRST_NAME, CUST_LAST_NAME, CU	1	90.92	90.92
SELECT CUST_CODE, CUST_FRST_NAME, CUST_LAST_NAME, CU	2	297.06	273.61
SELECT CUST_CODE, CUST_CITY FROM GOSALESC	3	20.09	7.99
SELECT CUST_CODE, CUST_FRST_NAME, CUST_LAST_NAME, CU	1	2.70	2.70
SELECT count(CUST_CODE) FROM GOSALESC	14	61.04	40.82
SELECT count(CUST_CODE) FROM GOSALESC	14	61.04	40.82
ReportGenerator			
CUST_CODE			
CUST_ORD			
SELECT CUST_CODE, STDDEV	4	7.11	2.18
SELECT ORD_NBR AS ORDER	5853	3335.63	46.98
SELECT CUST_CODE, SUM(ORD	3	5.80	2.38
SELECT CUST_CODE, AVG(ORD	4	5.17	1.40
SELECT CUST_CODE, CORRELA	3	3.24	1.20

Zusammenhänge zwischen Anwendung und DB2 erkennen

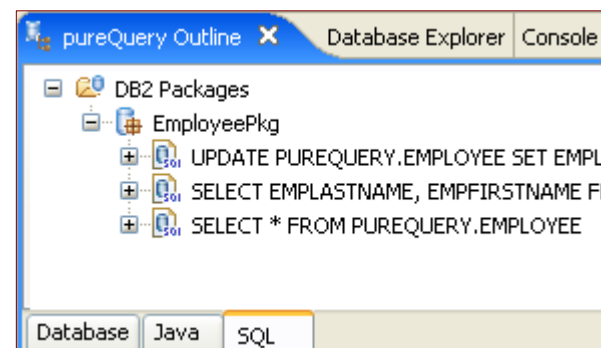
Abschätzen der Auswirkungen von Java Anwendungen auf DB2



