May 6-10, 2007

San Jose Convention Center

San Jose, California, USA

Session: C02

Application Performance Tuning in DB2 9 for z/OS

IDUG® 2007 North America

Gene Fuh
IBM Silicon Valley Laboratory

May 7, 2007 11:10 a.m. – 12:10 p.m.

Platform: DB2 for z/OS

GoFurther

Agenda

- Introduction
- Identifying Problem Query
- Problem Resolution with Design Advisors
- Problem Resolution with Tuning Tools
- Capturing Application Workloads
- Performing Health Check for Application Workloads with Design Advisors
- Monitoring Workload Exceptions

Introduction

Application Performance Problem

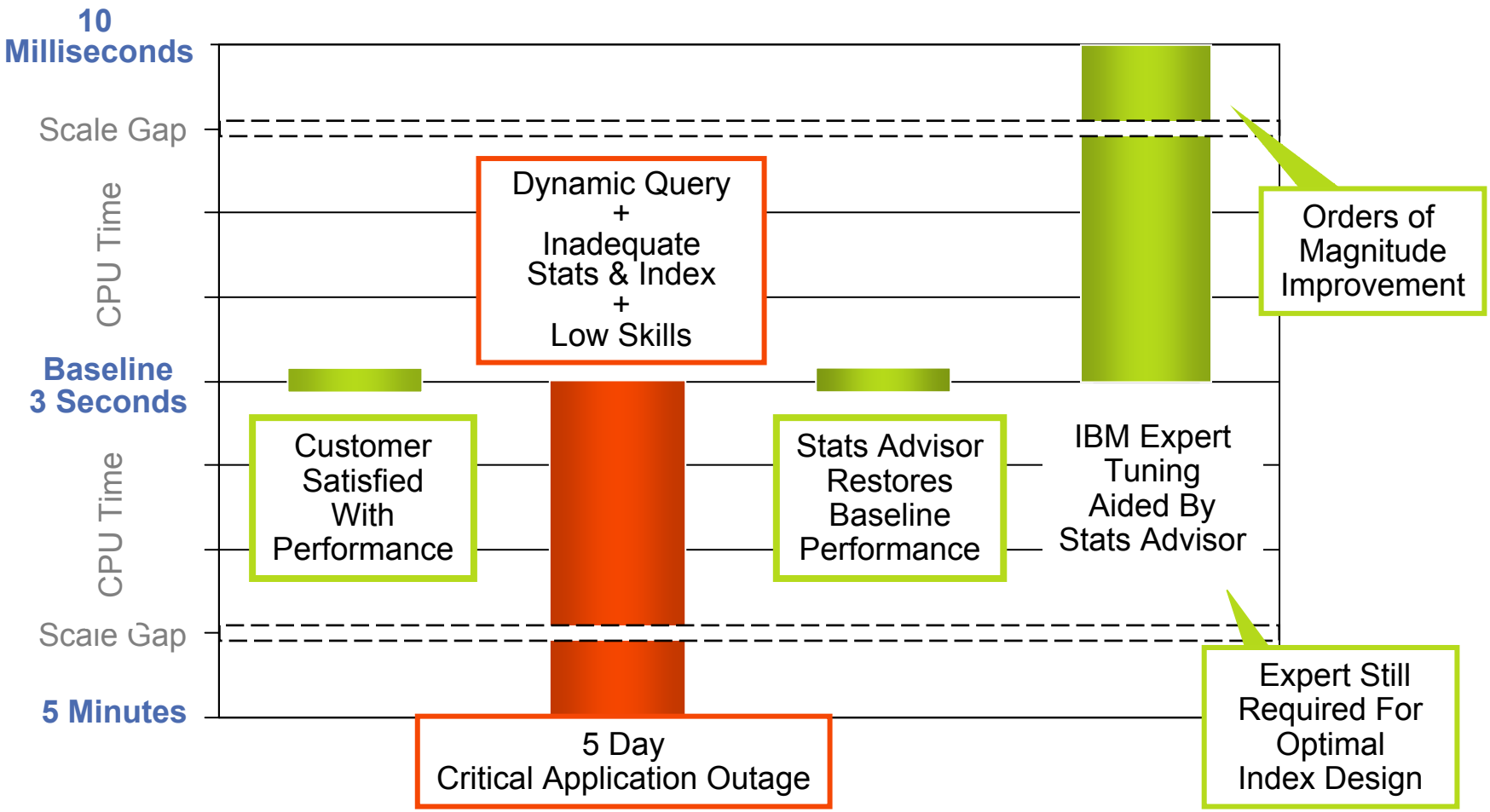
- Applications are designed and implemented very quickly
- Often there is insufficient skill and resource to perform an adequate review of SQL performance and database physical design
- Entire applications can be developed and/or enhanced with performance "surprises" discovered in production
- Tuning an entire workload requires analyzing each query in the workload, the frequency of execution, and cost of individual operations
- The overwhelming amount of resource required to perform the review often means the analysis is either not done, or done incompletely

DB2 Solution

- Problem Query Identification
  - **Snapping queries from various sources**
  - **Monitoring performance exceptions**
- Problem Query Resolution
  - **Design Advisors for recommendation of stats, index, and query design**
  - **Turing tools for deep analysis of problem query**
    - **Query format to present a readable query**
    - **Annotation of optimizer rewritten query to embed critical information**
    - **Intelligent report to show the underlying physical design with critical information**
    - **Visual explain to show the access path choice**
    - **Visual optimization hints to implement emergency solution**
    - **Service SQL to send relevant doc to IBM for diagnosis**
- Tuning Application Workload
  - **Automatic collection of application queries**
  - **Design Advisors for regular health check**
  - **Monitoring application performance exceptions**
  - **Problem resolution with Design Advisors and tuning tools**

Identifying Problem Query

Scenario: Critical application outage



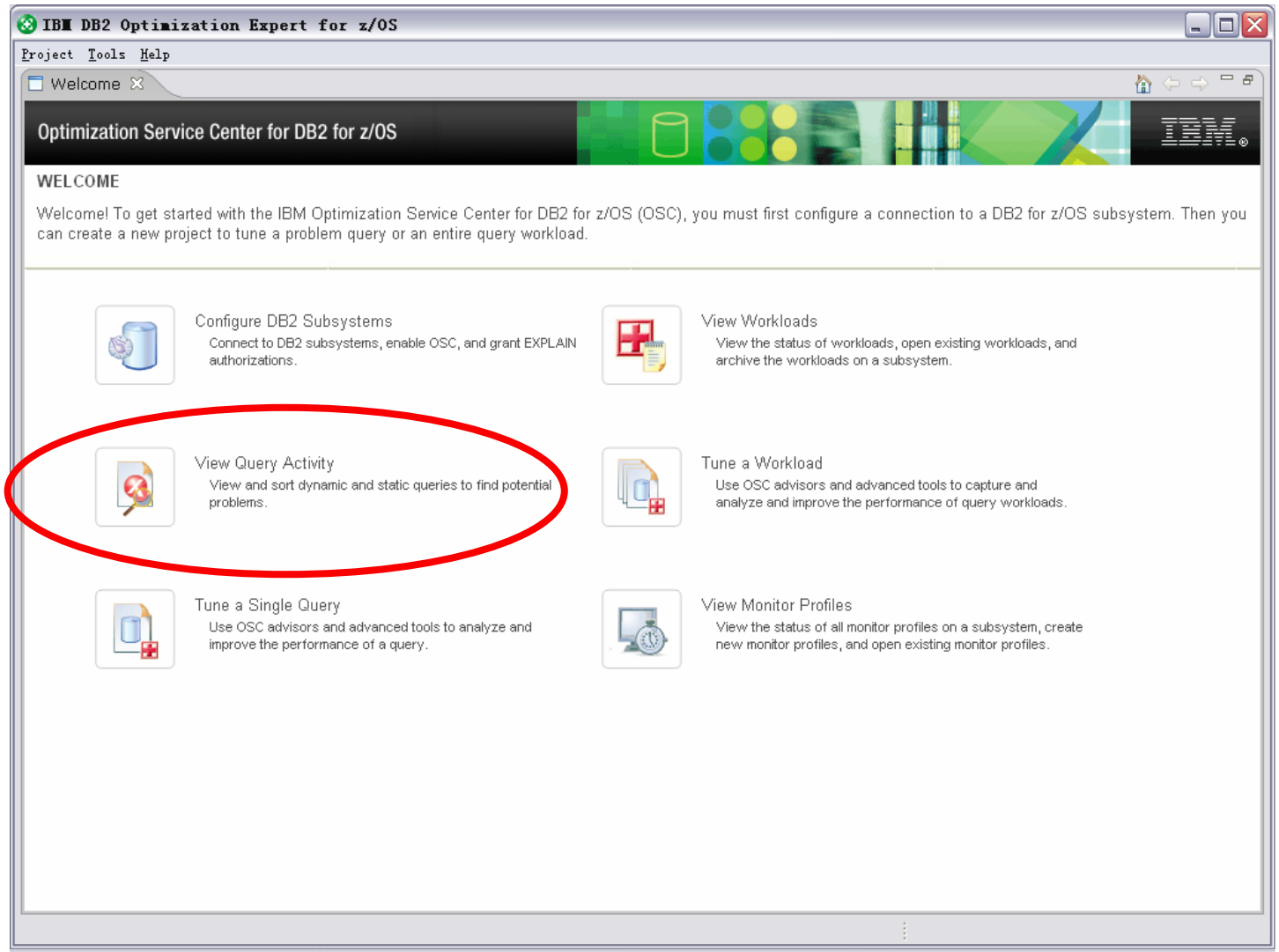
Scenario: Critical application outage

Post-mortem Analysis

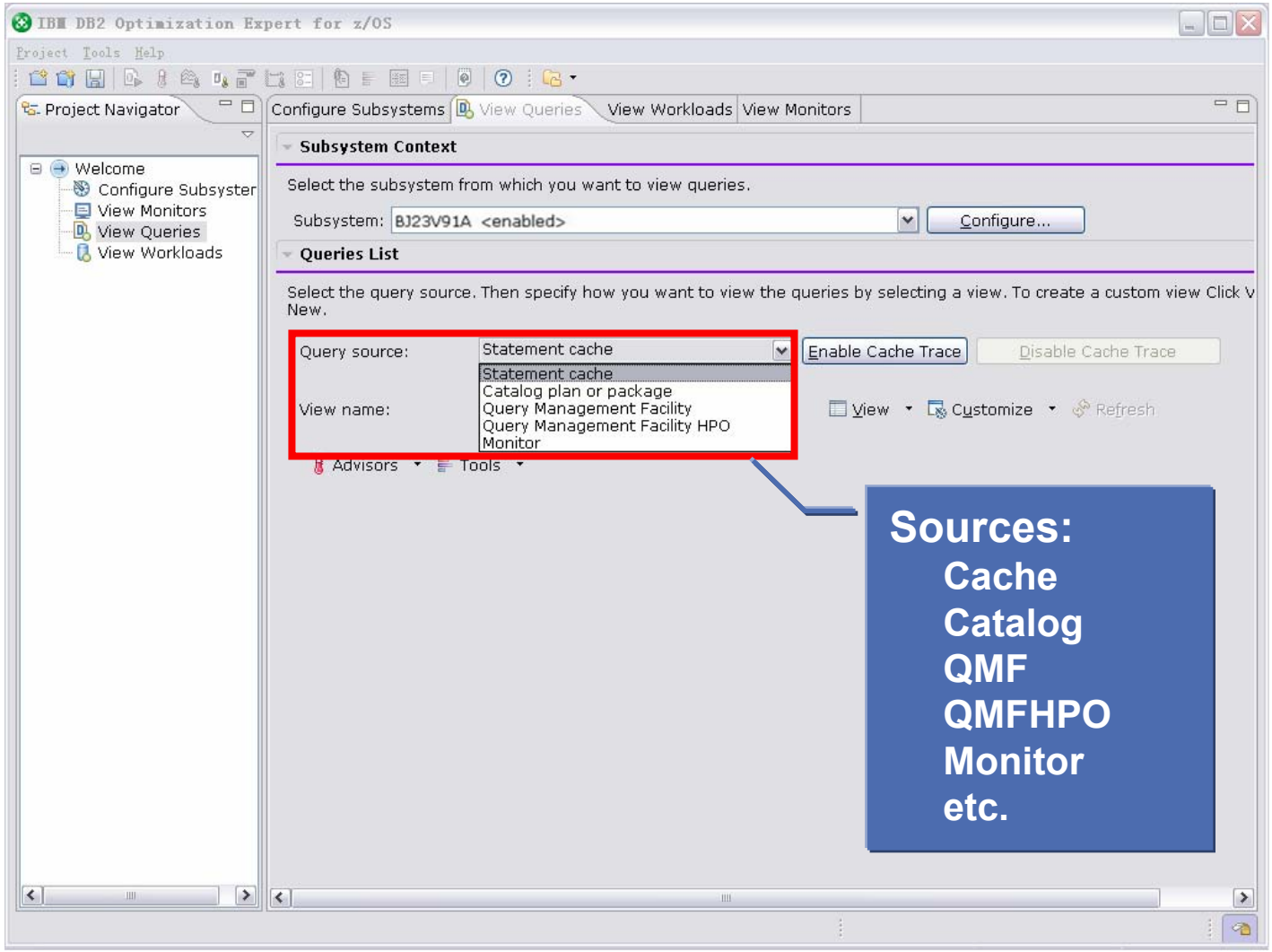
- It sounds easy but actually not
  - **Inadequate query tuning and physical database design skills**
  - **Took 3 days to identify the problem query**
  - **No idea how to investigate the performance problem**
- Customer costs
  - **Time and money lost due to application outage**
  - **Loss of confidence in IT team**
  - **Extensive and costly performance review performed for simple regression**
- Lack of expert design rules and methodology
  - **Original query performance was actual suboptimal**
  - **Good design rules and methodology would result in faster & more stable performance**

- Identifying Problem Query
  - From Dynamic Statement Cache
  - Many other query sources are supported
- Identifying Problem Query with Query Monitors
  - Automatic notification for performance exceptions

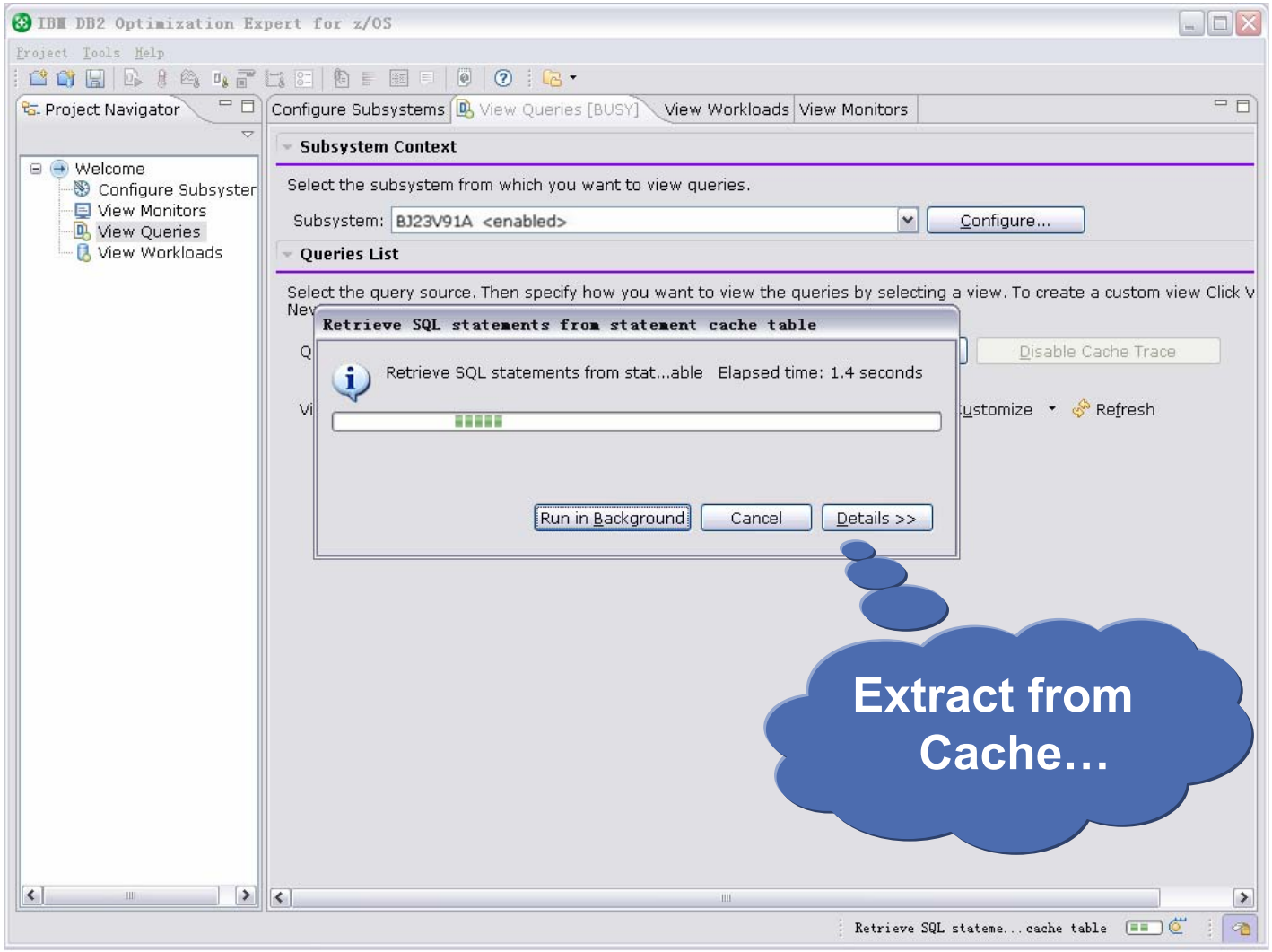
Identifying Problem Query – Welcome Panel



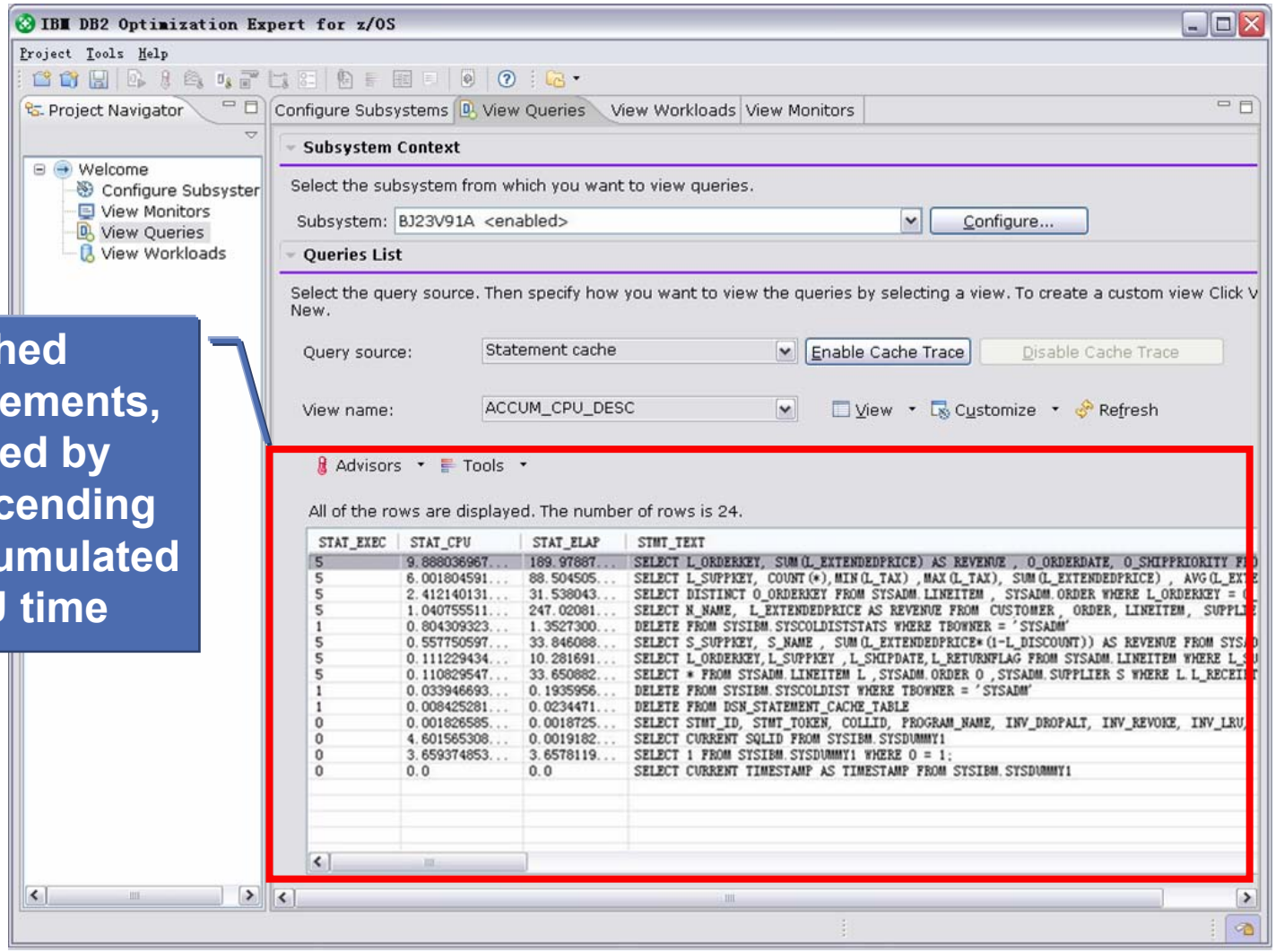
Identifying Problem Query – Dynamic Statement Cache



Identifying Problem Query – Dynamic Statement Cache

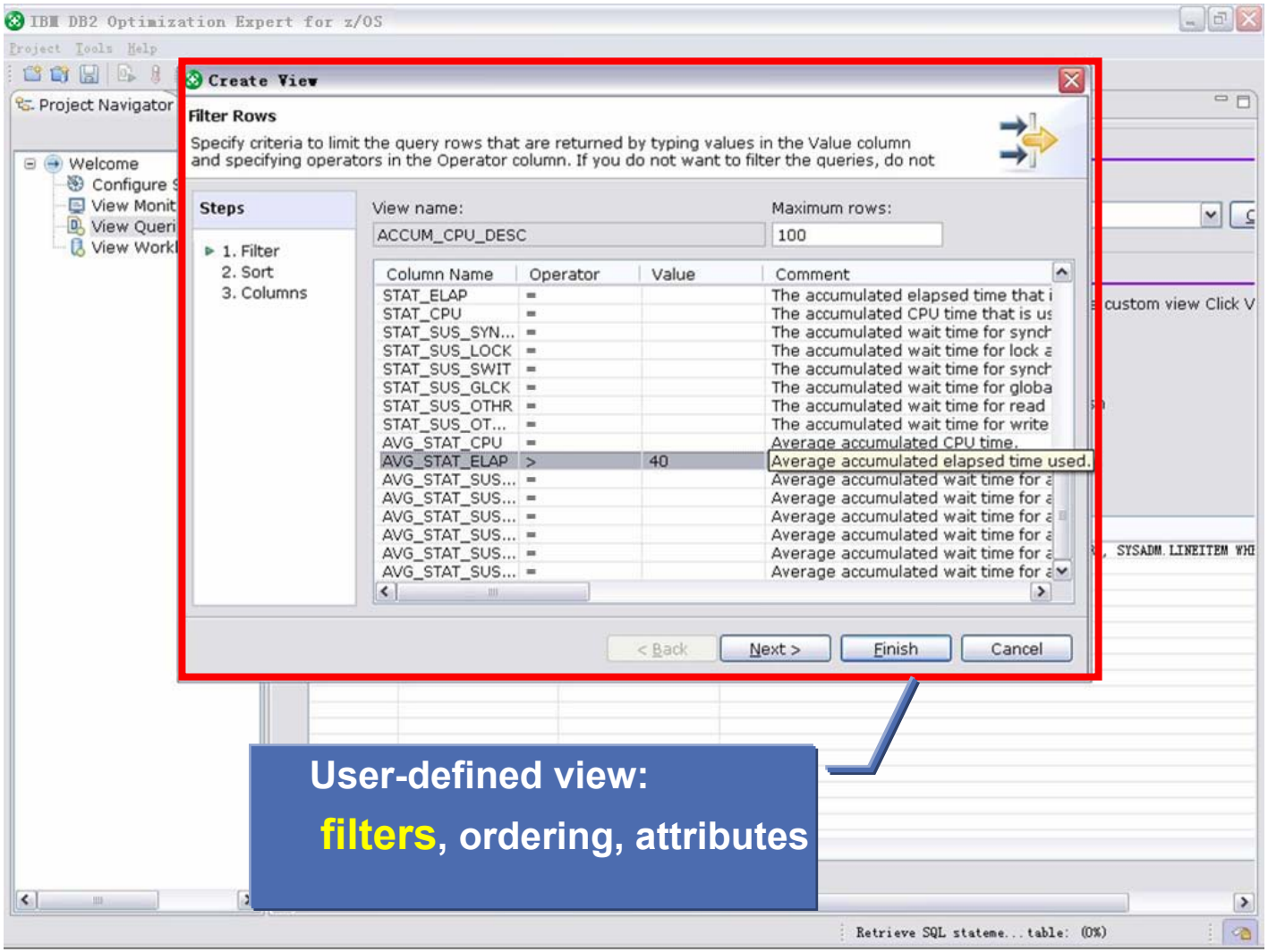


Identifying Problem Query – Dynamic Statement Cache

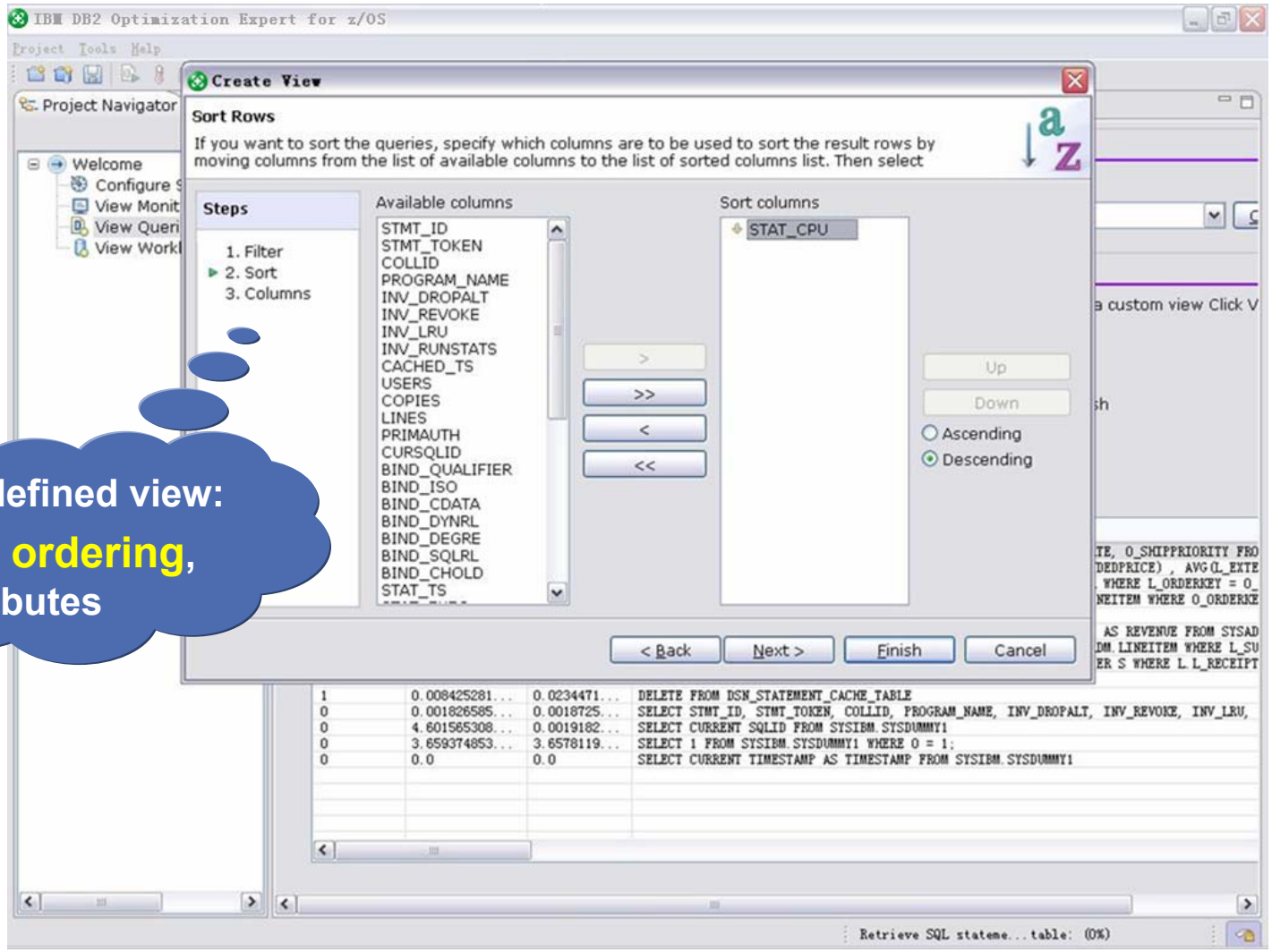


Cached Statements, sorted by descending accumulated CPU time

Identifying Problem Query – Dynamic Statement Cache

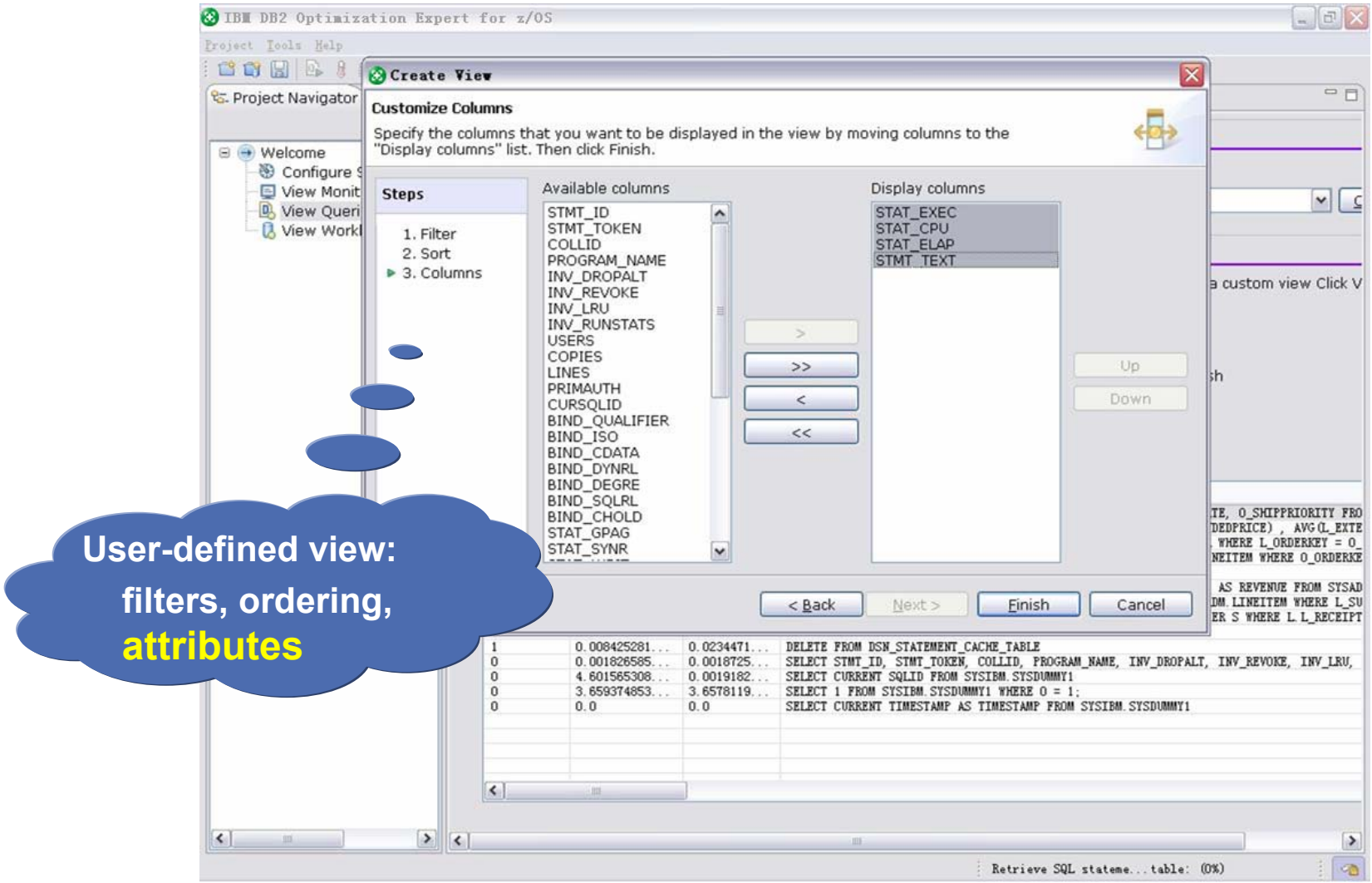


Identifying Problem Query – Dynamic Statement Cache



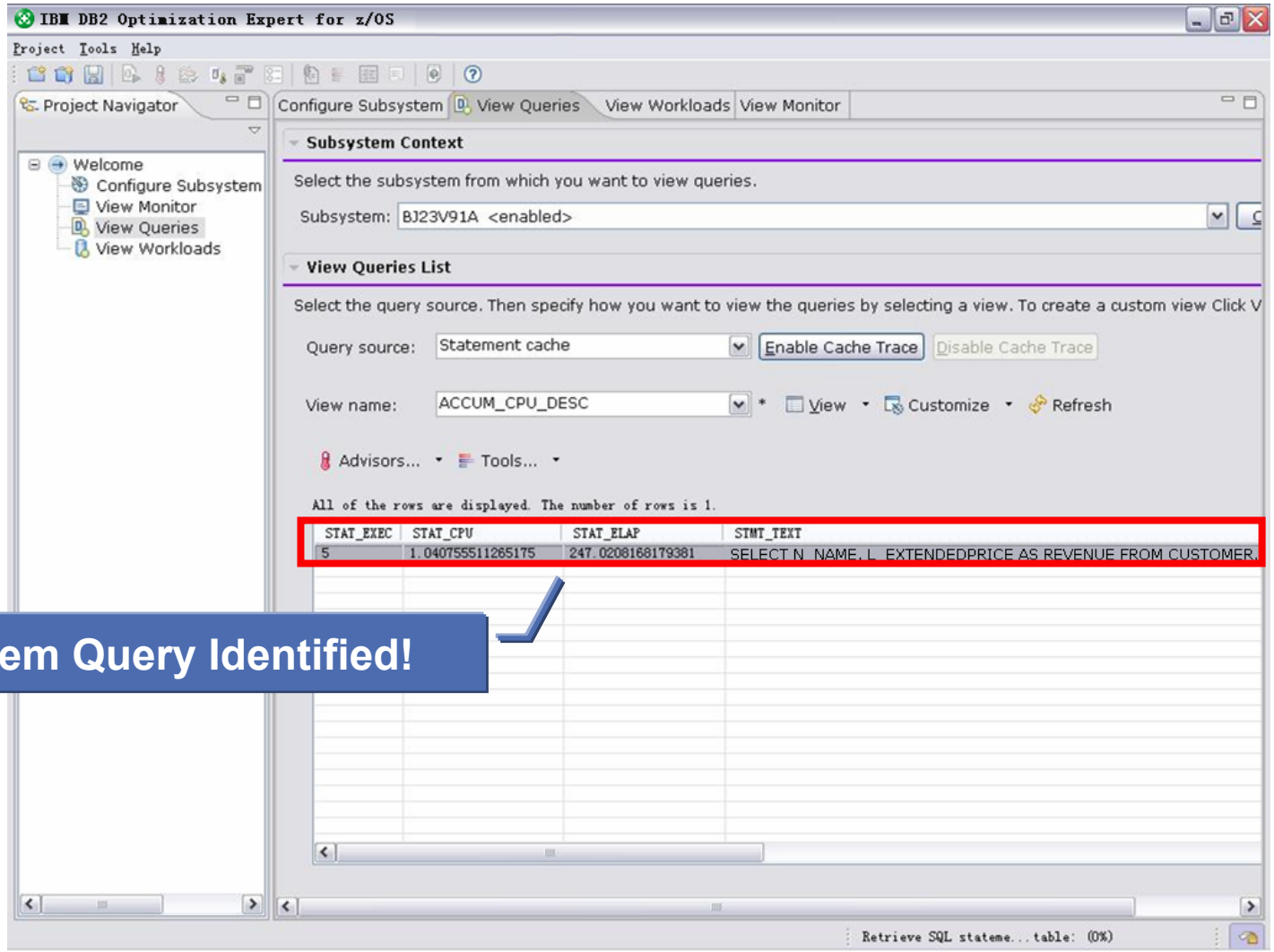
User-defined view:
filters, ordering, attributes

Identifying Problem Query – Dynamic Statement Cache



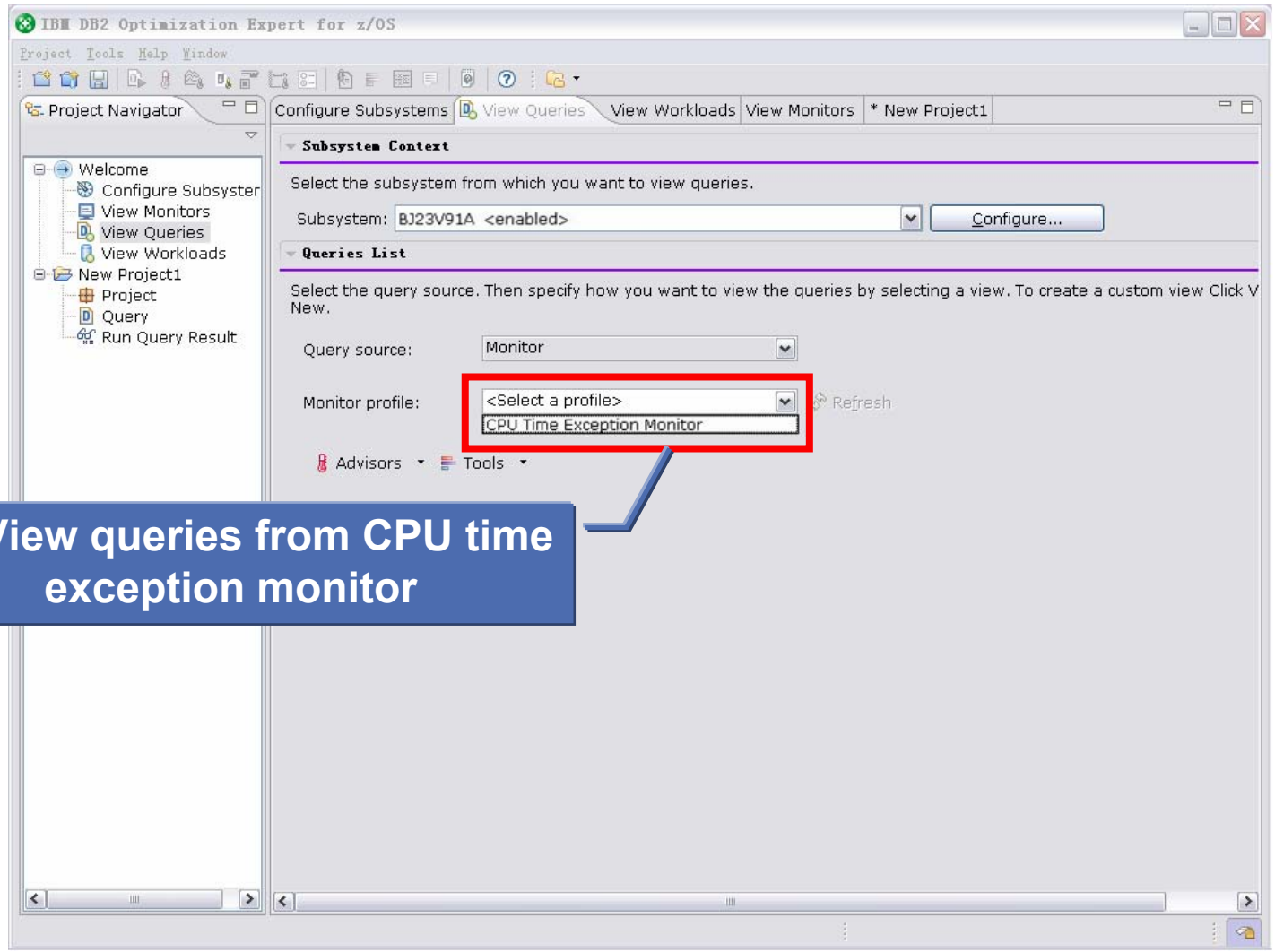
User-defined view:
filters, ordering,
attributes

Identifying Problem Query – Dynamic Statement Cache

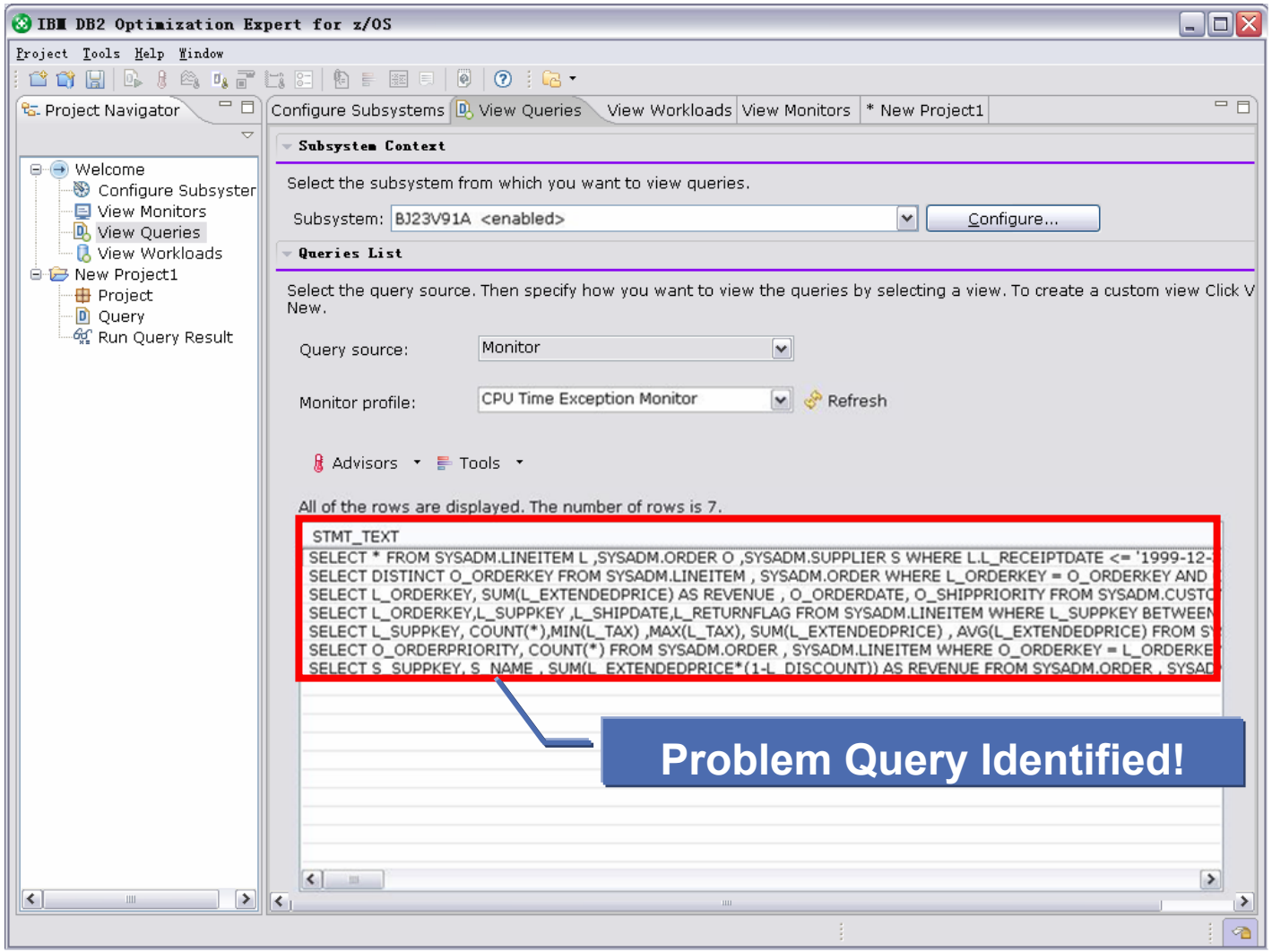


Problem Query Identified!