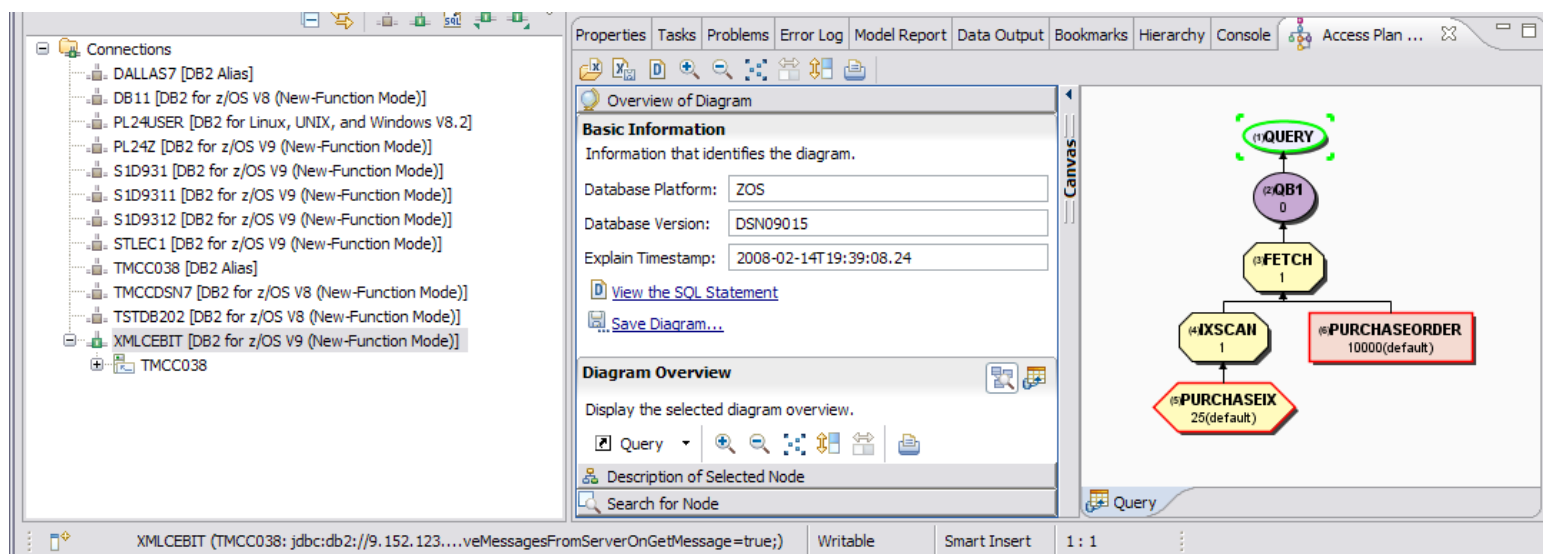
Database objects



Java source code

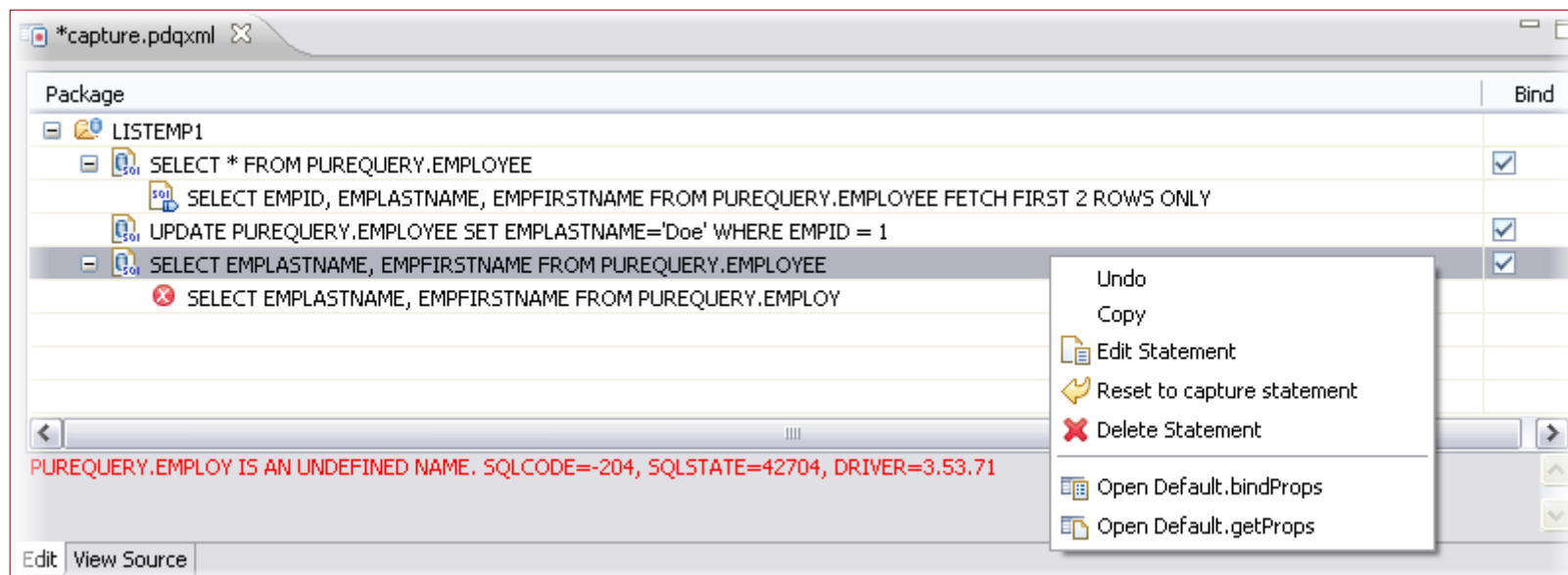


DB2 packages



Zudem erlaubt pureQuery eine bessere Kontrolle der SQL Statements

- Statements die **Performance Probleme** verursachen oder **unerwünschte Ergebnisse** zurückliefern, können mittels pureQuery und Optim Development Studio geändert werden.
- In der Folge setzt die pureQuery Runtime dann die **modifizierten** oder **gänzlich alternativen Statements** an die Datenbank ab.
- Literal Replacement:** Wenn die Anwendung ohne Parameter Marker arbeitet, kann pureQuery die genutzten Literale durch **Parameter Marker / Host Variablen** ersetzen



Besonderheiten bei der Verwendung des einfachen Statement Objekts

- Der **Dynamic Statement Cache** von DB2 hält die **Zugriffspfade** von dynamischen SQL Statements vor und zwar **exakt in der ausgeführten Form**.
- Beim vorherigen Beispiel wird der Zugriffsfad für

```
SELECT LASTNAME FROM DSN8910.EMP WHERE EMPNO < '000110'
```

im Dynamic Statement Cache abgelegt.

- Verwendet man nun allerdings ein Statement der Form

```
SELECT LASTNAME FROM DSN8910.EMP WHERE EMPNO < '000210'
```

muss der gesamte Zugriffspfad neu berechnet werden.

- Eventuell vorhandene **Statement Caches** von **Anwendungsservern greifen i.d.R. nicht bei Verwendung** des einfachen **Statement Objekts**.
- Ein simples Statement kann bei der Ausführung alles oder nichts sein...