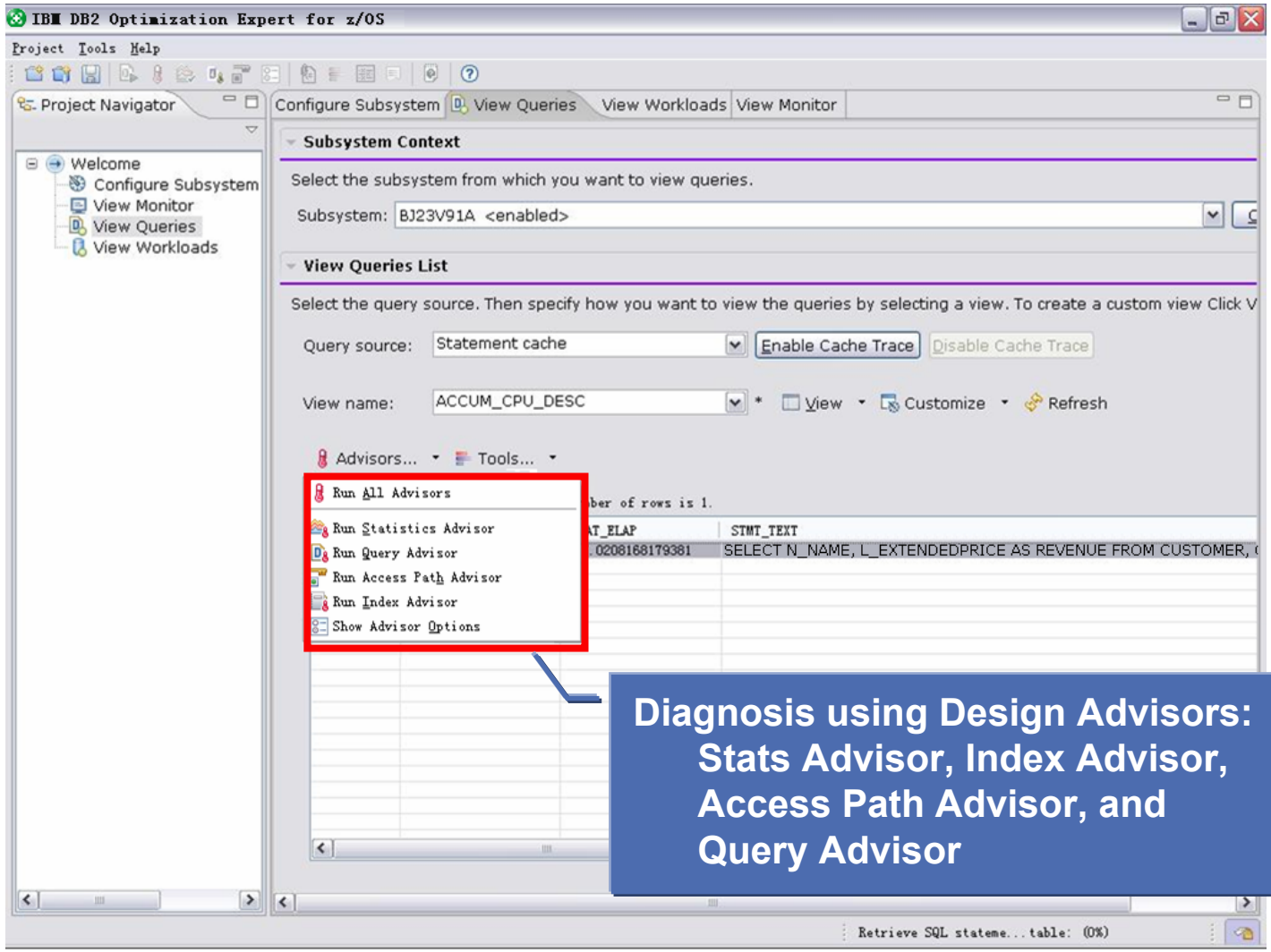
Identifying Problem Query – Query Monitor



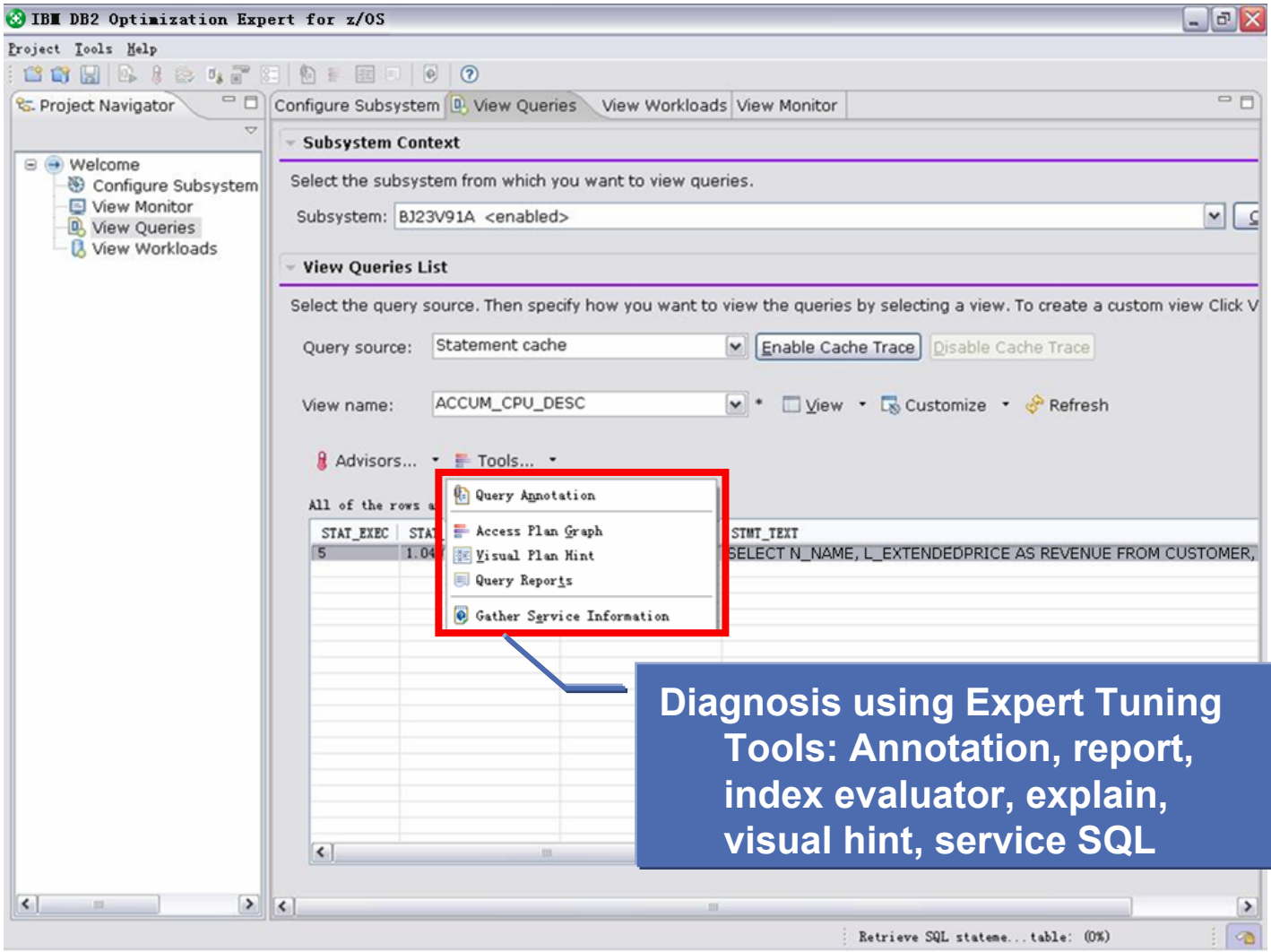
Identifying Problem Query – Query Monitor



Ready for Problem Resolution – All Users



Ready for Problem Resolution – Expert Users

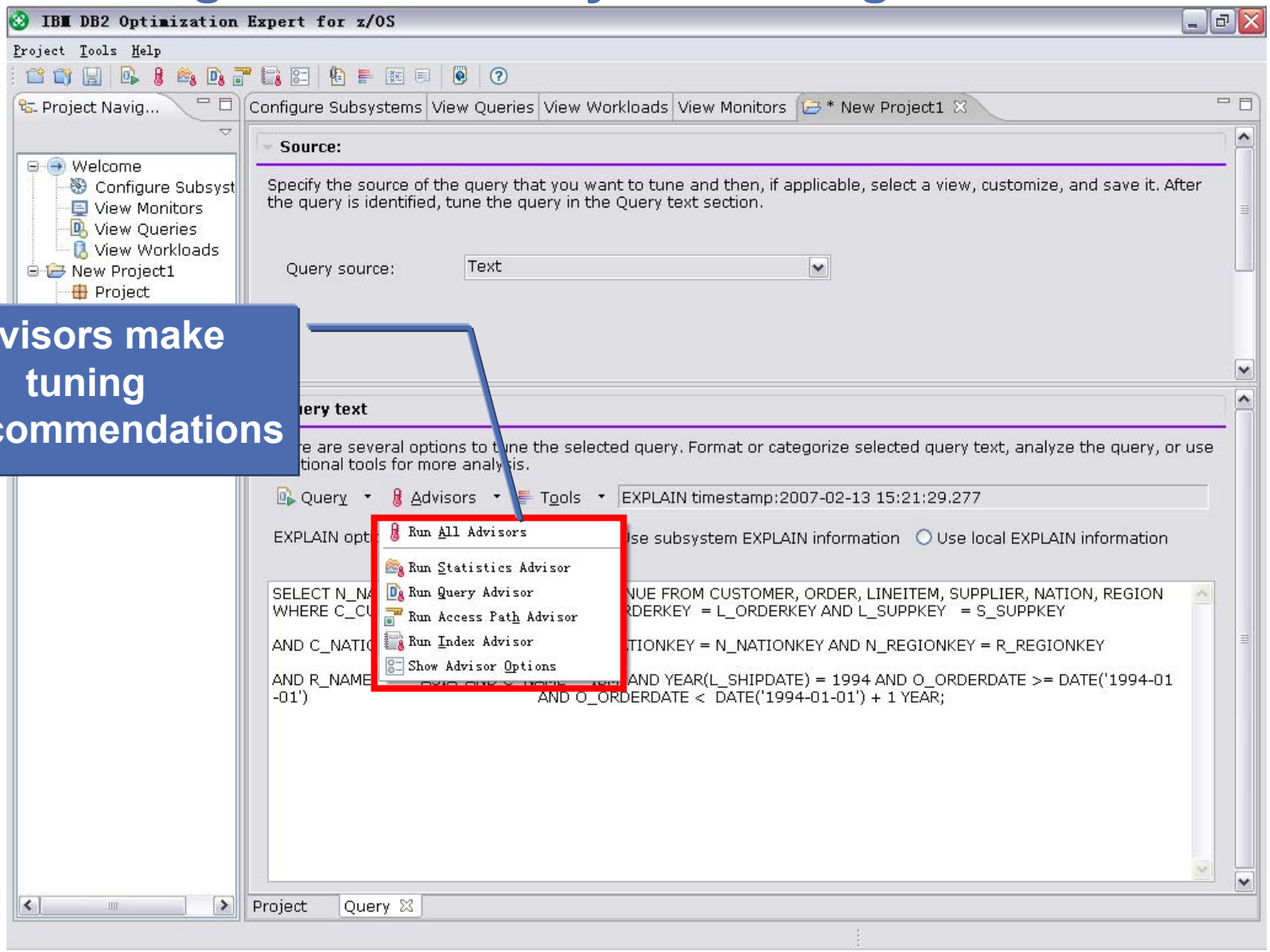


Diagnosis using Expert Tuning Tools: Annotation, report, index evaluator, explain, visual hint, service SQL

Problem Resolution with Design Advisors

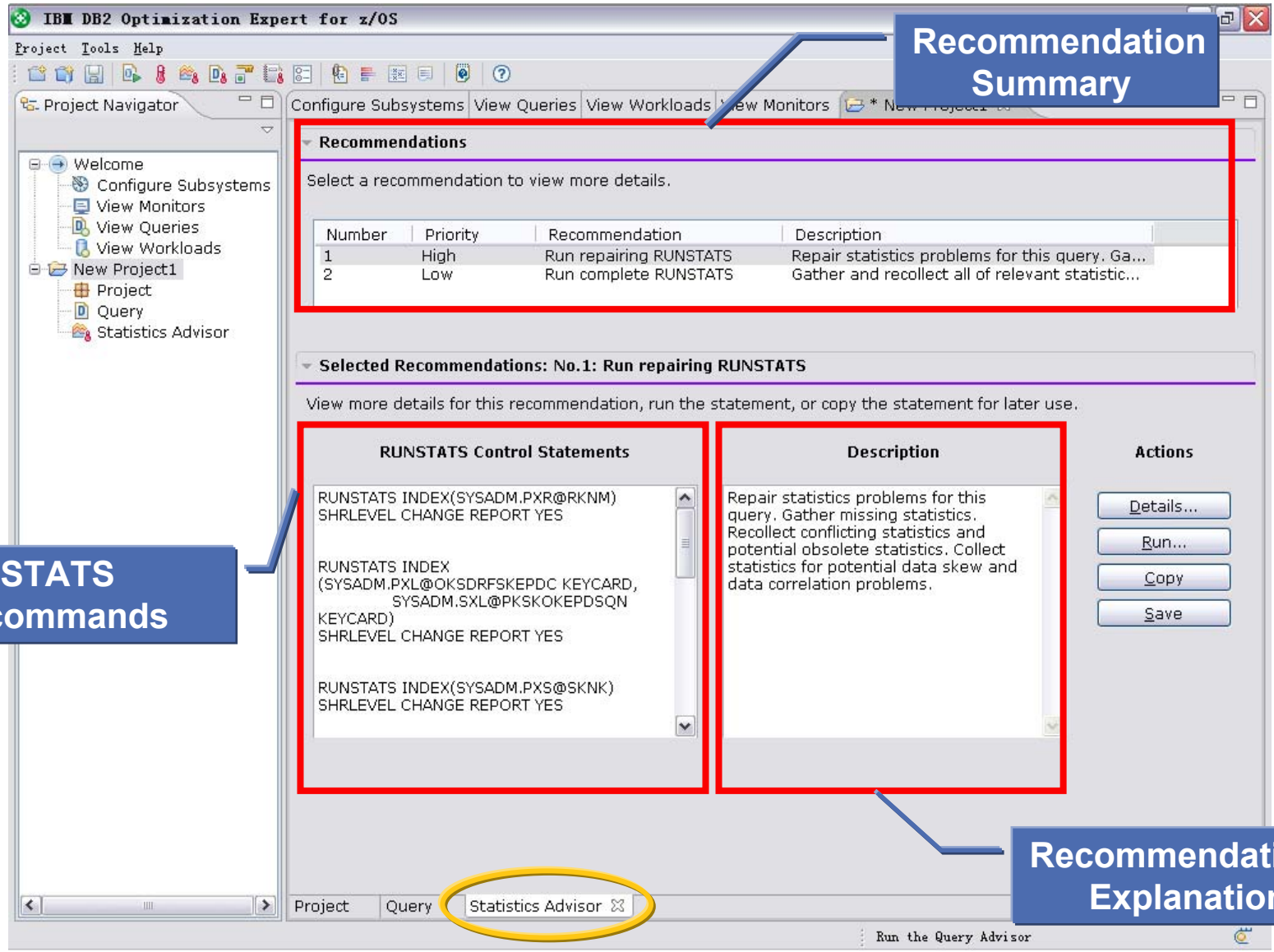
- Quick way to identify the cause of performance problem
- Recommended for all users
- Three Advisors
  - Statistics Advisor
  - Index Advisor
  - Query Advisor
  - Access Path Advisor