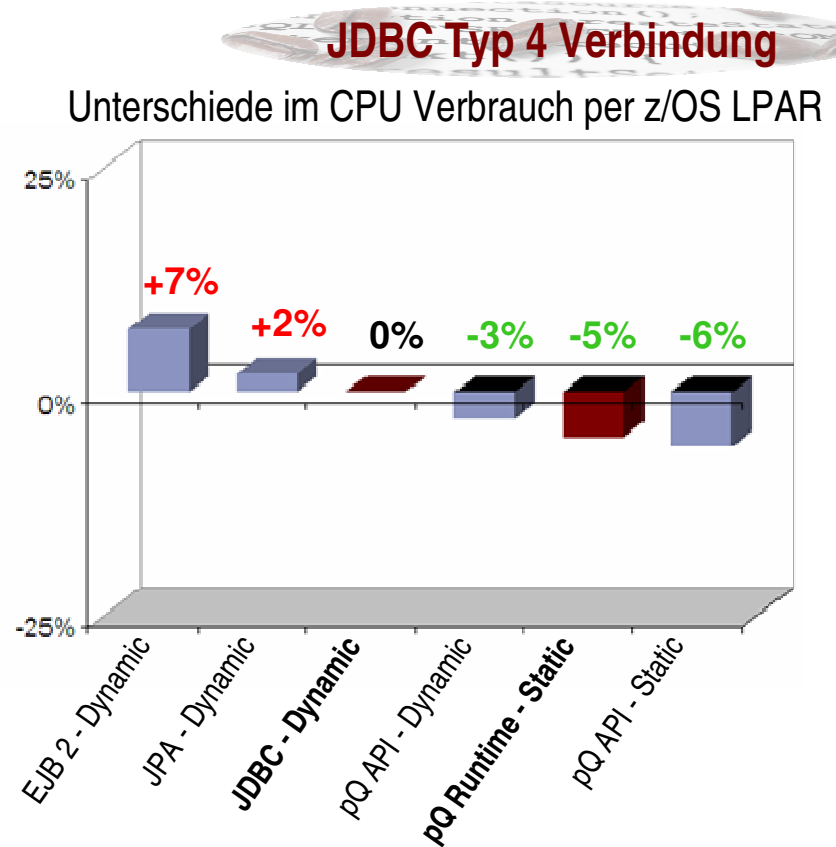
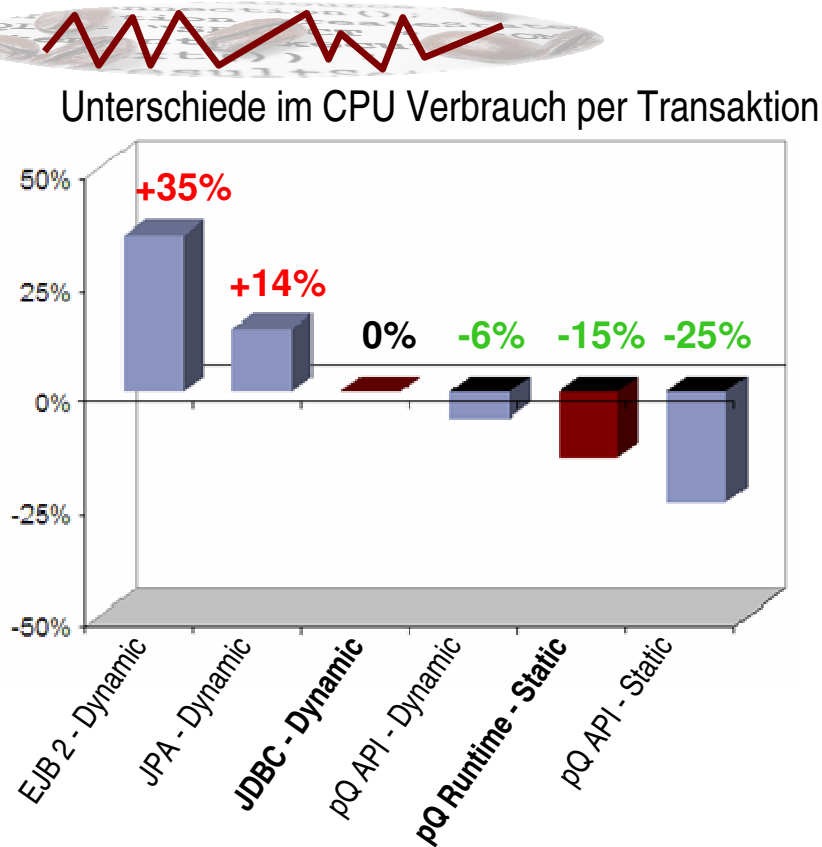
pureQuery LiteralReplacement

- Das **Verwenden von Prepared Statements** kann Abhilfe schaffen, indem das Statement von DB2 nur mit Platzhaltern (**Parameter-Markern**) vorbereitet wird.
- Das Statement selbst und der ermittelte **Zugriffspfad** sind dabei **zunächst generisch** und werden auch so im Dynamic Statement Cache abgelegt.

```
SELECT LASTNAME FROM DSN8910.EMP WHERE EMPNO < ?
```

- Die jeweiligen Parameterwerte werden erst kurz vor der eigentlichen Ausführung eingesetzt. Im DB2 erfolgt lediglich noch eine **eventuelle Reoptimierung** (bei **REOPT ONCE, ALWAYS** oder **AUTO**).
- Dies steigert die Wiederwendbarkeit der Informationen im Cache erheblich und erlaubt zudem die Nutzung von zusätzlichen Statement Caches auf Anwendungsserverebene.
- **ÄHNLICH** dem Vorgehen mit Hostvariablen bei COBOL, etc...
- Zudem **verbessertes Sicherheitskonzept**, da Verhinderung von SQL Injection.


Beim Blick auf den CPU Verbrauch schneidet pureQuery ebenfalls gut ab



- Beim statischen Ausführen der Statements mittels pureQuery Client Optimization ließ sich der **CPU Verbrauch** im Vergleich zu dynamischen JDBC Zugriffen pro Transaktion um ca. 15% **senken**, auf **z/OS LPAR Sicht** um 5%.

Verwendung von pureQuery in den verschiedensten Systemlandschaften

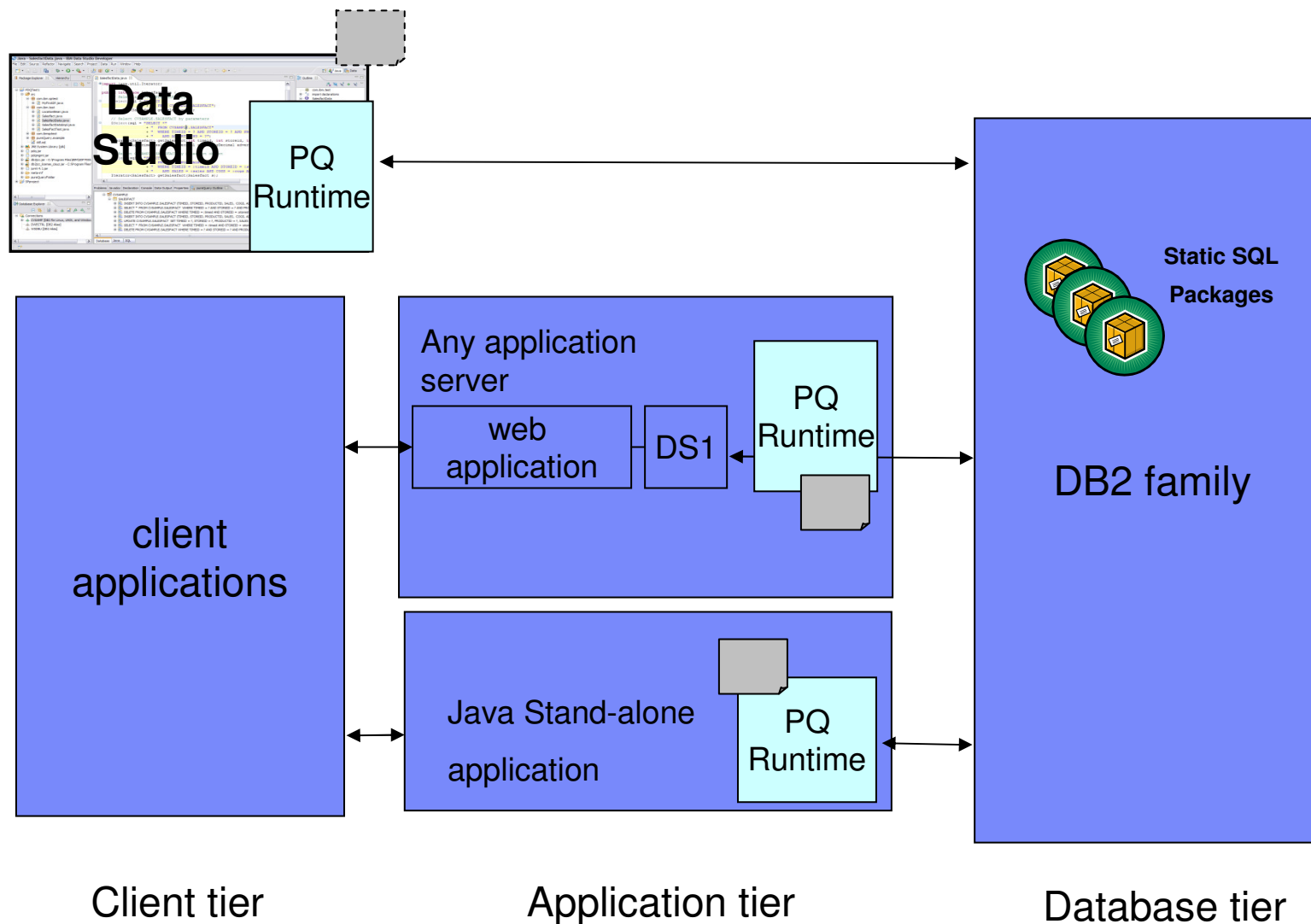
Technische Anforderungen zur pureQuery Runtime V2.2 Nutzung *

- 
- **z/OS oder verteilte Plattformen (AIX, Linux, Sun Solaris, Windows)**
 - **DB2 for z/OS V8, DB2 for LUW V8.2, DB2 for i5/OS V5R3, IDS V10**
 - **IBM Data Server Driver for JDBC and SQLJ, Version 3.57**
 - **Java Runtime Environment, Version 5.0**