Tuning Problem Query with Design Advisors

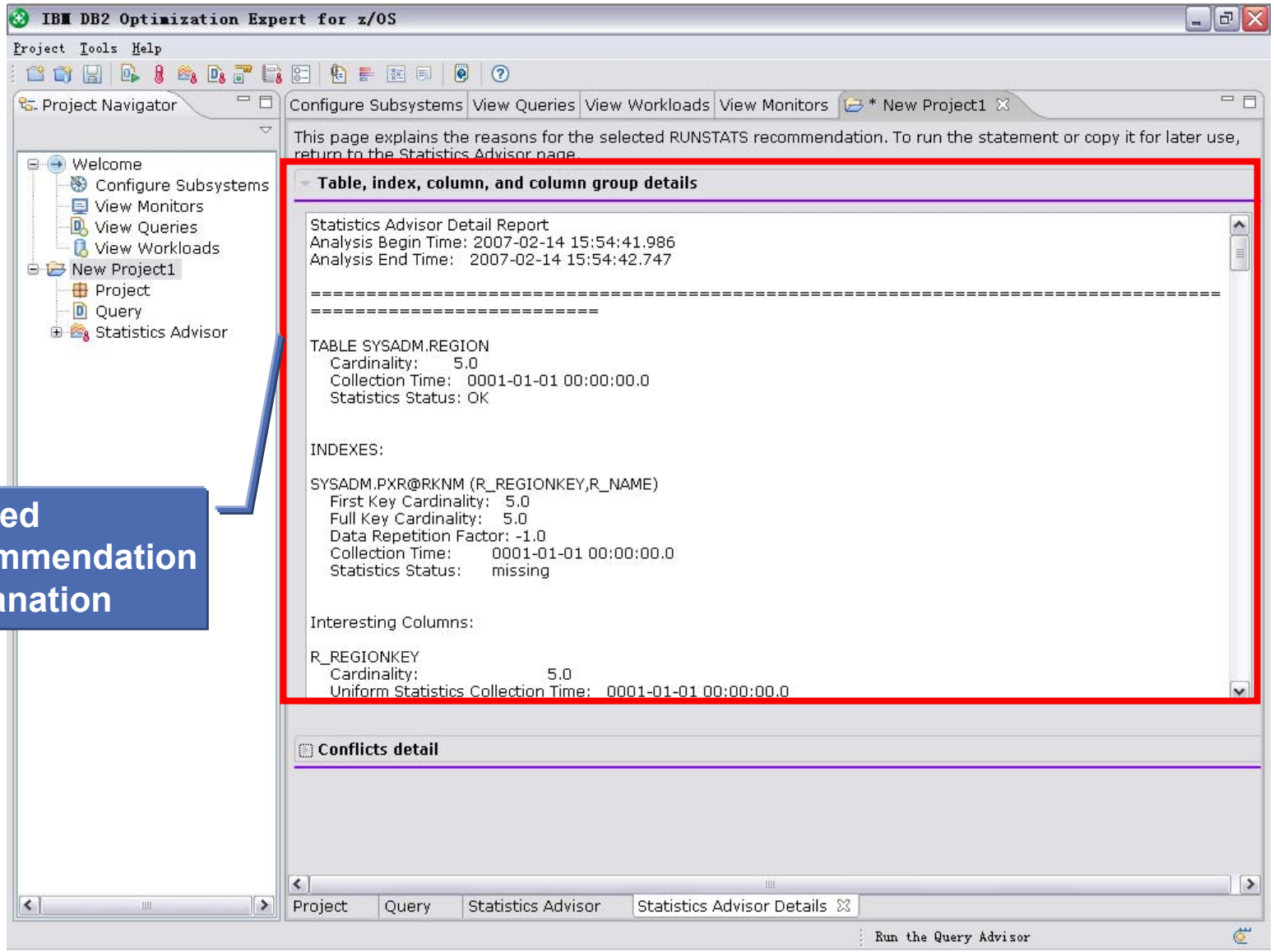


Advisors make tuning recommendations

Tuning Problem Query with Stats Advisor

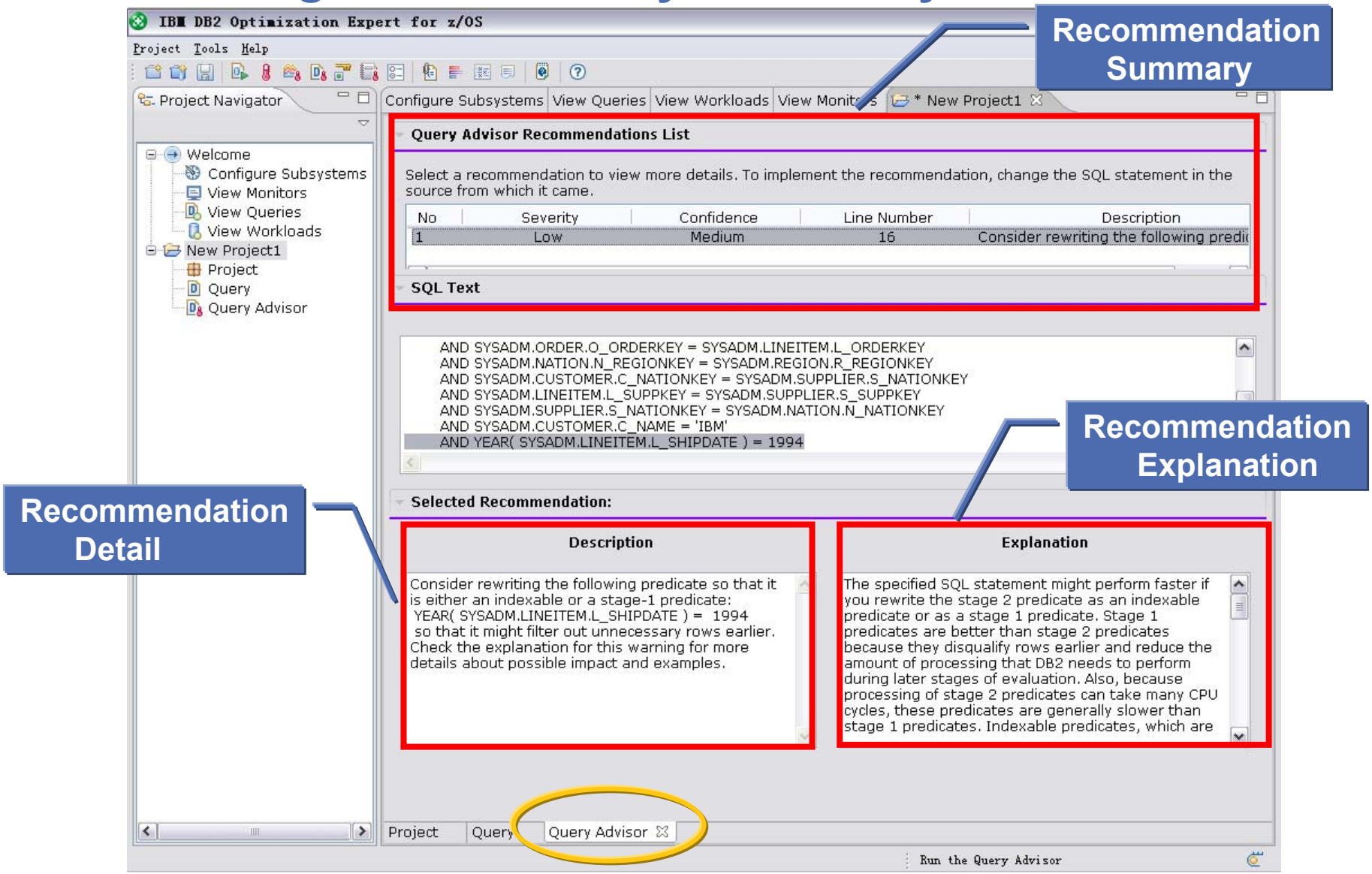


Tuning Problem Query with Stats Advisor



Detailed Recommendation Explanation

Tuning Problem Query with Query Advisor

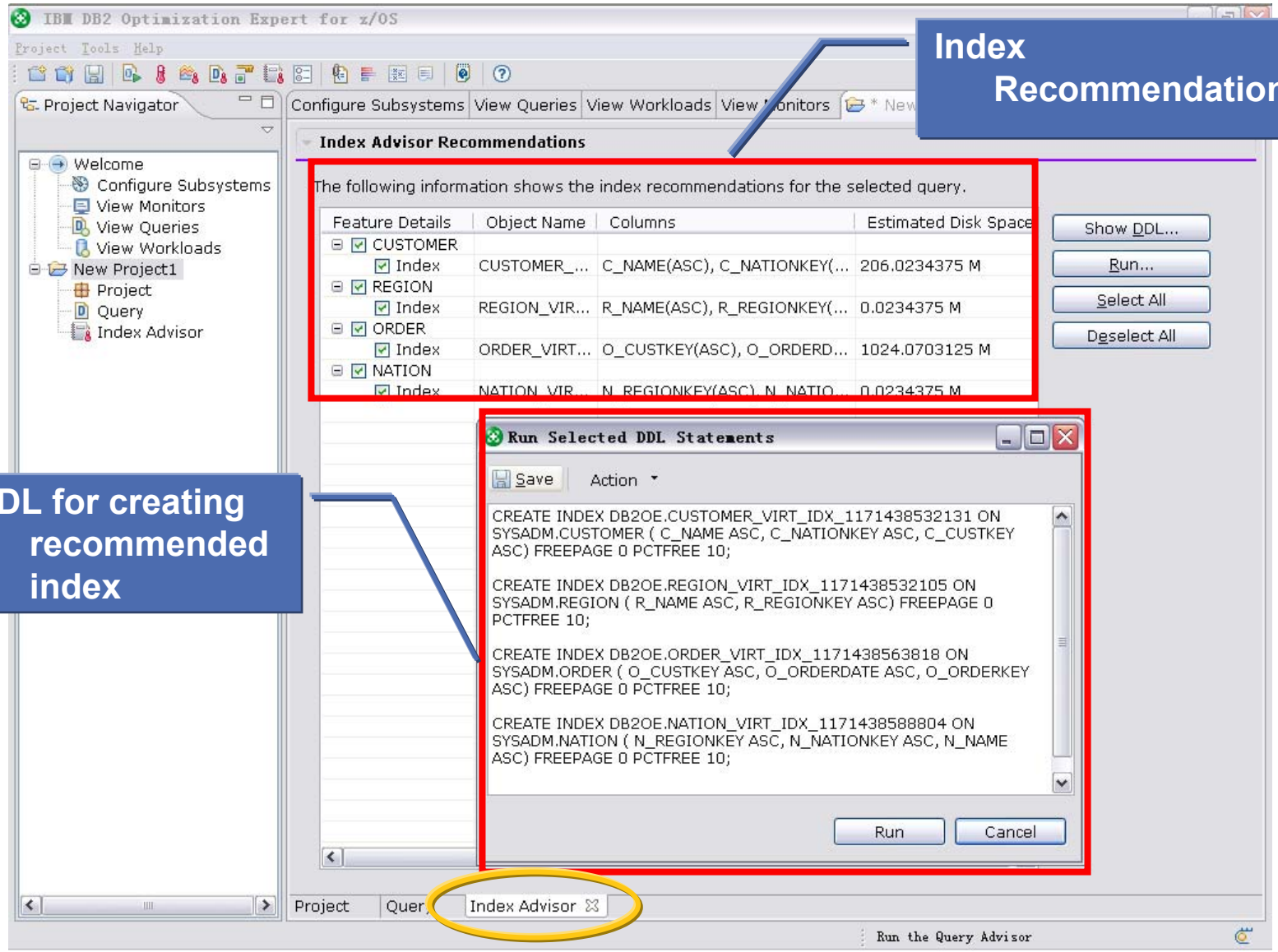


Recommendation Summary

Recommendation Explanation

Recommendation Detail

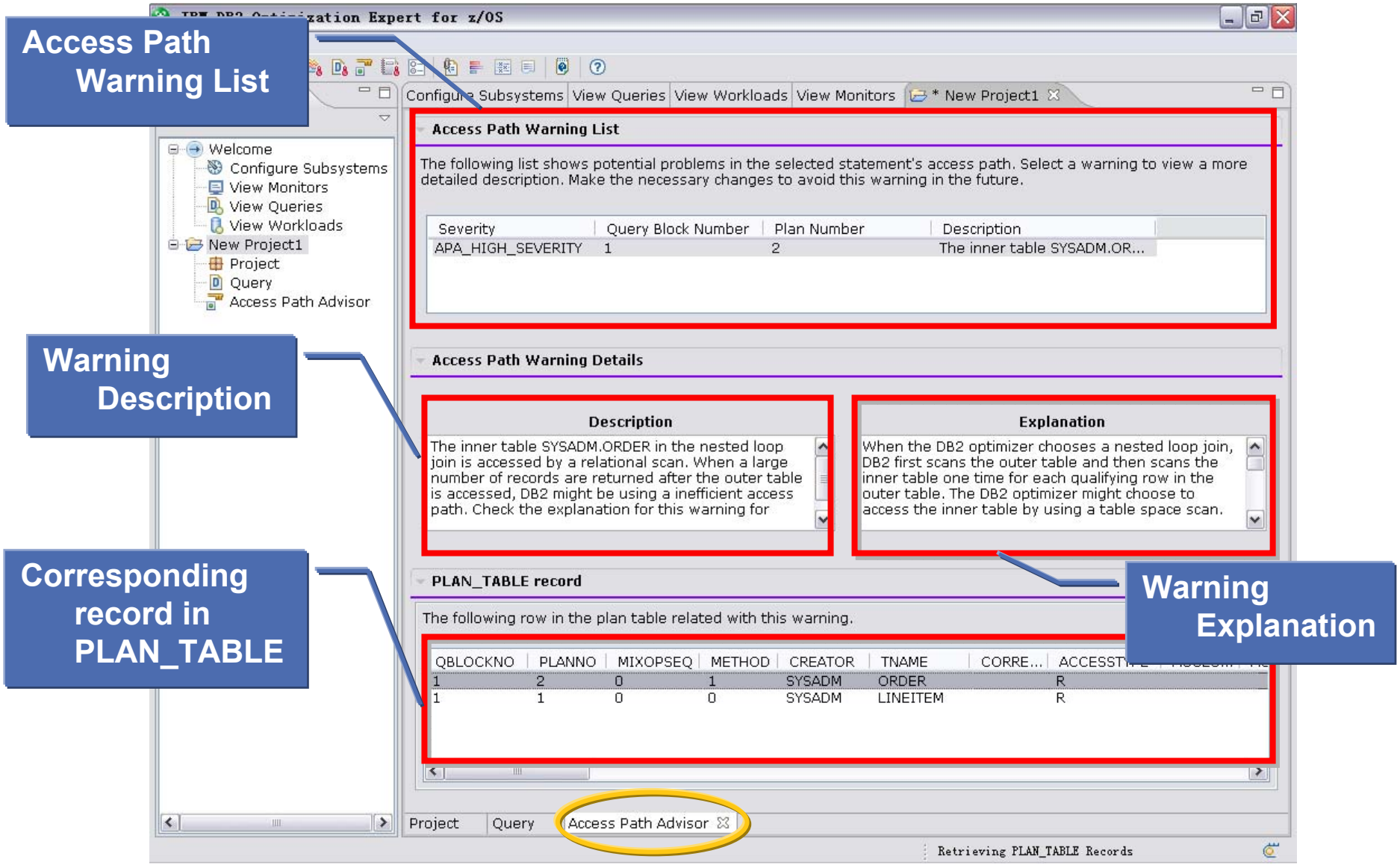
Tuning Problem Query with Index Advisor



Index Recommendation

DDL for creating recommended index

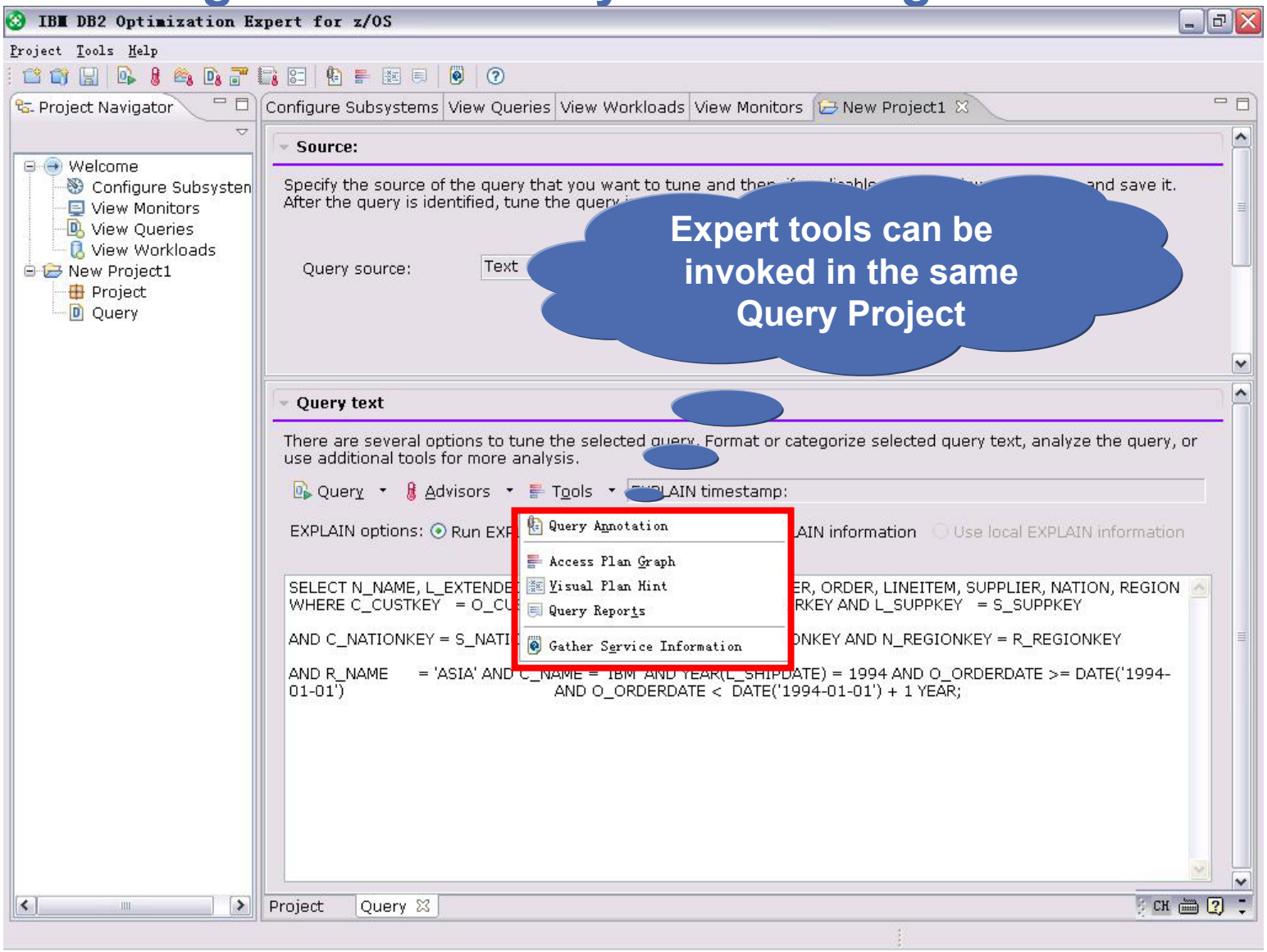
Tuning Problem Query with Access Path Advisor



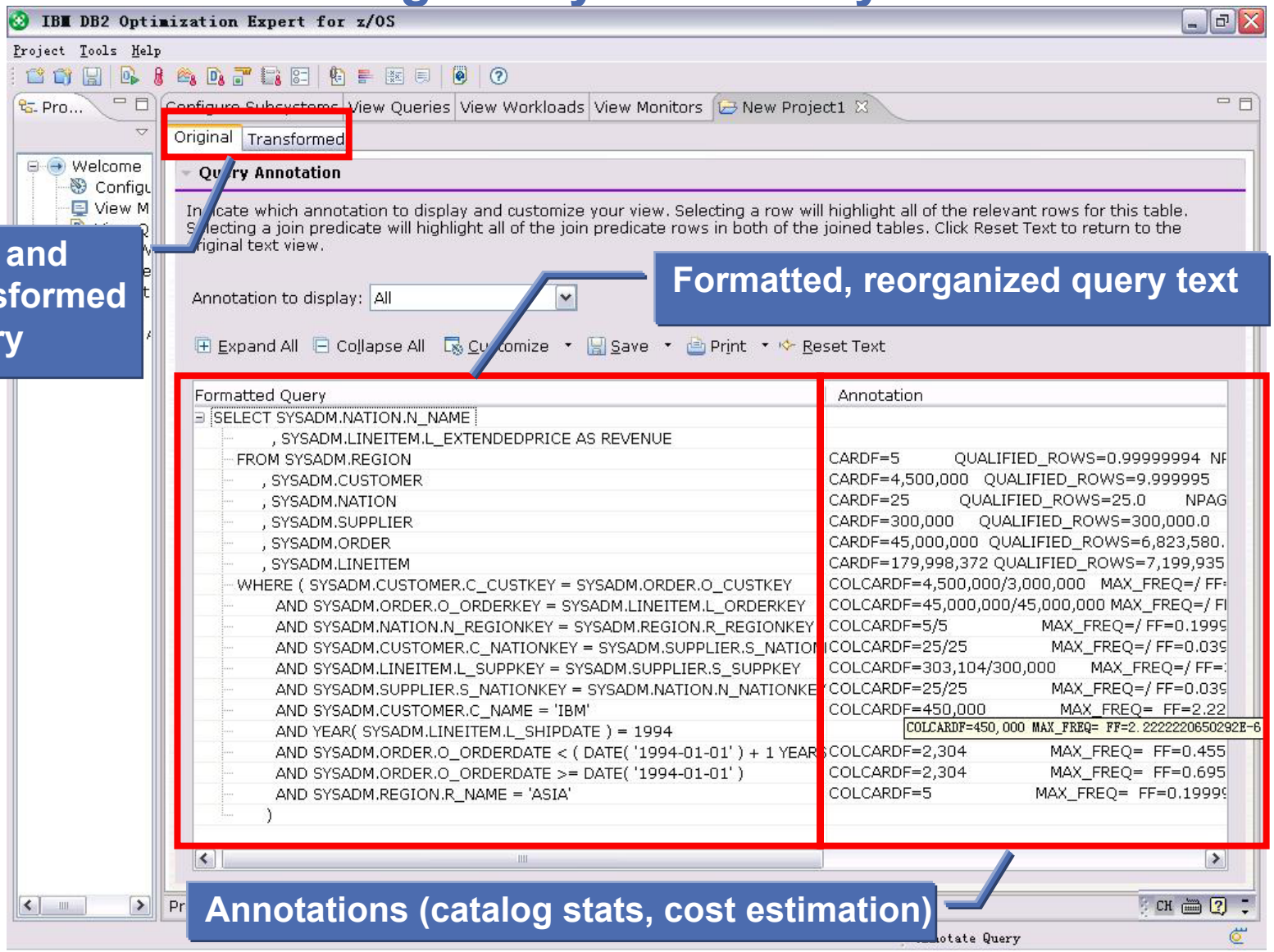
Problem Resolution with Tuning Tools

- Deep-dive into the root cause of the problem
- Recommended for expert users
- Expert Tuning Tools
  - Query Annotation
  - Query Report
  - Visual Explain
  - Visual Plan Hint
  - Service SQL

Tuning Problem Query with Tuning Tools



Understanding Query with Query Annotation

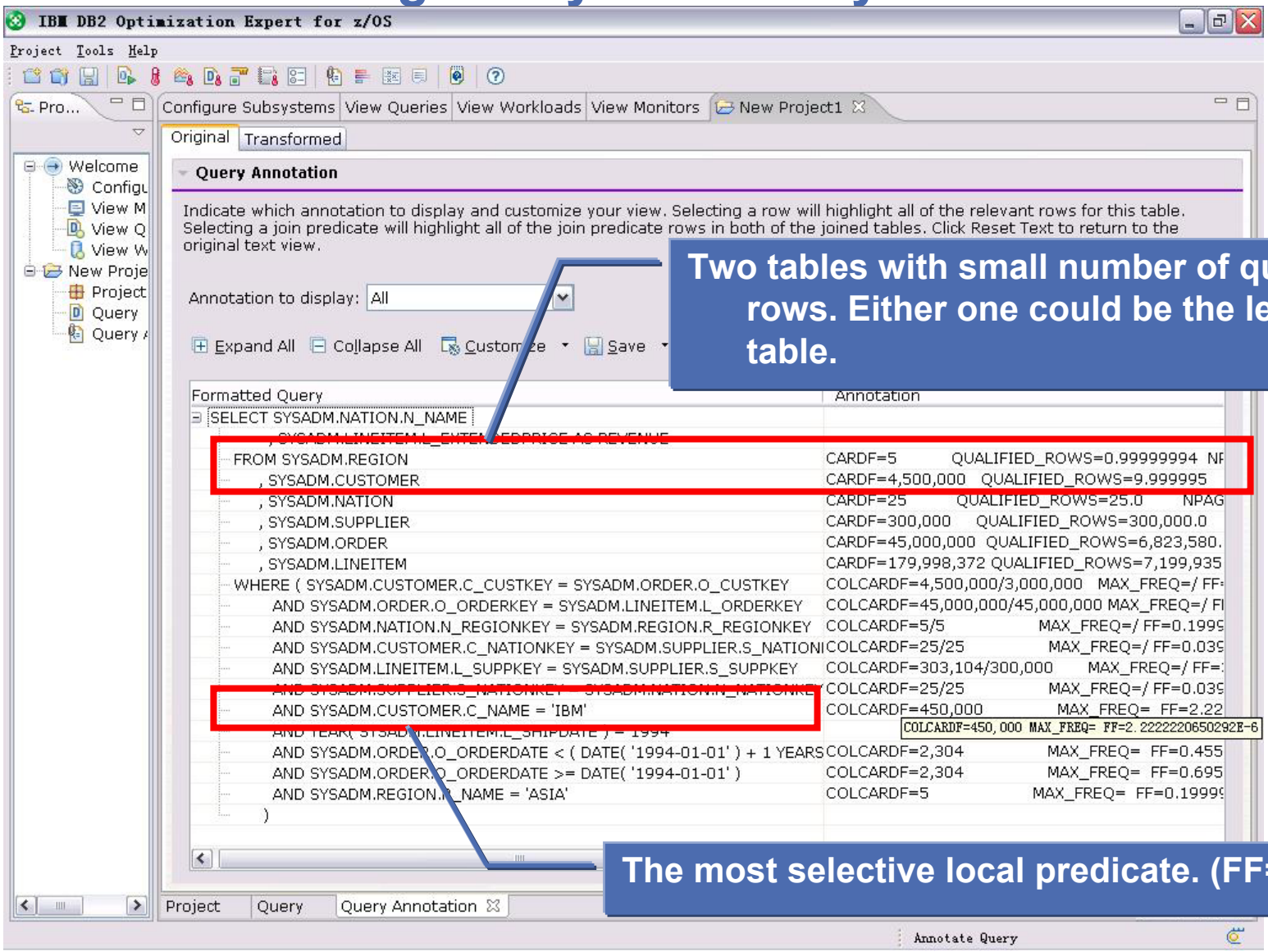


Original and transformed Query

Formatted, reorganized query text

Annotations (catalog stats, cost estimation)

Understanding Query with Query Annotation



Two tables with small number of qualified rows. Either one could be the leading table.

The most selective local predicate. (FF=2.22E-6)

Query Report – Table Report

	Legend for column names that have been truncated
PARTS	-- Number of partitions in tablespace
QUALROWS	-- Optimizer's estimate for how many rows qualify if this table were the outer table
CLU	-- Is the index the clustering index?
UR	-- Unique rule
CR	-- Clusterratiof

TABLE_SPACE	NACTIVEF	PARTS	SEGSIZE	PG_SIZE						
DB4REGN.TSREGION	900.0	5	0	4						
		TABLE	CARDF	NPAGESF	TABNO	QUALROWS				
		SYSADM.REGION	5	0	6	0.99999994				
		INDEX	CLU	UR	NLEAF	NLEVEL	CR	KEYCOLNAME	COLCARDF	MCARDF
		SYSADM.PXR@RKNM	Y	U	5	2	1.0	R_REGIONKEY	5	5
								R_NAME	5	5
		COLUMN_GROUP	MCARDF							
TABLE_SPACE	NACTIVEF	PARTS	SEGSIZE	PG_SIZE						
DB4NATN.TSNATION	4500.0	25	0	4						
		TABLE	CARDF	NPAGESF	TABNO	QUALROWS				
		SYSADM.NATION	25	0	5	25.0				
		INDEX								
		SYSADM.PXN@NKNMRI								
		COLUMN_GROUP								
TABLE_SPACE	NACTIVEF	PARTS								
DB4CUST.TSCUST	192569.0	60	0	4						
		TABLE	CARDF	NPAGESF	TABNO	QUALROWS				
		SYSADM.CUSTOMER	4500000	0	1	9.999995				
		INDEX	CLU	UR	NLEAF	NLEVEL	CR	KEYCOLNAME	COLCARDF	MCARDF
		SYSADM.PXC@CKNKMS	Y	U	29040	3	1.0	C_CUSTKEY	4500000	4500000
								C_NATIONKEY	25	-1
								C_MKTSEGMENT	5	4500000
		SYSADM.UXC@NKCK	N	U	17858	3	0.999	C_NATIONKEY	25	25
								C_CUSTKEY	4500000	4500000
		COLUMN_GROUP	MCARDF							

Two indexes on the table 'CUSTOMER', but neither contains the column key 'C_NAME'.

Query Report – Predicate Report

	Legend for column names that have been truncated
FF	--Filter factor
BT	--Whether this predicate is a boolean term predicate
S1	--Whether the predicate is a stage 1 predicate
JN	--Whether this predicate is a join predicate
AJ	--Whether the predicat
PTC	--Whether the predicat hat belong to a query selection when the set of predicates
MARKE	--Whether this predica
PRED O	--The predicate numbe

Three EQUAL join predicates and one EQUAL local predicate on the table 'CUSTOMER'

One EQUAL join predicate and one EQUAL local predicate on the table 'REGION'

TABLE	TABNO	COLNAME	COLNO	COLCARDF	HIGH2KEY	LOW2KEY	MAX_FREQ	TYPE	OTH_TABLE	OTH_TABNO	OTH_COL
SYSADM.CUSTOMER	1	C_CUSTKEY	1	4500000	8044AA1F40404040	8000000240404040		EQUAL	SYSADM.ORDER	2	O_CUSTK
		C_NATIONKEY	4	25	8000001740404040	8000000140404040		EQUAL	SYSADM.SUPPLIER	4	S_NATION
								EQUAL	SYSADM.NATION	5	N_NATION
		C_NAME	2	450000	Customer	Customer		EQUAL		0	VALU
SYSADM.ORDER	2	O_CUSTKEY	2	3000000	8044AA1E40404040	8000000240404040		EQUAL	SYSADM.CUSTOMER	1	C_CUSTK
		O_ORDERKEY	1	45000000	8ABA94E740404040	8000000240404040		EQUAL	SYSADM.LINEITEM	3	L_ORDER
		O_ORDERDATE	5	2304	1998-08-02	1992-01-01		RANGE		0	VALU
								RANGE		0	VALU
SYSADM.LINEITEM	3	L_ORDERKEY	1	45000000	8ABA94E740404040	8000000240404040		EQUAL	SYSADM.ORDER	2	O_ORDER
									PPLIER	4	S_SUPPK
SYSADM UPPLIER	4								EITEM	3	L_SUPPK
									TOMER	1	C_NATION
									ATION	5	N_NATION
SY ADM.NATION	5								PPLIER	4	S_NATION
									TOMER	1	C_NATION
		N_REGIONKEY	3	5	8000000440404040	1077952576		EQUAL	SYSADM.REGION	6	R_REGION
SYSADM.REGION	6	R_REGIONKEY	1	5	8000000340404040	8000000140404040		EQUAL	SYSADM.NATION	5	N_REGION
		R_NAME	2	5	MIDDLE E	AMERICA		EQUAL		0	VALU

Query Report – Index Report

TABLE	CORR_NAME				
SYSADM.REGION					
	INDEX	INDEX_ONLY	ONE_FETCH	EQUAL_UNIQUE	GB_OB_DISTINCT
	SYSADM.PXR@RKNM	Y	N	N	N
	KEYCOL	ORDER	COLUMN_CARD	MULTI_COL_CARD	PRED
	R_REGIONKEY	ASCENDING	5.0	5.0	SYSADM.NATION.N_REGIONKEY=SYSADM.REGION.R_REGIONKEY (FF:0.19999998807907104)
	R_NAME	ASCENDING	5.0	5.0	SYSADM.REGION.R_NAME='ASIA' (FF:0.19999998807907104)

TABLE	CORR_NAME				
SYSADM.NATION					
	INDEX				
	SYSADM.PXN@NKNMRK				
	KEYCOL	ORDER	COLUMN_CARD	MULTI_COL_CARD	PRED
	N_NATIONKEY	ASCENDING	25.0	25.0	SYSADM.SUPPLIER.S_NATIONKEY=SYSADM.NATION.N_NATIONKEY
					N.N_NATIONKE
	N_REGIONKEY	ASCENDING	5.0	25.0	SYSADM.NATION.N_REGIONKEY=SYSADM.REGION.R_REGIONKEY (FF:0.19999998807907104)