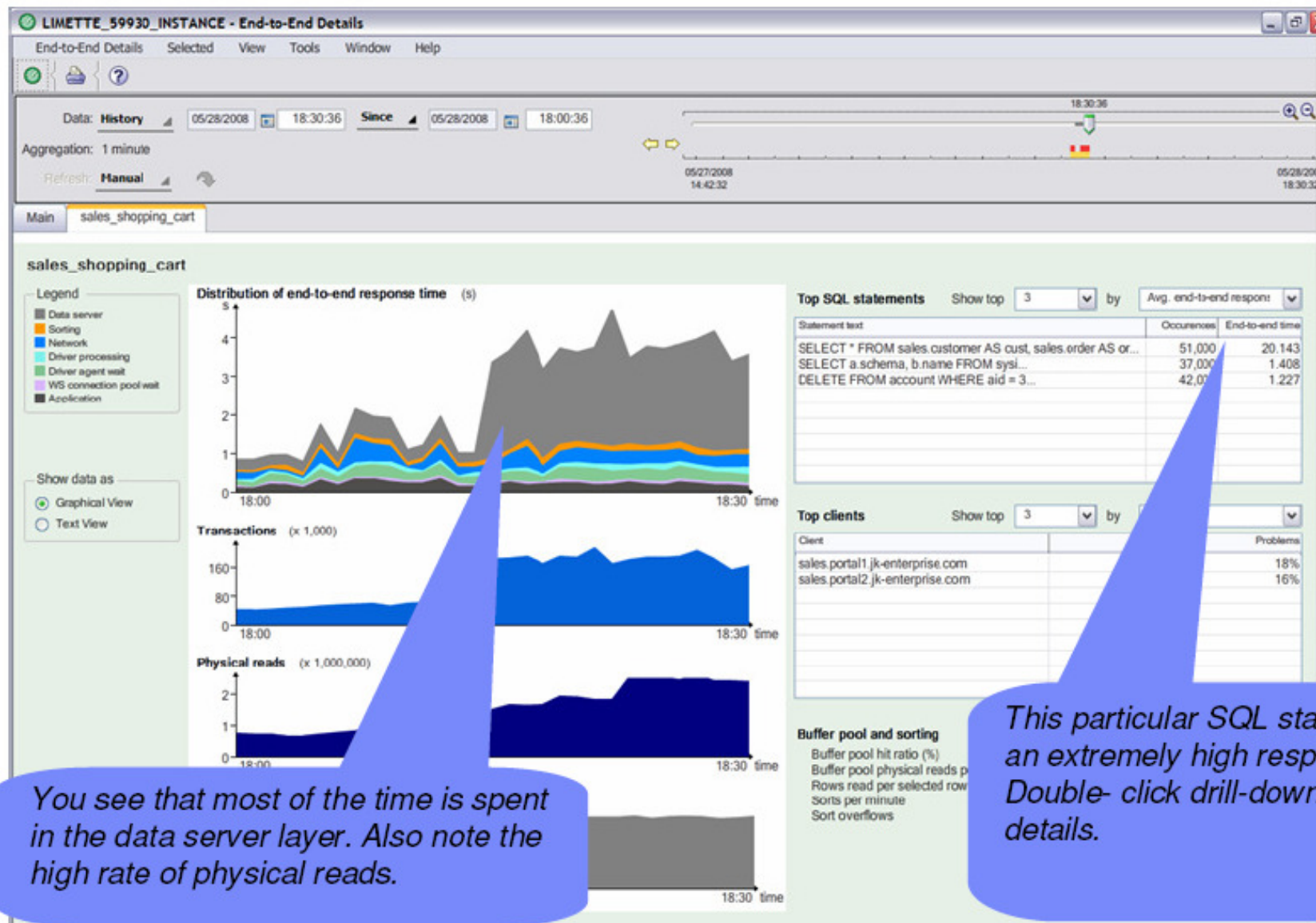
* Minimalanforderungen, d.h. angegebene Version oder höher

A typical pureQuery deployment topology

After deployment



Optional: IBM Performance Expert – Extended Insight Feature



Welche Anwendungen eignen sich gut?

- Anwendungen mit „statischem Charakter“
 - Viele wiederkehrende SQLs
 - Hoher Execution Count
 - Nutzung von Parameter Markers oder Literalen
 - CPU Reduzierung gesucht
 - Erweiterte Tuning Möglichkeiten gesucht
 - Statisches Sicherheitsmodell
 - Mehr Einblick erwünscht

- Vorsicht bei Anwendungen, die DDL dynamisch absetzen
- Vorsicht bei Nutzung von Special Registers
 - SET CURRENT SCHEMA...

pureQuery API

Anwendungsentwickler können optional bereits während der Programmierung die pureQuery API als Alternative zu JDBC/SQLJ oder Persistenzframeworks nutzen.