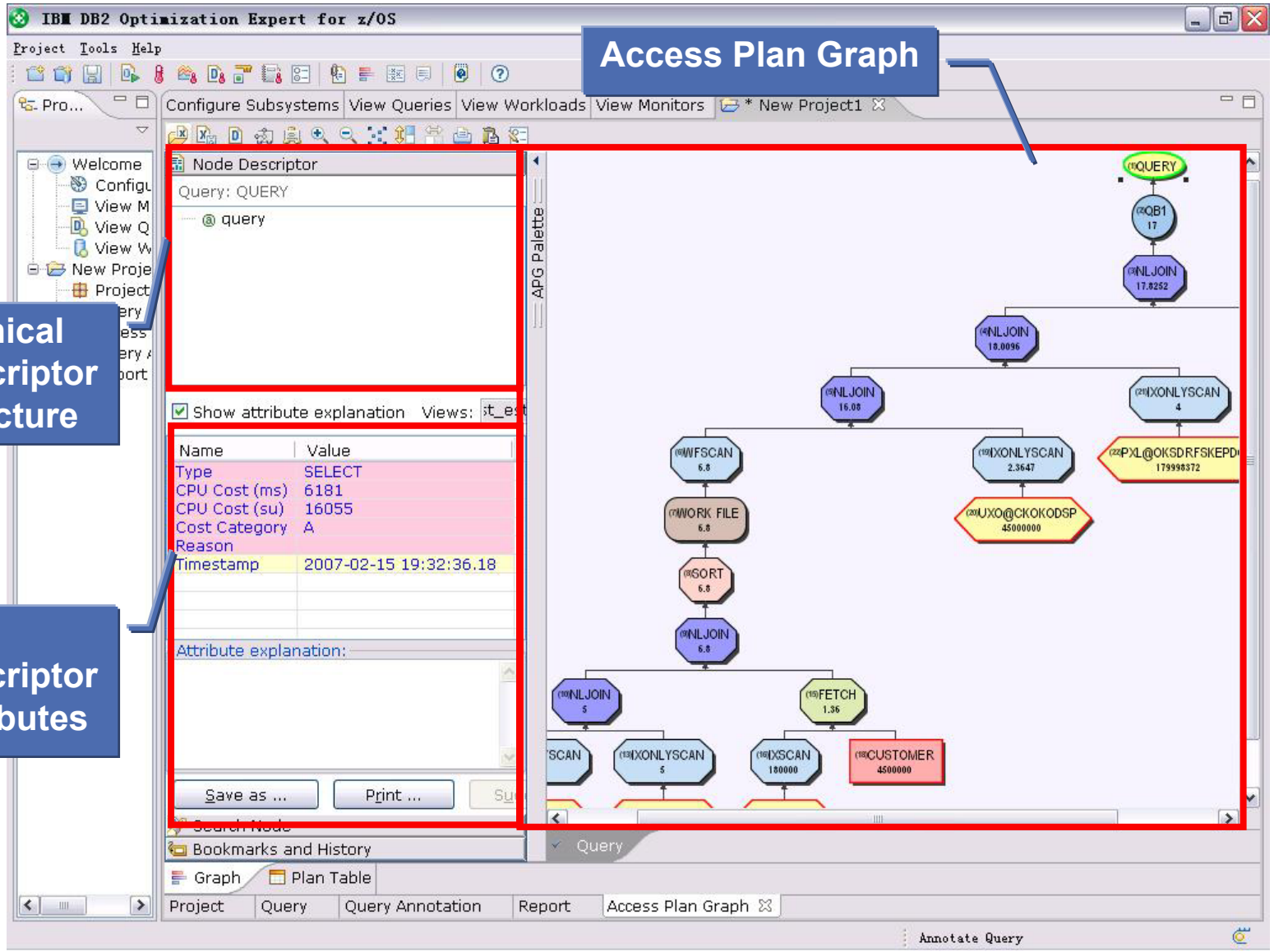
QB: PLAN:3

TABLE	CORR_NAME				
SYSADM.CUSTOMER					
	INDEX	INDEX_ONLY	ONE_FETCH	EQUAL_UNIQUE	GB_OB_DISTINCT
	SYSADM.PXC@CKNKMS	N	N	N	N
	KEYCOL	ORDER	COLUMN_CARD	MULTI_COL_CARD	PRED
	C_CUSTKEY	ASCENDING	4500000.0	4500000.0	SYSADM.CUSTOMER.C_CUSTKEY=SYSADM.ORDER.O_CUSTKEY (FF:2.2222218376555247E-7)
	C_NATIONKEY	ASCENDING	25.0	-1.0	SYSADM.CUSTOMER.C_NATIONKEY=SYSADM.SUPPLIER.S_NATIONKI (FF:0.03999999910593033)
					SYSADM.CUSTOMER.C_NATIONKEY=SYSADM.NATION.N_NATIONKE (FF:0.03999999910593033)
	C_MKTSEGMENT	ASCENDING	5.0	4500000.0	
	INDEX	INDEX_ONLY	ONE_FETCH	EQUAL_UNIQUE	GB_OB_DISTINCT
	SYSADM.UXC@NKCK	N	N	N	N
	KEYCOL	ORDER	COLUMN_CARD	MULTI_COL_CARD	PRED

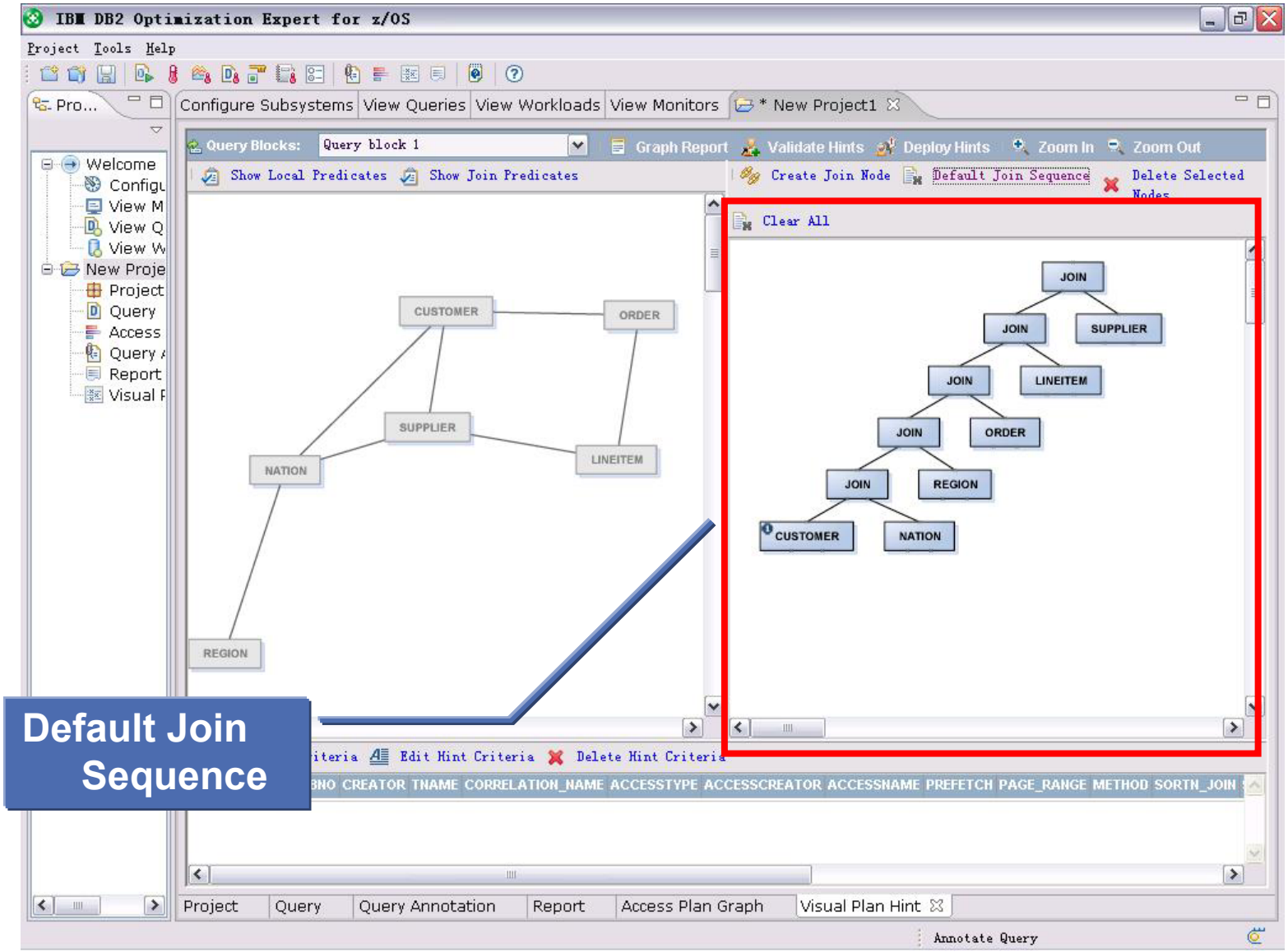
This index supports index-only scan on the table 'REGION'.

These two indexes supports index + fetch scan on the table 'CUSTOMER'.

Understanding the Access Path with Visual Explain

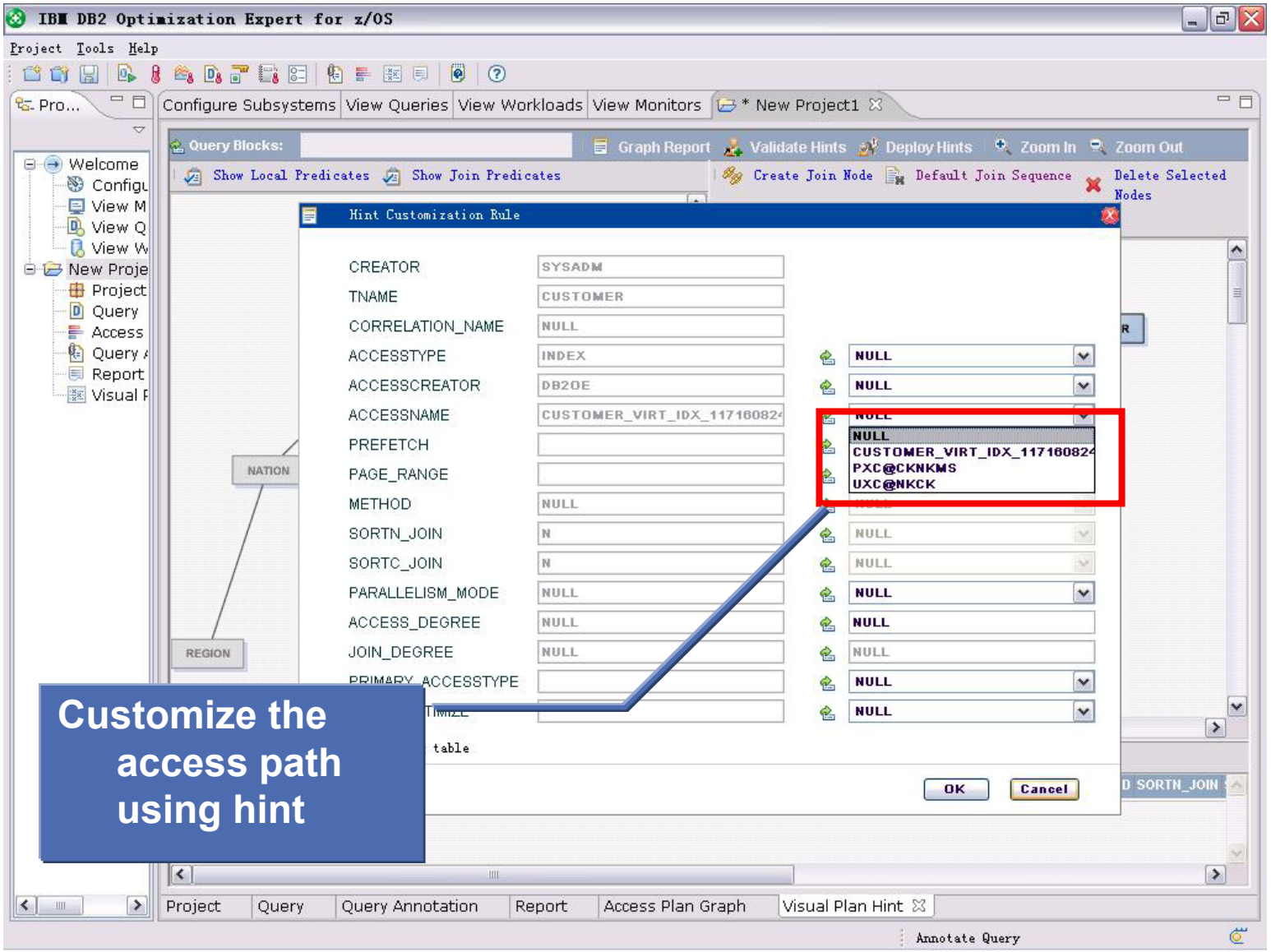


Implementing Emergency Solution with Visual Plan Hint

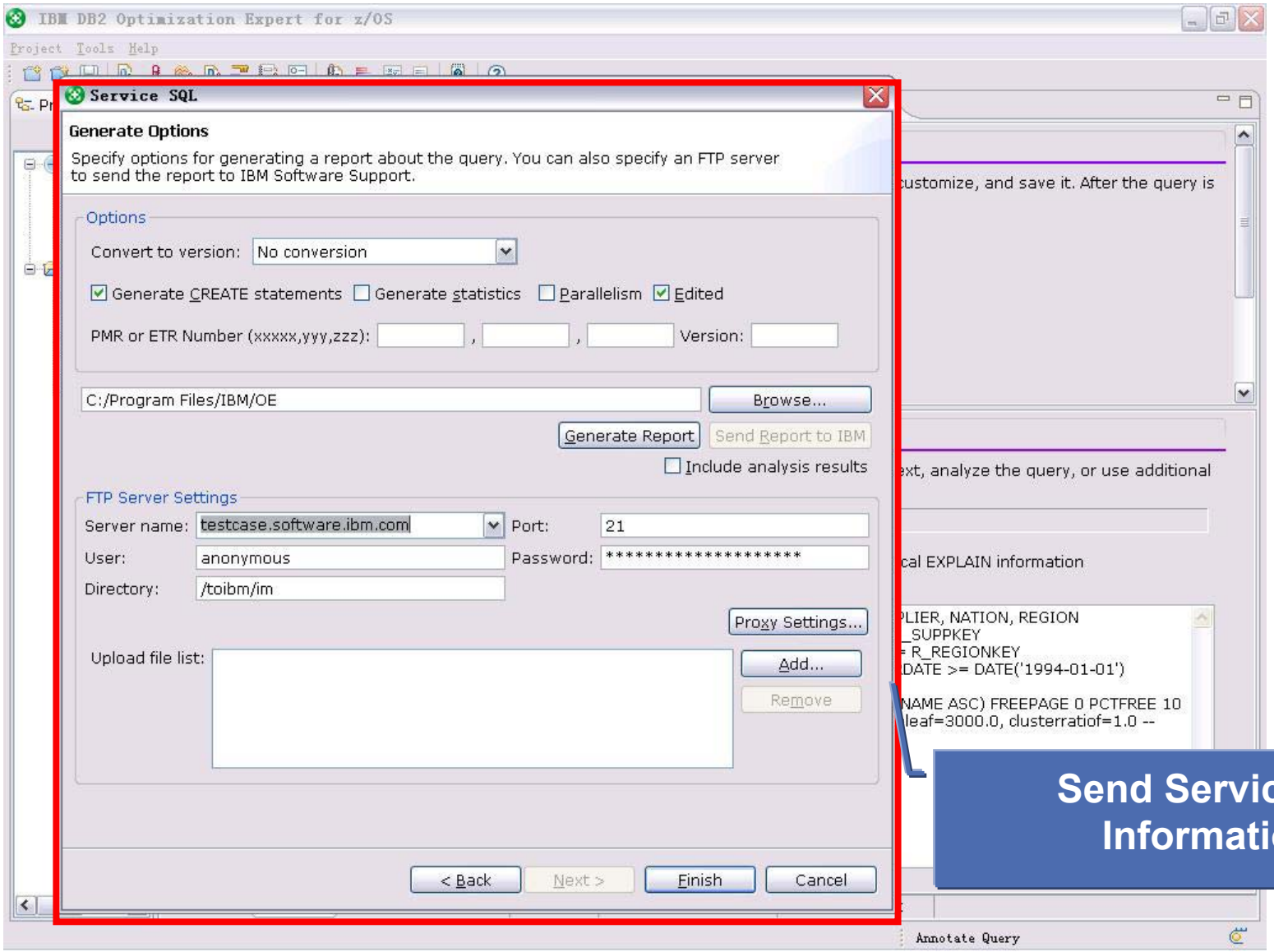


Default Join Sequence

Implementing Emergency Solution with Visual Plan Hint

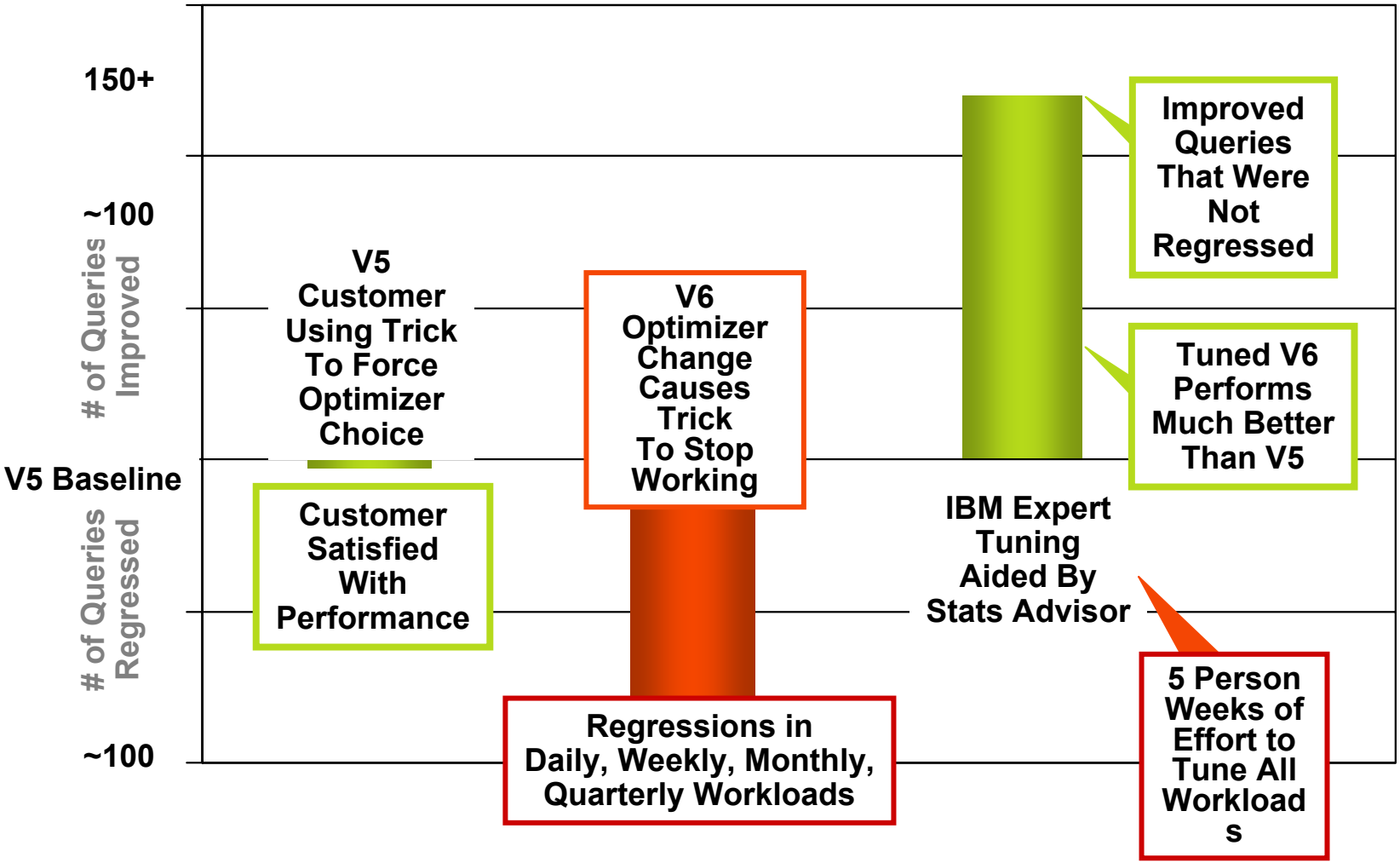


Automatic Collection of Relevant Information for PMR Analysis



Capturing Application Workloads

Scenario: Version to version migration failure

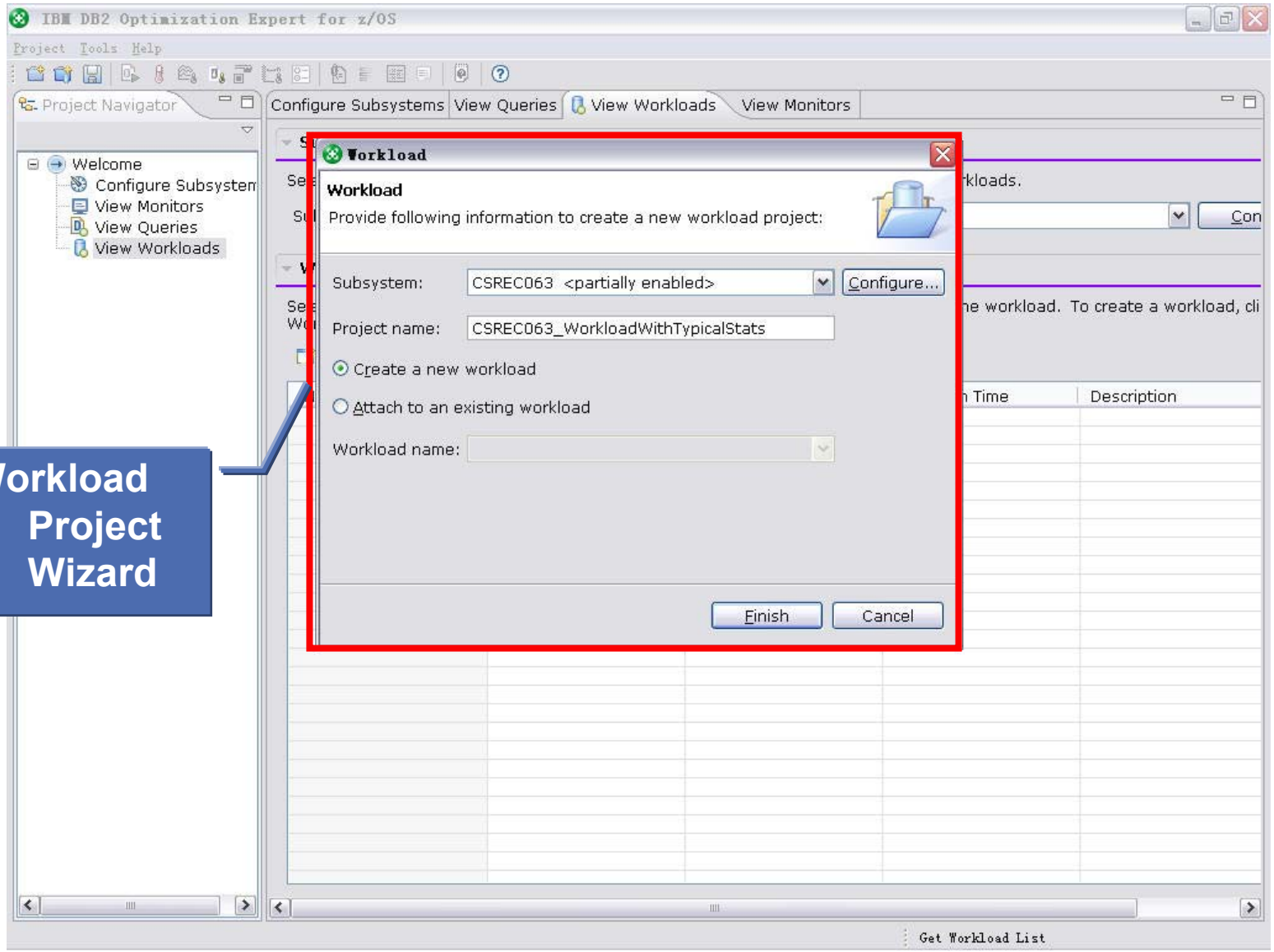


Scenario: Version to version migration failure

Post-mortem Analysis

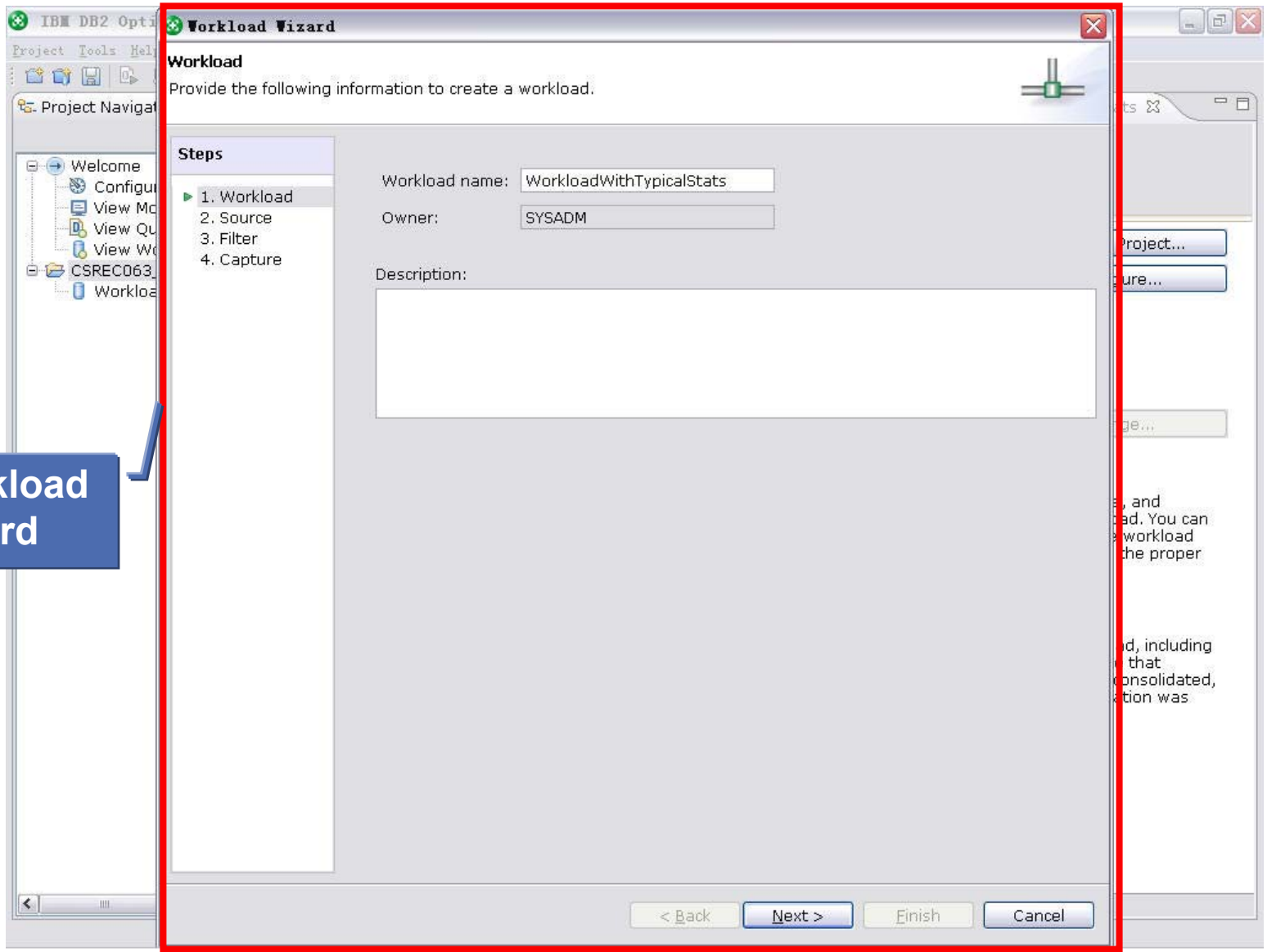
- Customer satisfied with V5 performance, but it was actually sub-optimal
- Complexity of problem tested the limits of expert analysis
  - **30 table joins nested behind several layers of views**
  - **Column names changed in views**
  - **Days to analyze a single query**
  - **100s of queries in multiple workloads**
- Customer costs
  - **6-8 month delay in migration**
  - **Person weeks of effort to resolve problems**
  - **Inability to take advantage of V6 enhancements**

Capturing Workload Queries



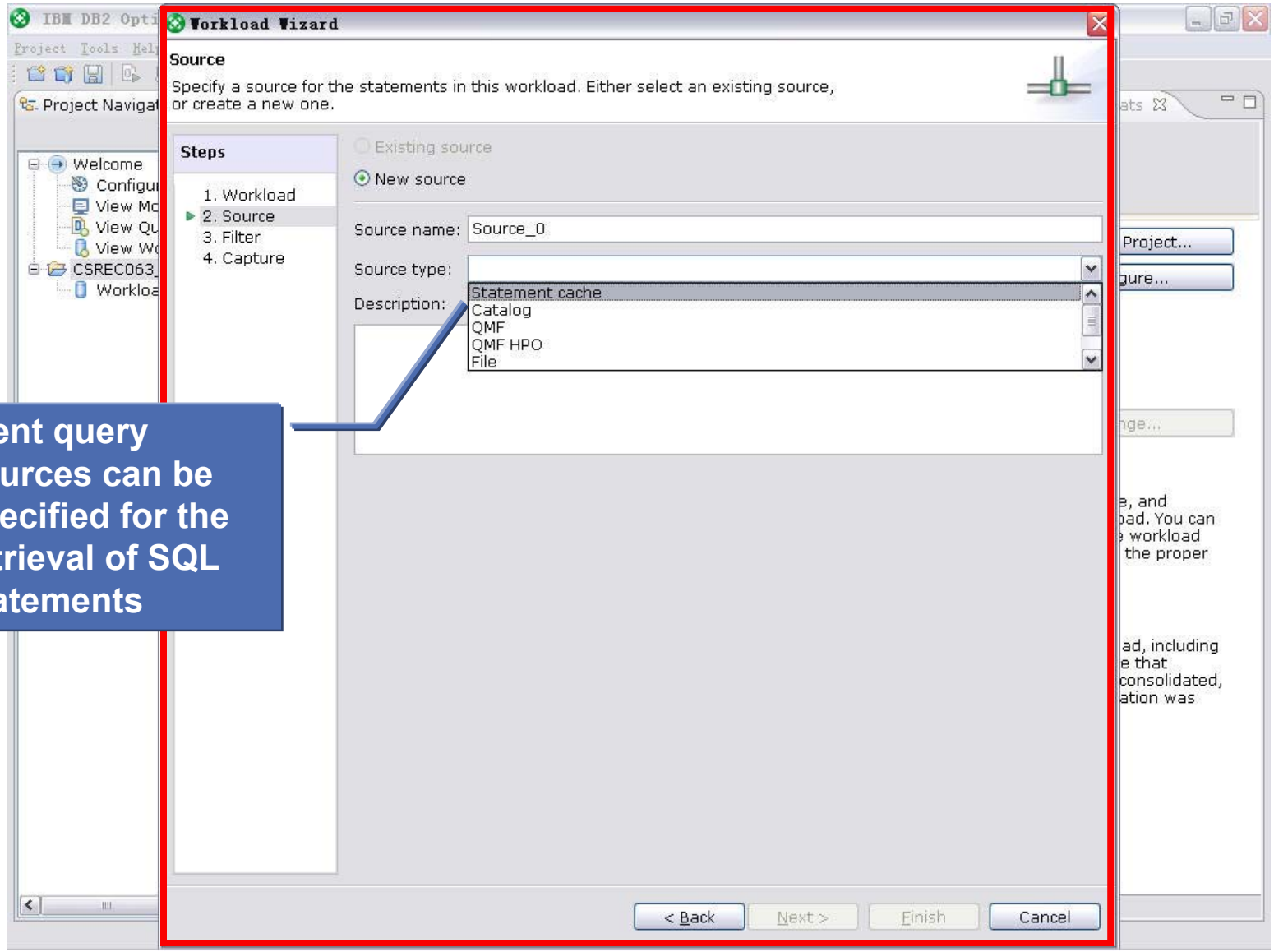
Workload Project Wizard

Capturing Workload Queries – General Info.

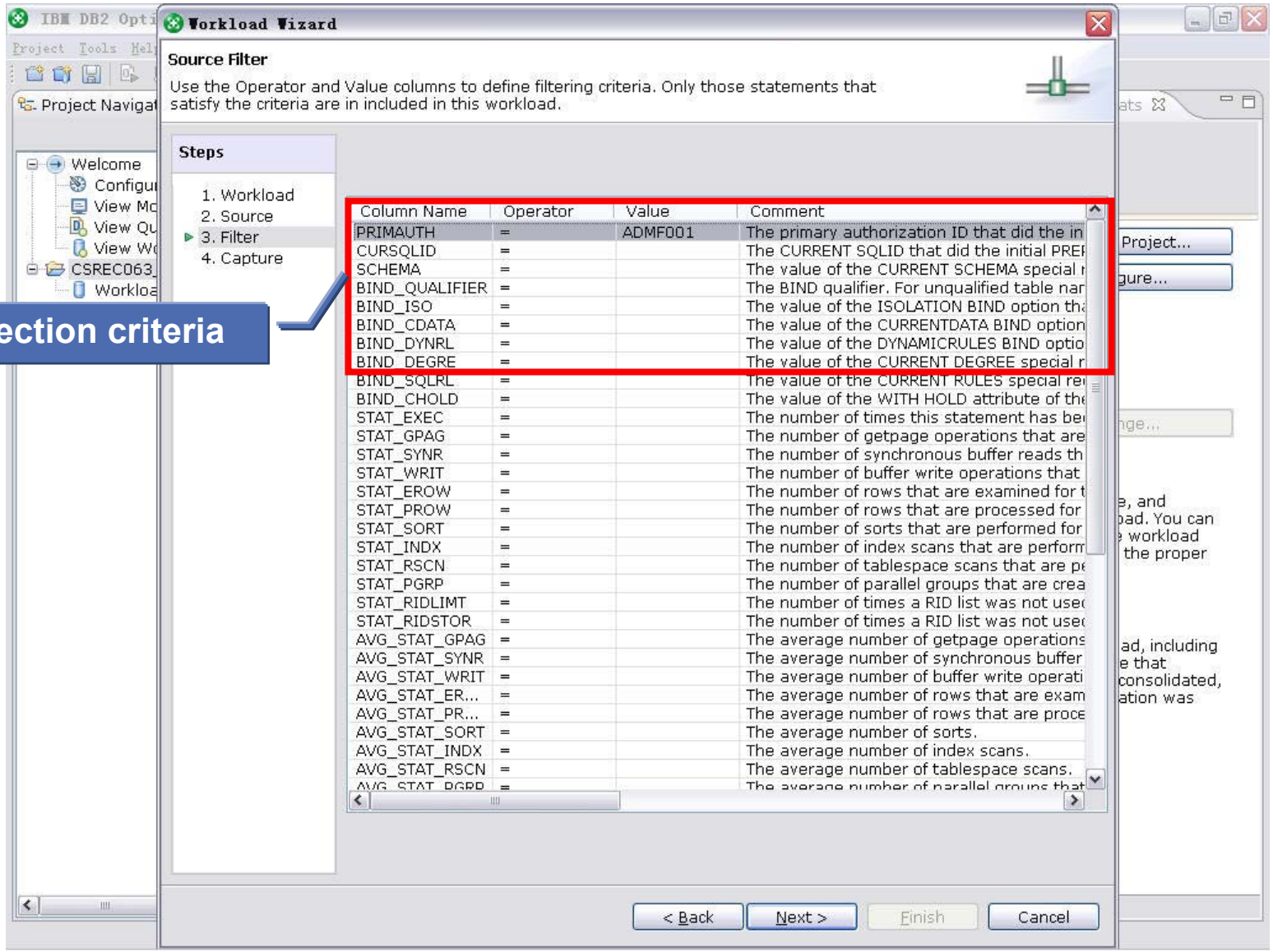


Workload Wizard

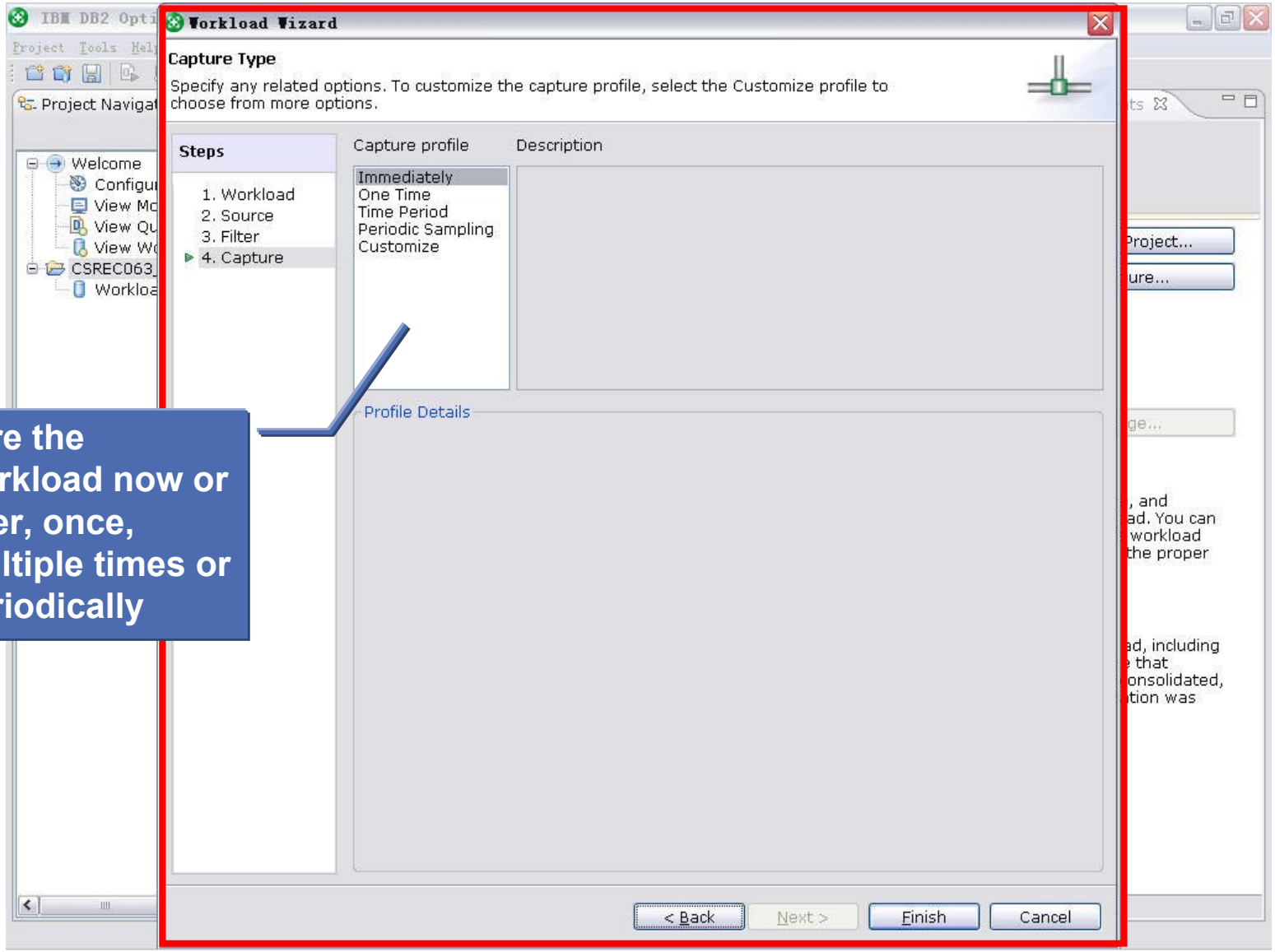
Capturing Workload Queries – Query Source