Diese API kann die DB2 Zugriffsschicht erzeugen. Beim Deployment kann dann definiert werden, ob statisches oder dynamisches SQL gegen die DB2 laufen soll.

Der Client Optimization Prozess (Capture der Anwendung) ist dann nicht notwendig.

Entwicklung mit Data Studio

SQL content assist

```
public static void getEmployee (Data db)
{
    @Sql SQL Annotation
    String sql = "SELECT * FROM ";
}
Press CTRL-SPACE here
public static void main(String[]
}

EMP_PHOTO
EMP_RESUME
EMPACT
EMPLOYEE
EMPMDC
EMPPROJECT
```

Entwicklung mit Data Studio

SQL content assist

SQL validation

```
// Select informix.customer by parameters
@Select(sql="select cid, name, country, street,
Customer getCustomer(Object cid);
```

```
public static void getEmployee (Data db)
{
    @Sql
    String sql = "SELECT e. FROM EMPLOYEE e";
}
```

Press CTRL-SPACE

```
public static void main(Str
}
```

- BONUS - DECIMAL(9, 2)
- COMM - DECIMAL(9, 2)
- EDLEVEL - SMALLINT
- EMPNO - CHAR(6)
- FIRSTNAME - VARCHAR(12)
- HIREDATE - DATE

Entwicklung mit Data Studio

SQL content assist

SQL validation

SQL execution

```
String url = "jdbc:db2://  
Class.forName("com.ibm.db  
con = DriverManager.getCo  
} catch (Exception e) {  
    e.printStackTrace();  
}  
return con;  
}  
Right click on SQL statement anywhere  
and select pureQuery>Run SQL  
public static void getEmployee (D  
{  
    @Sql  
    String sql = "SELECT e.FIRSTN  
                "FROM EMPLOYEE e, EMP RESUME  
                "WHERE e.EMPNO = r.EMPNO";  
}
```

pureQuery

Run As
Debug As
Profile As
Validate
Analysis
Team
Compare With
Replace With
Link Utilities
Preferences...

Show in Database Explorer

Run SQL

Generate Bean...

Generate pureQuery code...

Generate XML

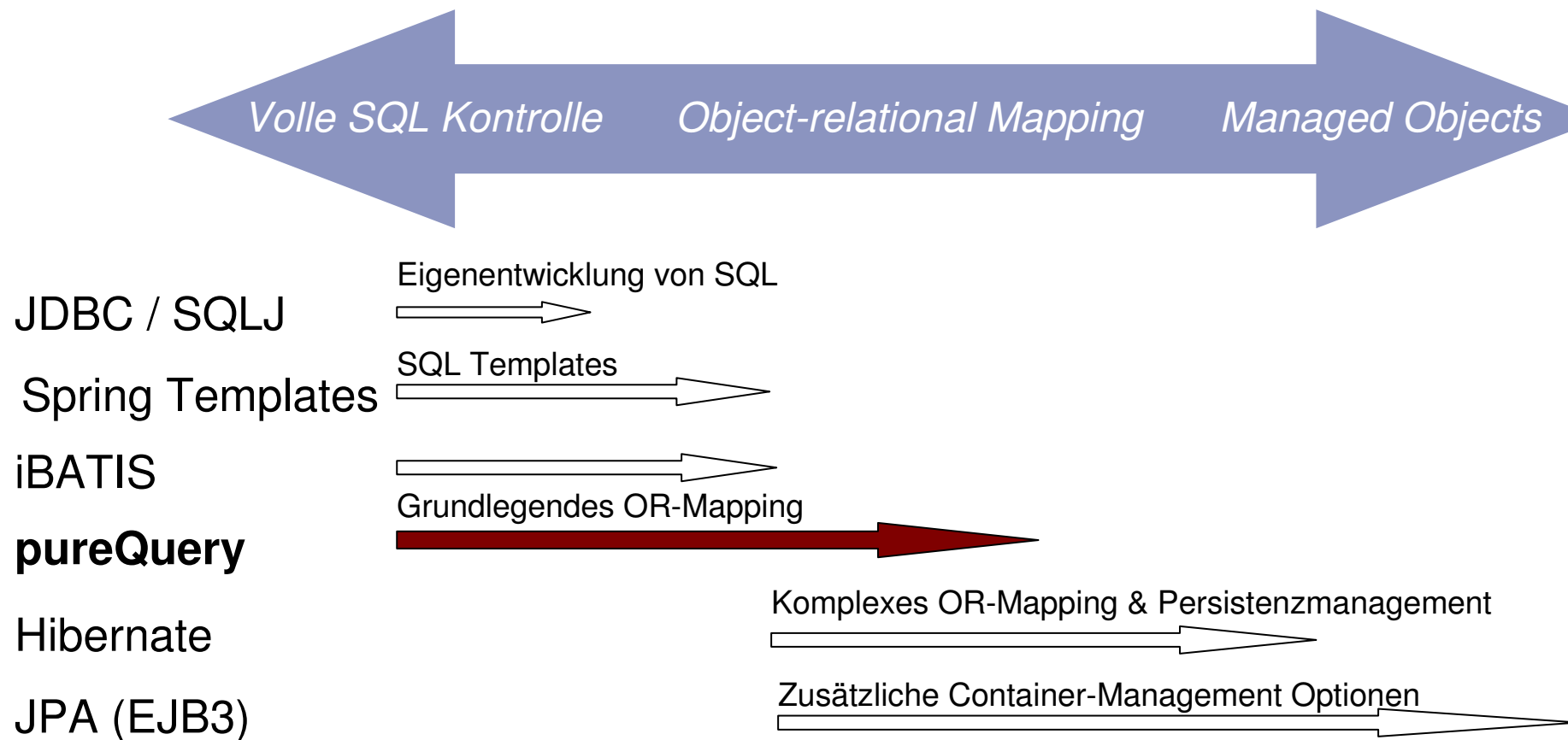
Generate DDL

Launch Visual Explain

Messages Parameters Results Profiling Data

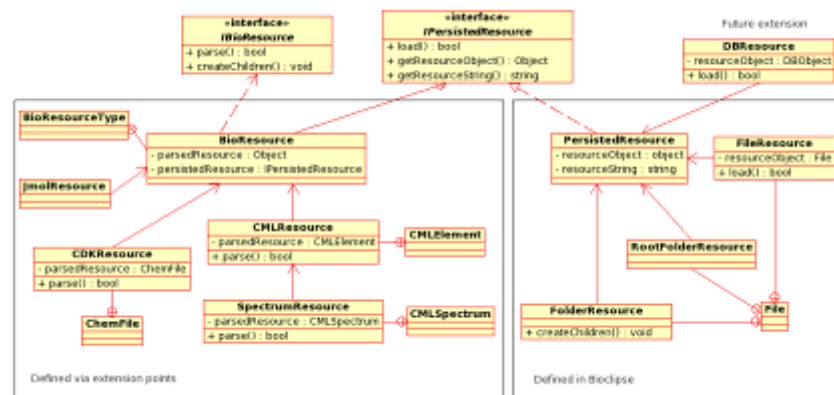
FIRSTNAME	LASTNAME	EMPNO	RESUME
DELORES	QUINTANA	000130	Resume: Delores M. Q...
DELORES	QUINTANA	000130	<!DOCTYPE HTML PUBLIC "-//IETF...
HEATHER	NICHOLLS	000140	Resume: Heather A. N...
HEATHER	NICHOLLS	000140	<!DOCTYPE HTML PUBLIC "-//IETF...
BRUCE	ADAMSON	000150	Resume: Bruce Adams

Zusätzliche Option durch Nutzung der pureQuery API als Mittelweg



Objekt-Mapping zwischen Java und Datenbankwelt mit der pureQuery API

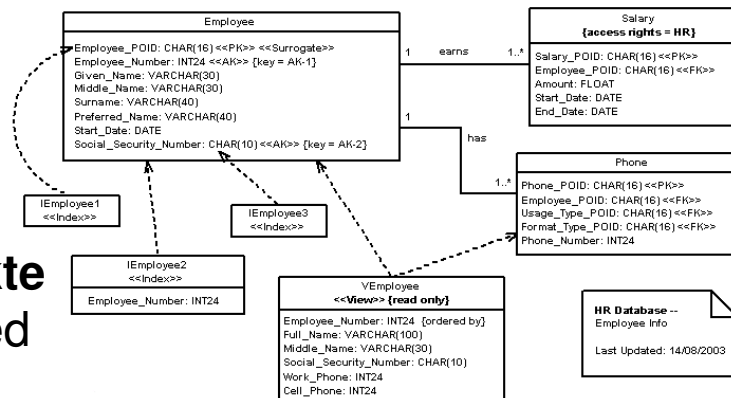
Anwendungs-
landschaft



Java-Datenzugriffsschicht

Datenmodell

SQL, Datenbankobjekte
(Tabellen, Views, Stored
Procedures, ...)



Top
Down

Meet in the
Middle

Bottom
Up

Möglichkeiten zur Realisierung des DB2-Zugriffs mit der pureQuery API

Reguläres JDBC – SQL direkt in Java-Anwendungen

```
ps = con.prepareStatement("SELECT EMPLASTNAME, EMPFIRSTNAME FROM PUREQUERY.EMPLOYEE WHERE EMPID = ?");
ps.setInt(1, id);
rs = ps.executeQuery();
rs.next();
Employee.lastname = rs.getString(1);
Employee.firstname = rs.getString(2);
```

Inline Style – SQL direkt in Java-Anwendungen

```
Iterator<Employee> Employees = db.queryIterator
    ("SELECT EMPLASTNAME, EMPFIRSTNAME FROM PUREQUERY.EMPLOYEE WHERE EMPID= :id", Employee.class, id);
```