Capturing Workload Queries – Selection Criteria

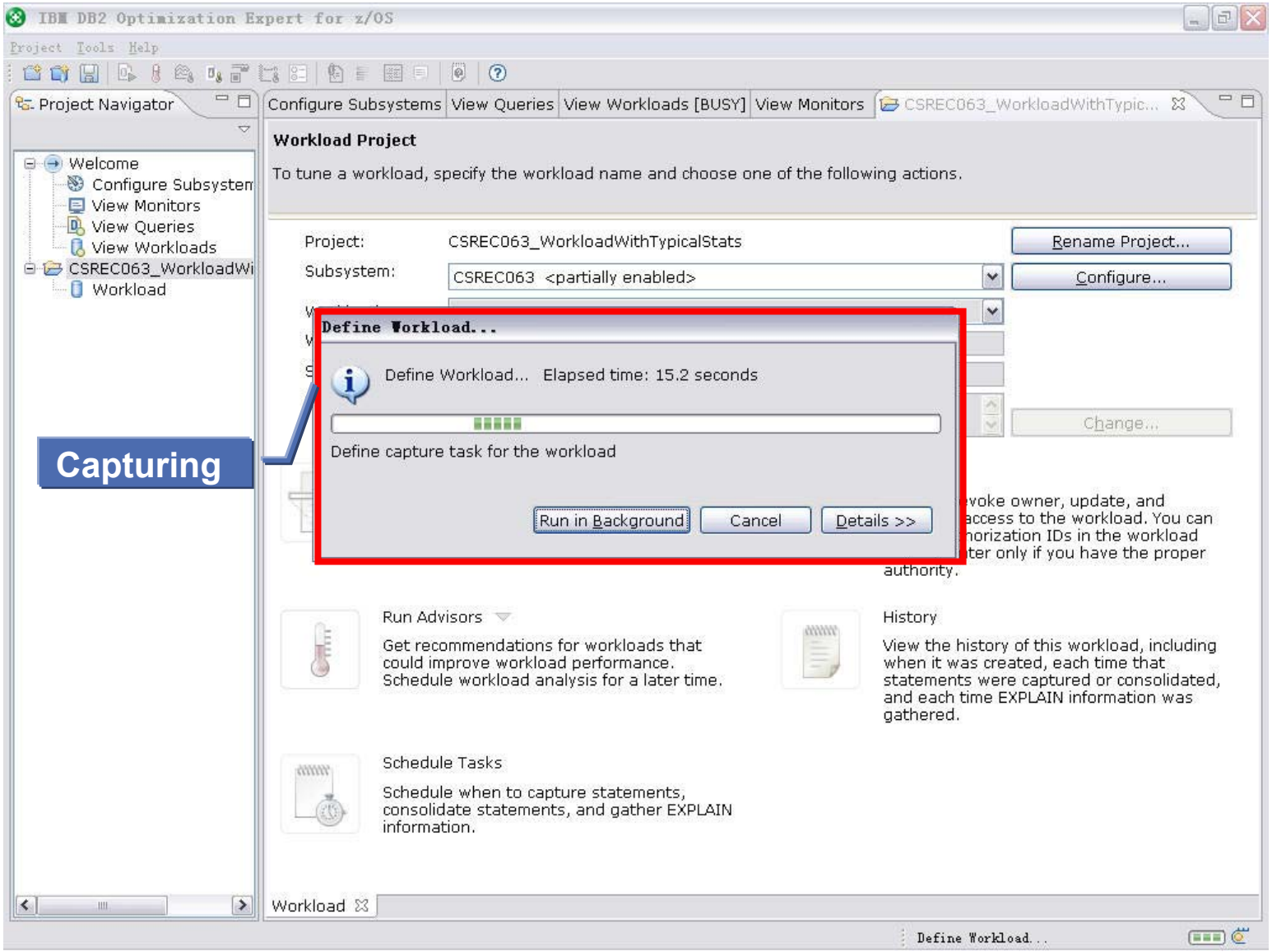


Capturing Workload Queries – When & How

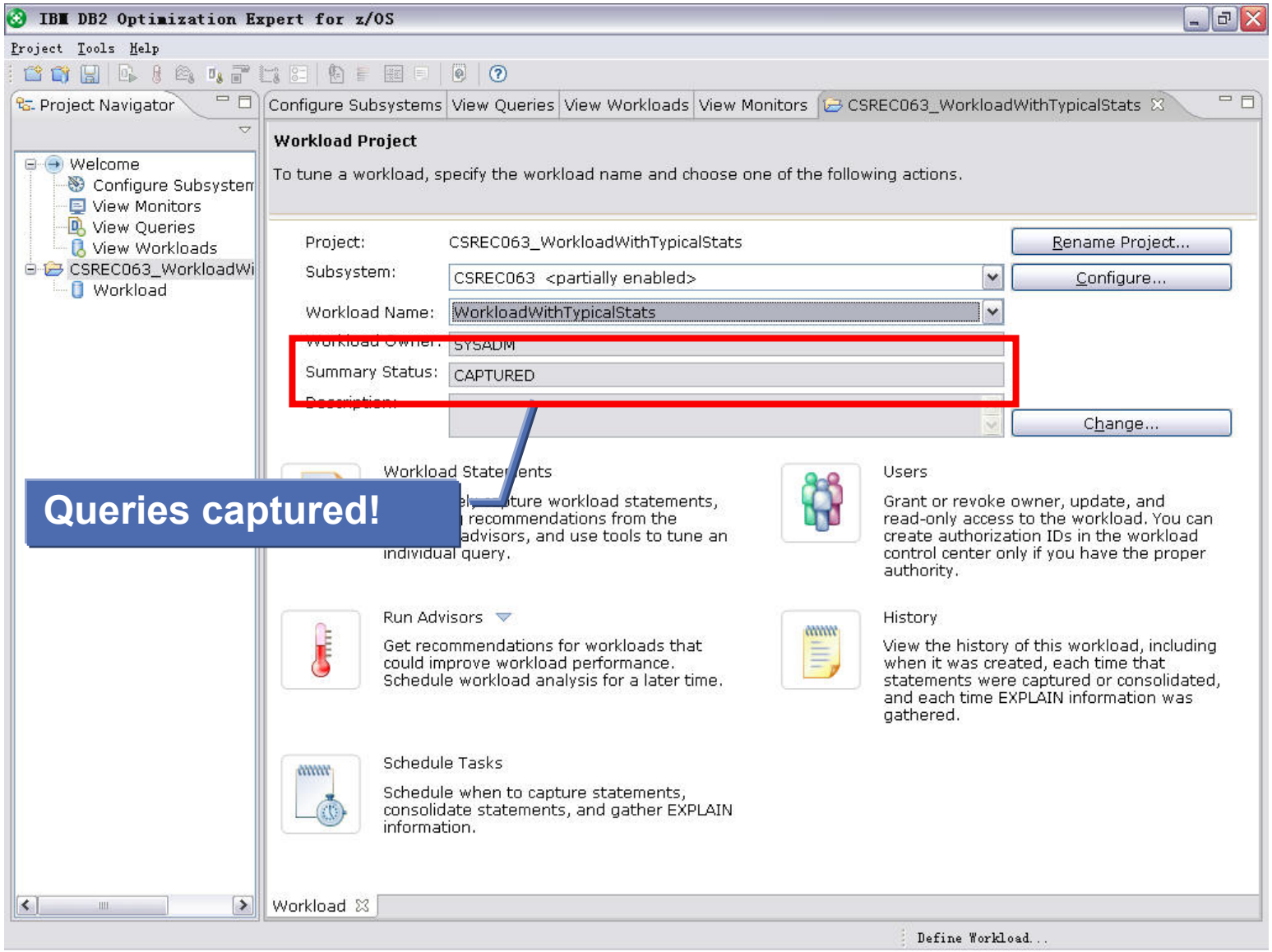


Capture the workload now or later, once, multiple times or periodically

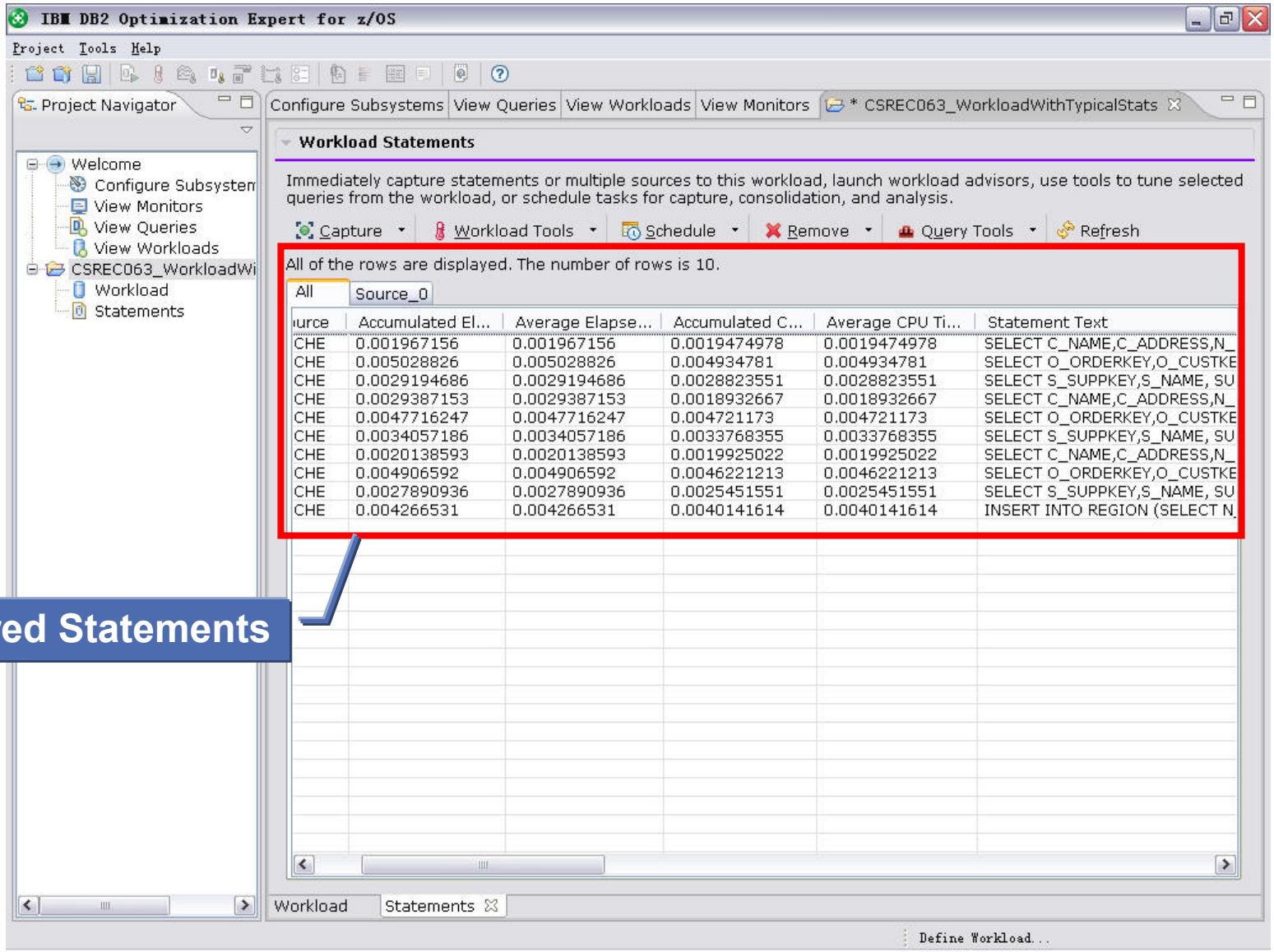
Capturing Workload Queries – Capturing



Capturing Workload Queries – Completion

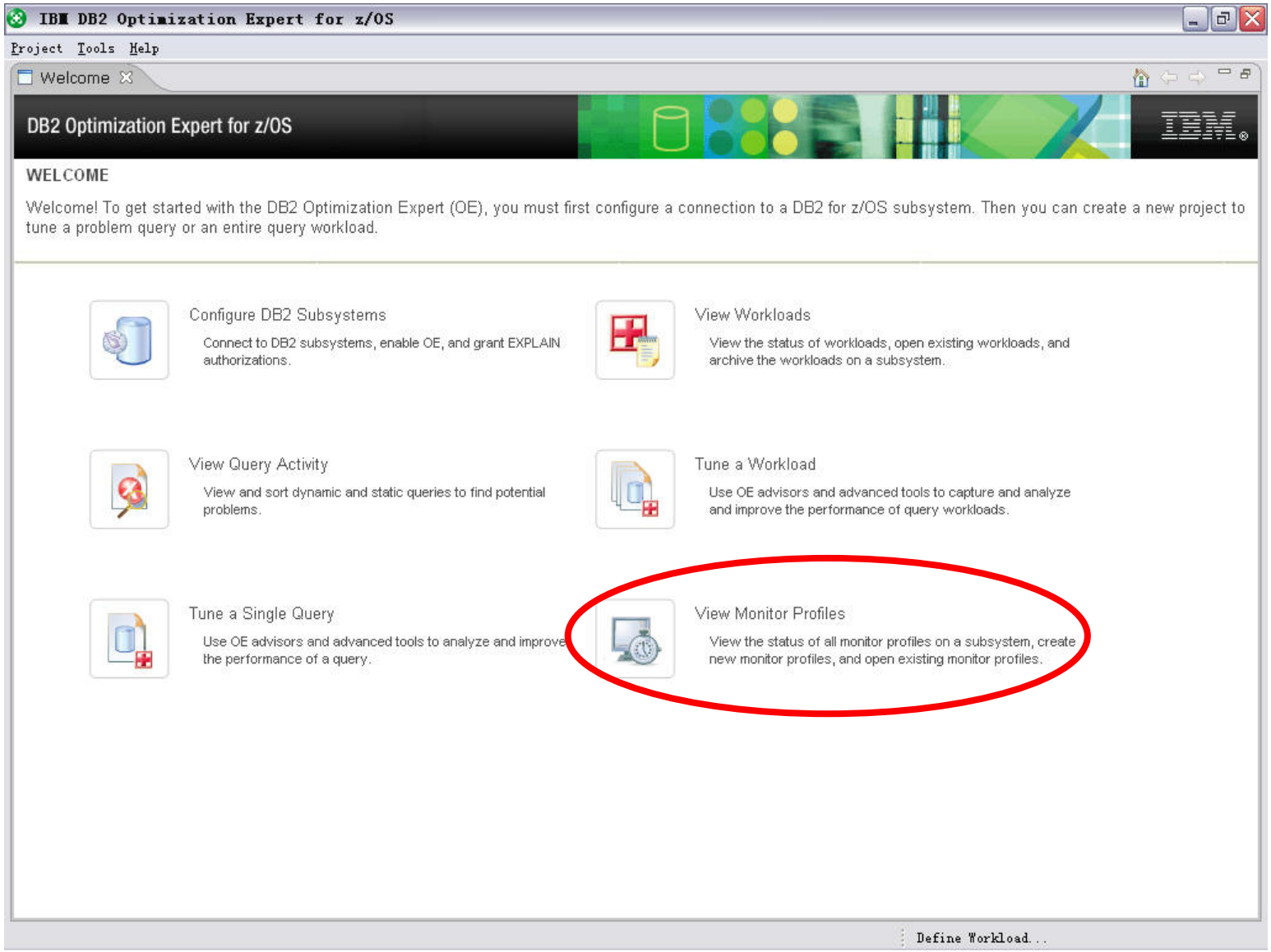


Capturing Workload Queries – Browsing

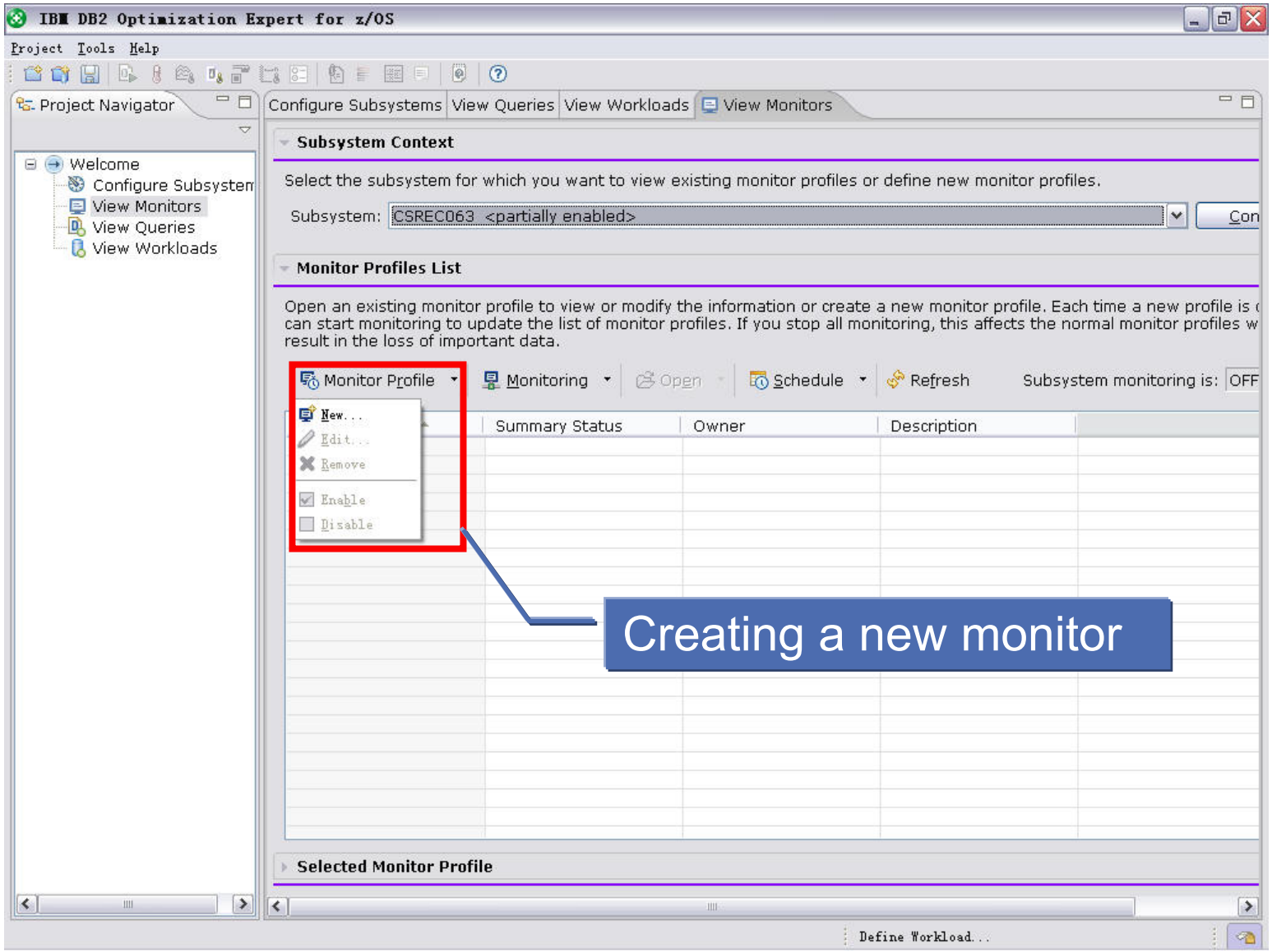


Captured Statements

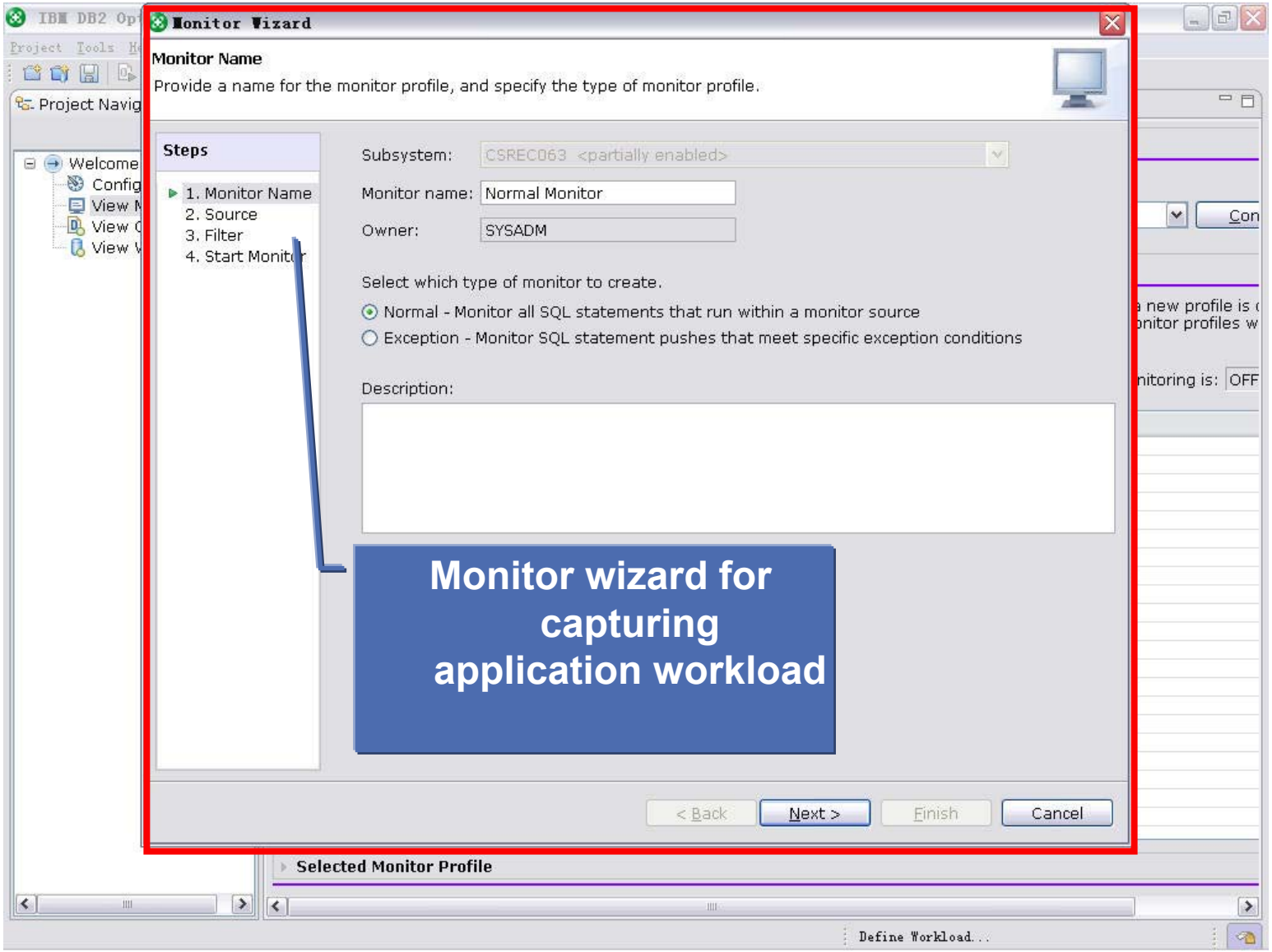
Capture Workload Queries with Monitor



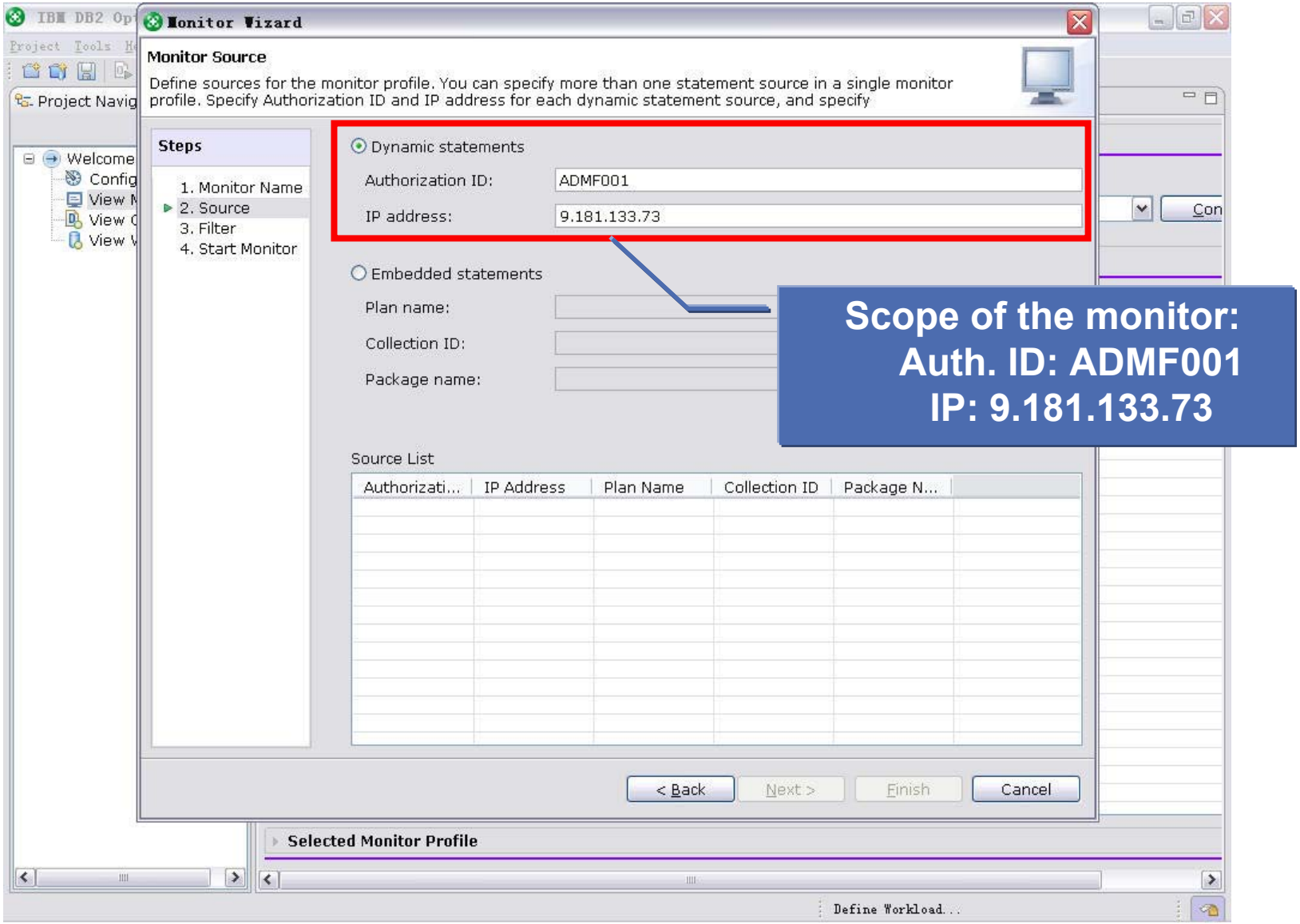
Capture Workload Queries with Monitor



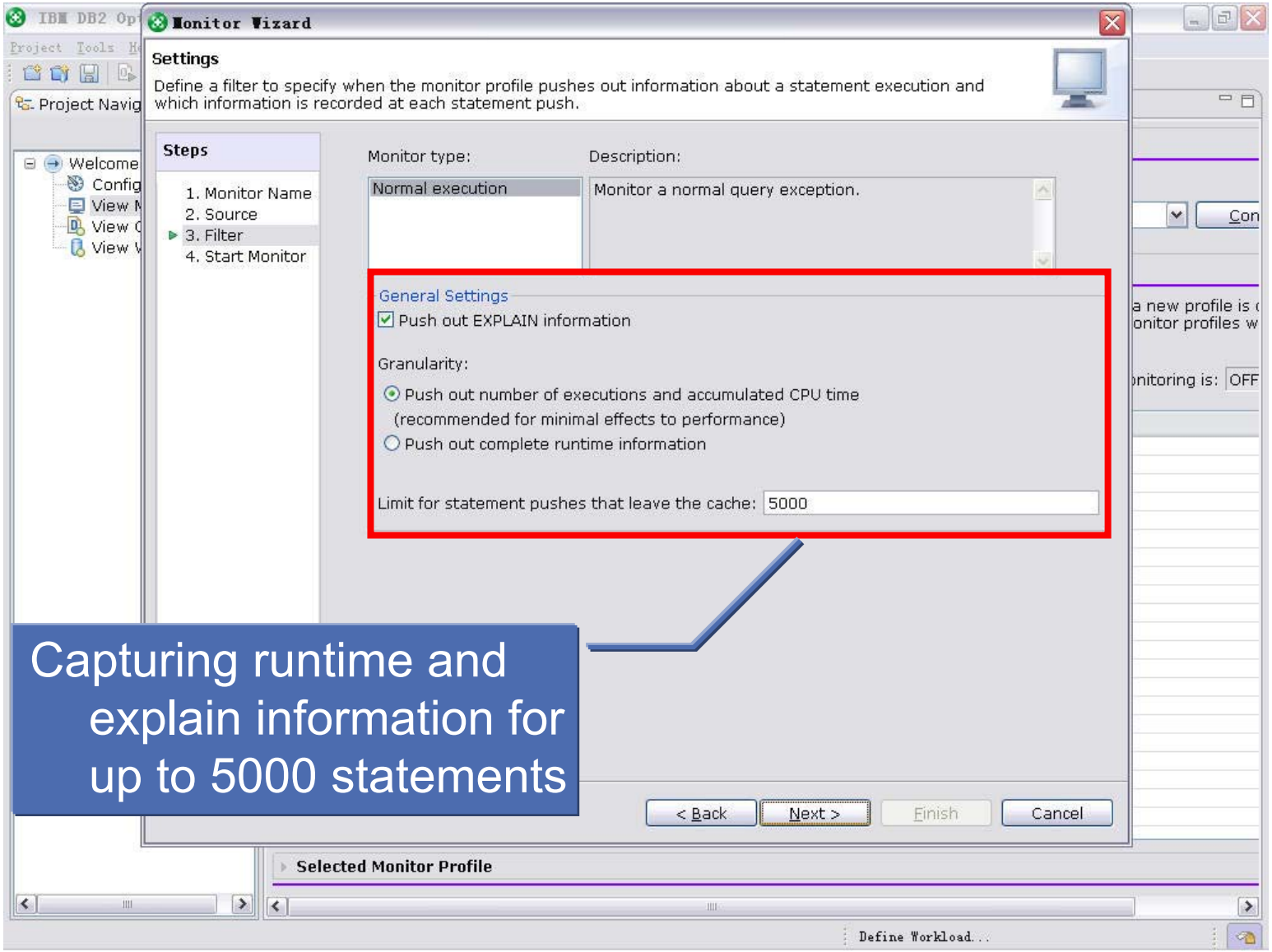
Capture Workload Queries with Monitor



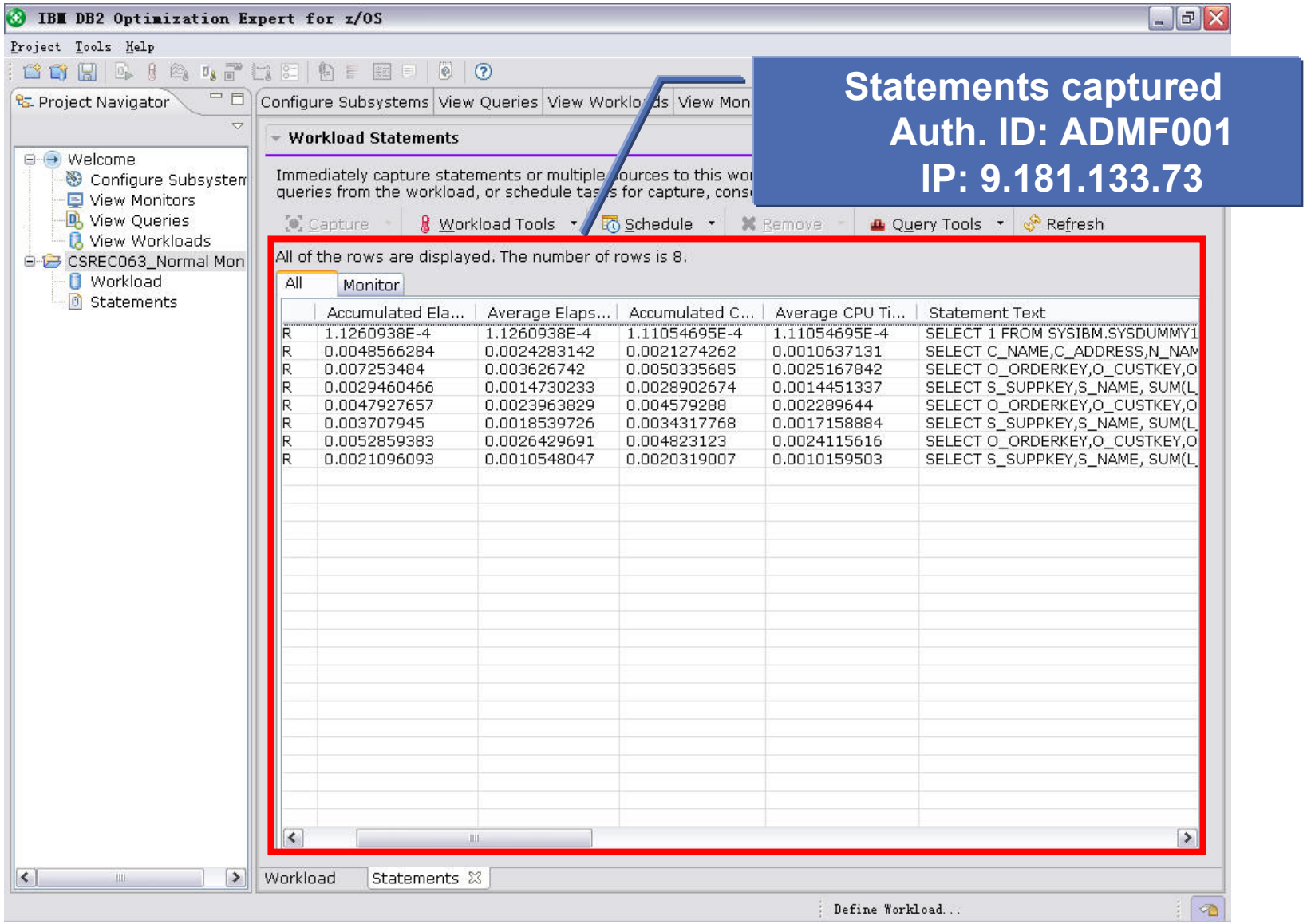
Capture Workload Queries with Monitor