Annotated Method Style – SQL gekapselt in Java-Interfaces

```
Iterator<Employee> Employees = data.getEmployee(id);
```

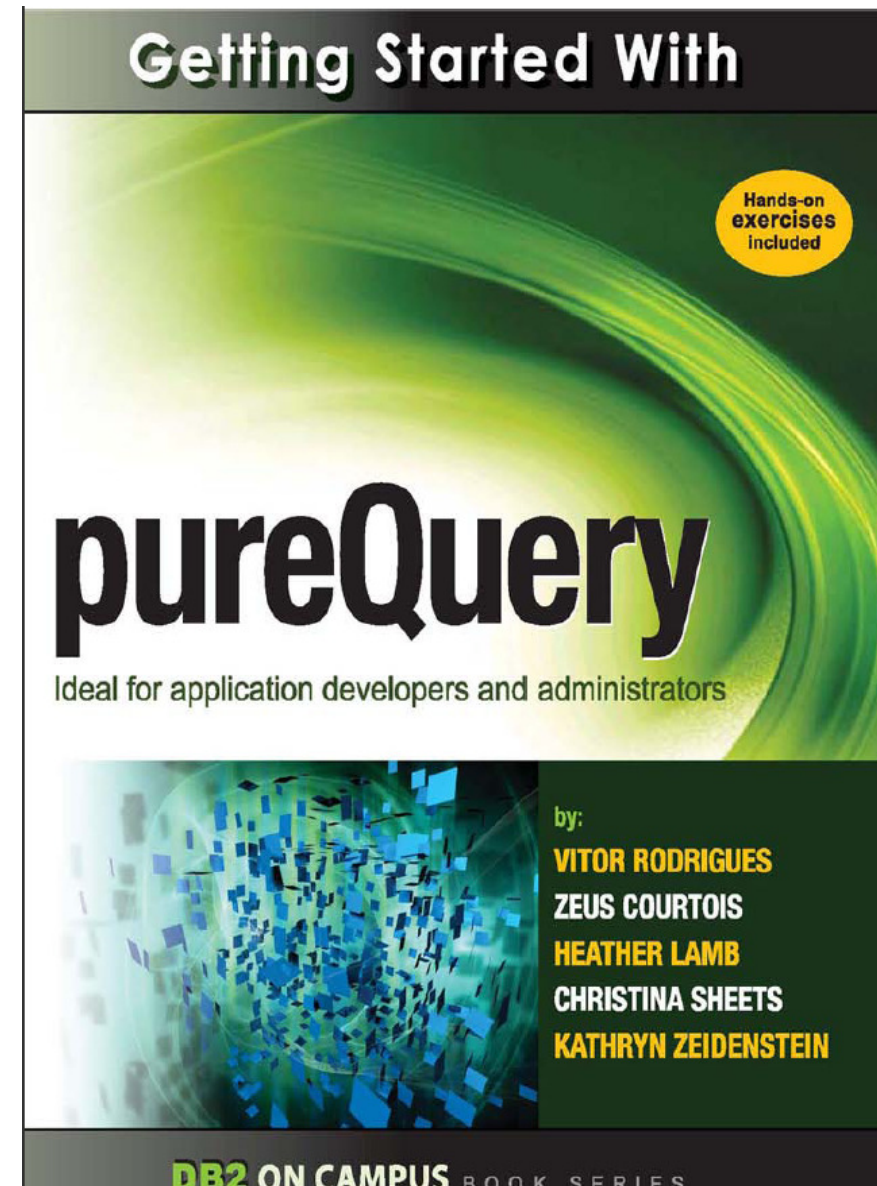
Geschäftslogik

```
@Select(sql = "SELECT EMPLASTNAME, EMPFIRSTNAME FROM PUREQUERY.EMPLOYEE WHERE EMPID = ?")
Iterator<Employee> getEmployee(int id);
```

Data Interface

Getting Started

- Gutes Einführungsbuch
- <https://www.ibm.com/developerworks/wikis/display/db2oncampus/FREE+ebook+-+Getting+started+with+pureQuery>



DB2 und Anwendungsoptimierung mit Optim

Heidrun Wietzorek, wietzo@de.ibm.com

Christian Daser, chris.daser@de.ibm.com



Angebot: Query Tuning Assessment

- Motivation: DB2 ist aufgrund der langjährigen Historie in diversen Einsatzszenarien vertreten
- Unterschiedliche Anforderungen an DB2
 - OLTP (CICS/IMS)
 - Batch Verarbeitung
 - Web (Java, ODBC,...)
 - DataWarehousing
 - Business Intelligence
 - Statistik & Geografische Datenverarbeitung
- Kostendruck als Schwerpunkt
 - IT Betrieb als Kostenfaktor
 - Projekte zur Kostensenkung
 - Mehr Workload mit den selben Strukturen/Kosten
 - Kapazitätsupgrades vermeiden
 - CPU als wichtiger Faktor auf Mainframe



→ QueryPerformance hat Potential zur CPU Reduktion