Capture Workload Queries with Monitor



Capture Workload Queries with Monitor

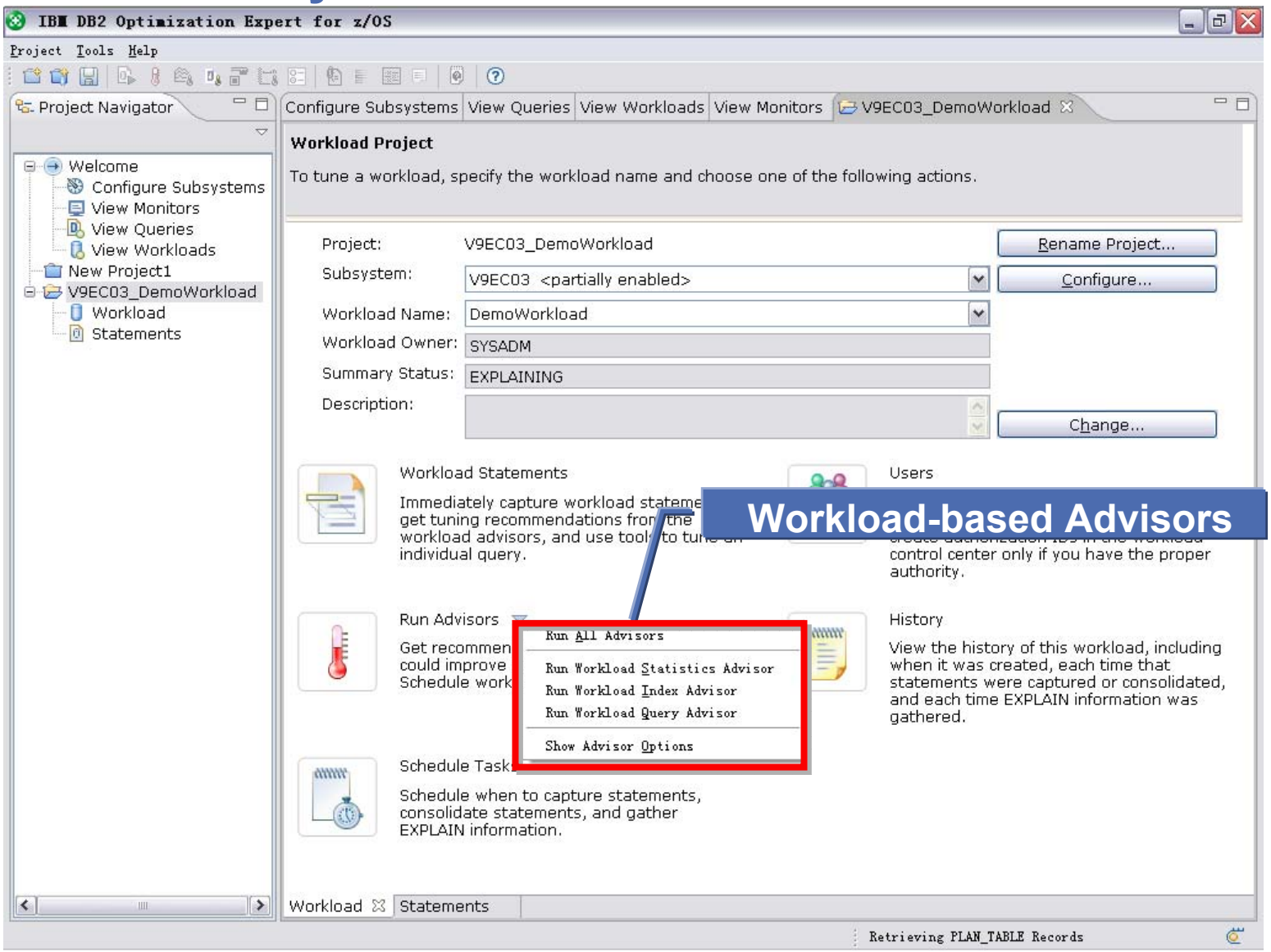


Performing Health Check for Application Workloads with Design Advisors

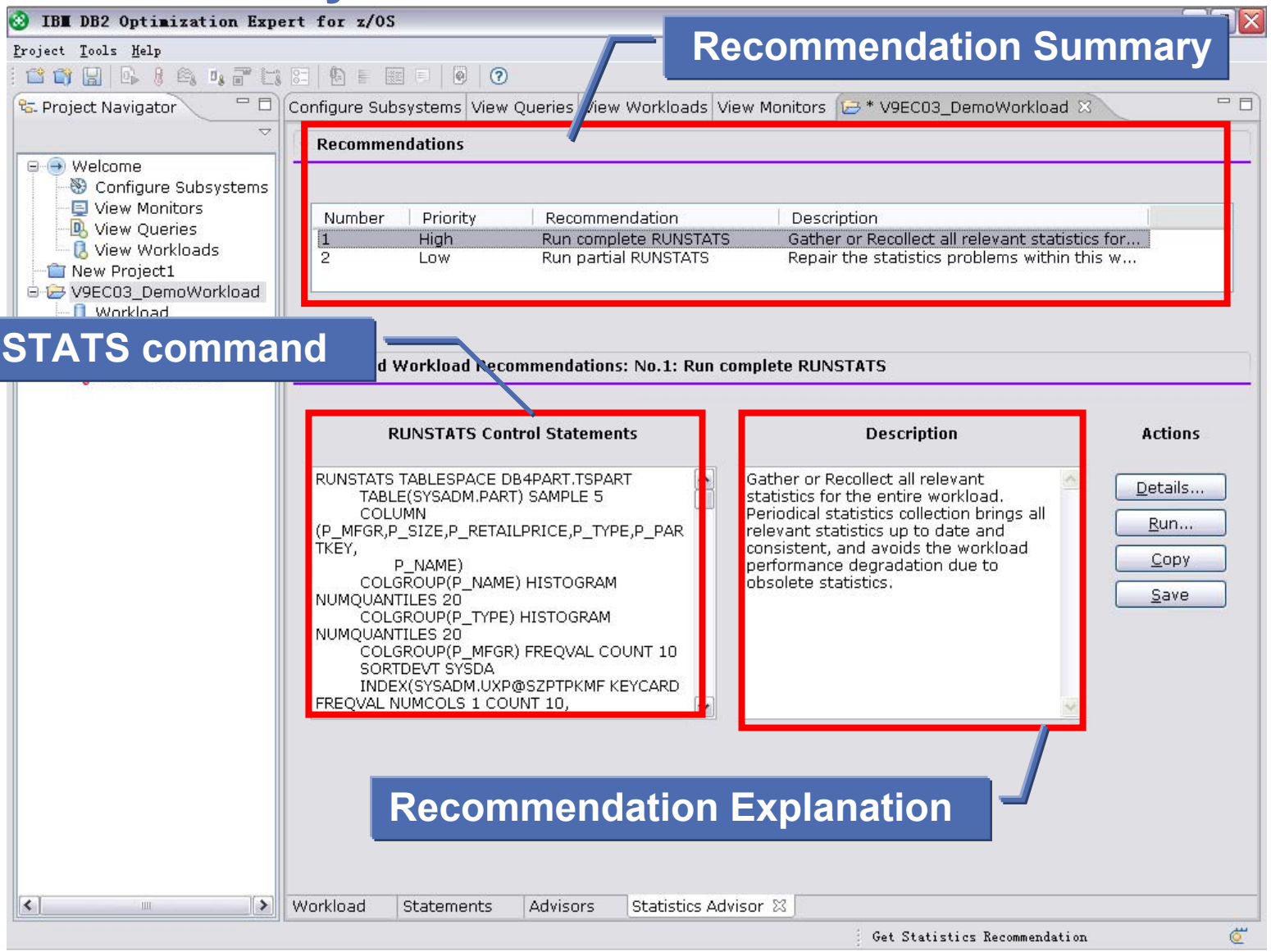
Design Advisors

- Stats Advisor – stats recommendation for access path selection
- Index Advisor – recommendation for index design
- Query Advisor – recommendation for query design

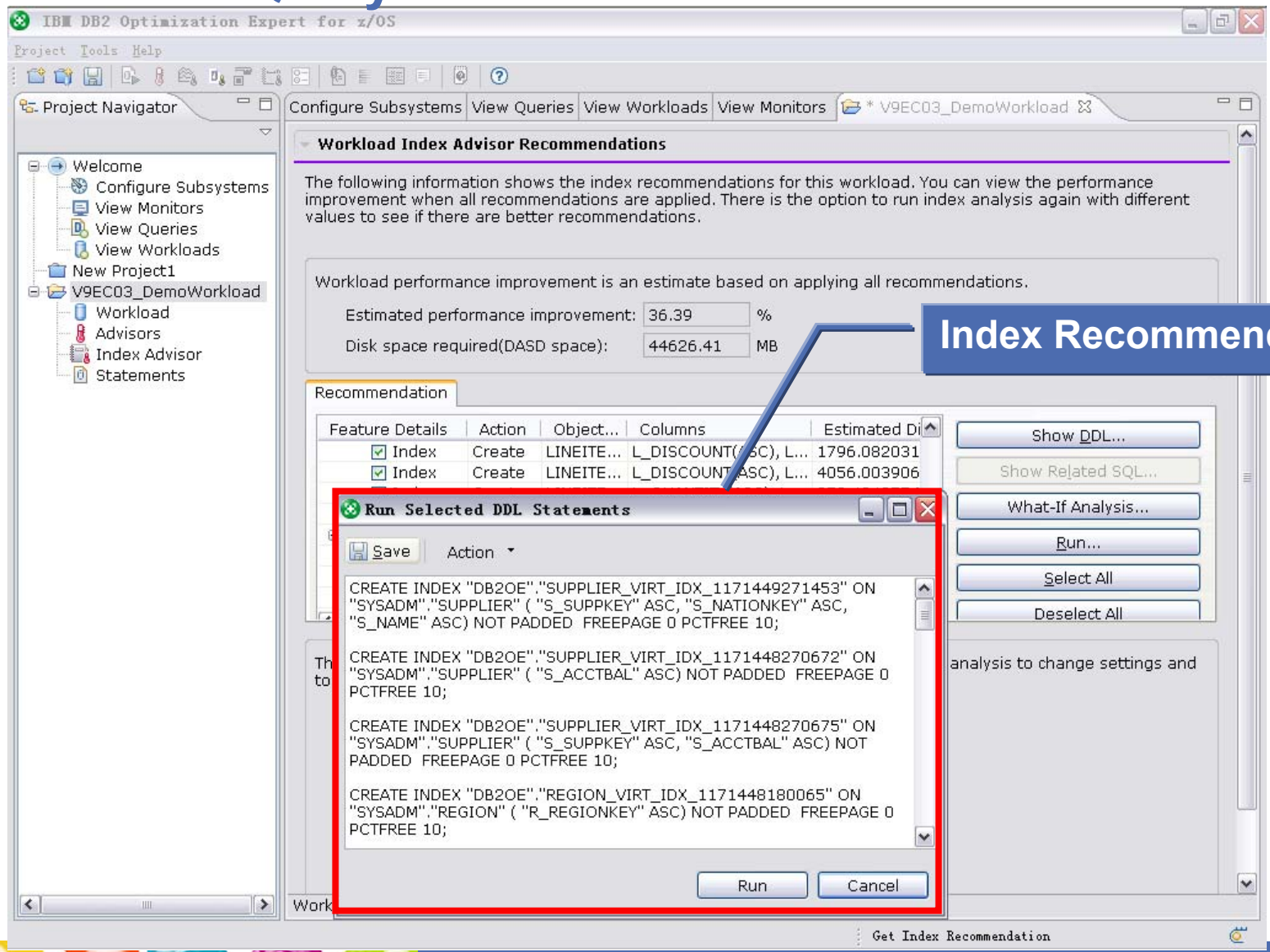
Tune Query Workload



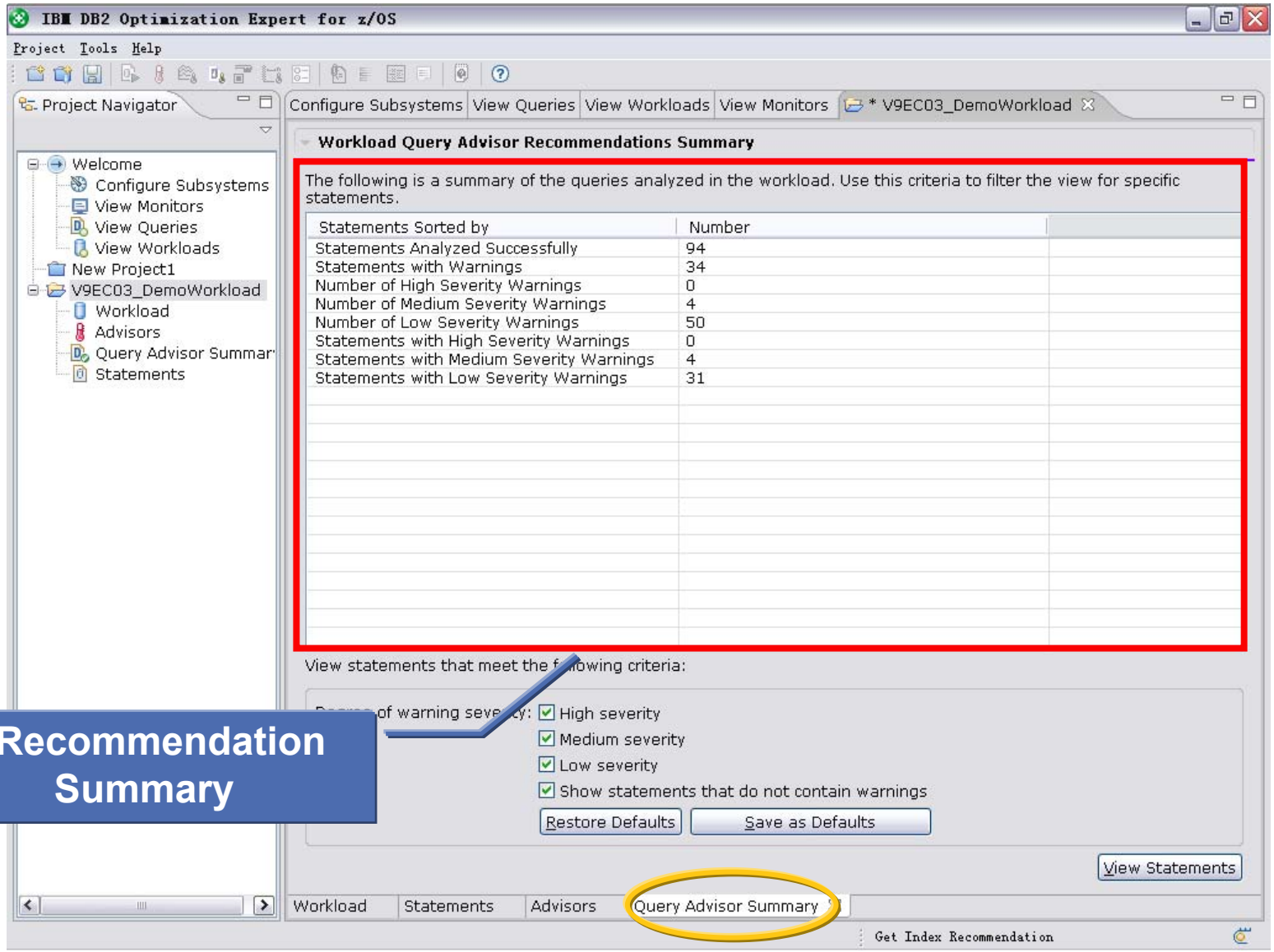
Tune Query Workload – Statistics Advisor



Tune Query Workload – Index Advisor

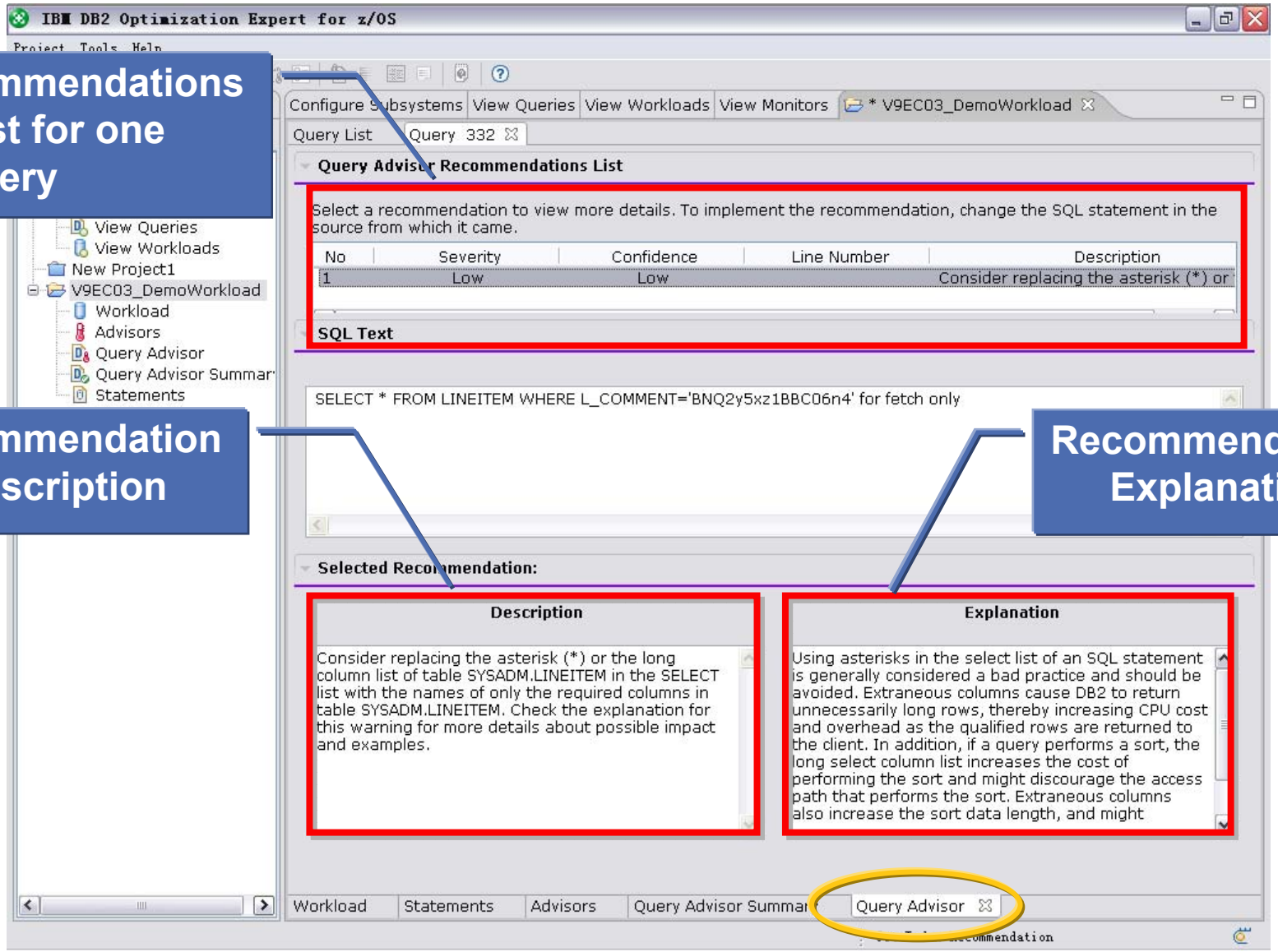


Tune Query Workload – Query Advisor



Recommendation Summary

Tune Query Workload – Query Advisor

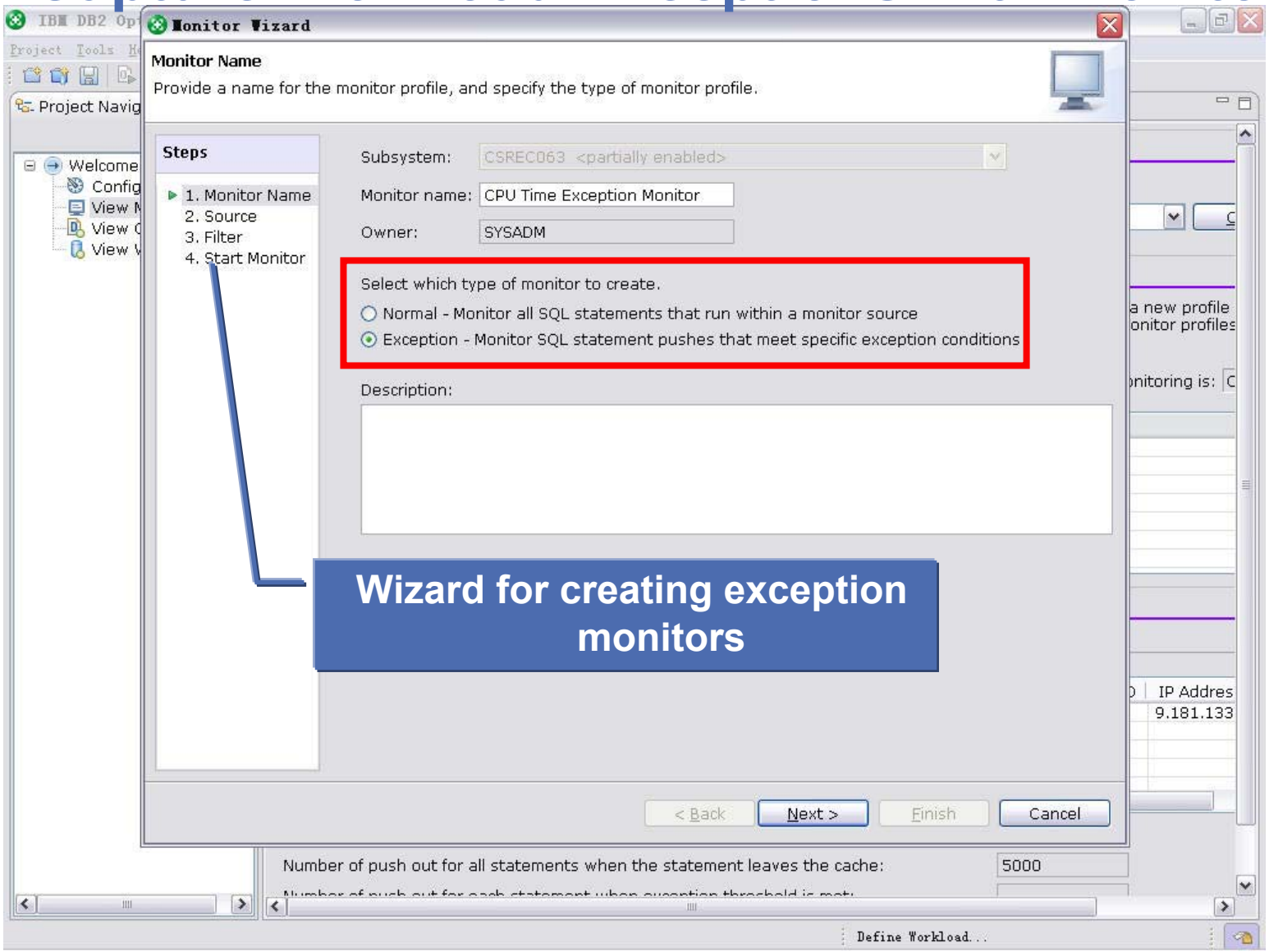


Monitoring Workload Exceptions

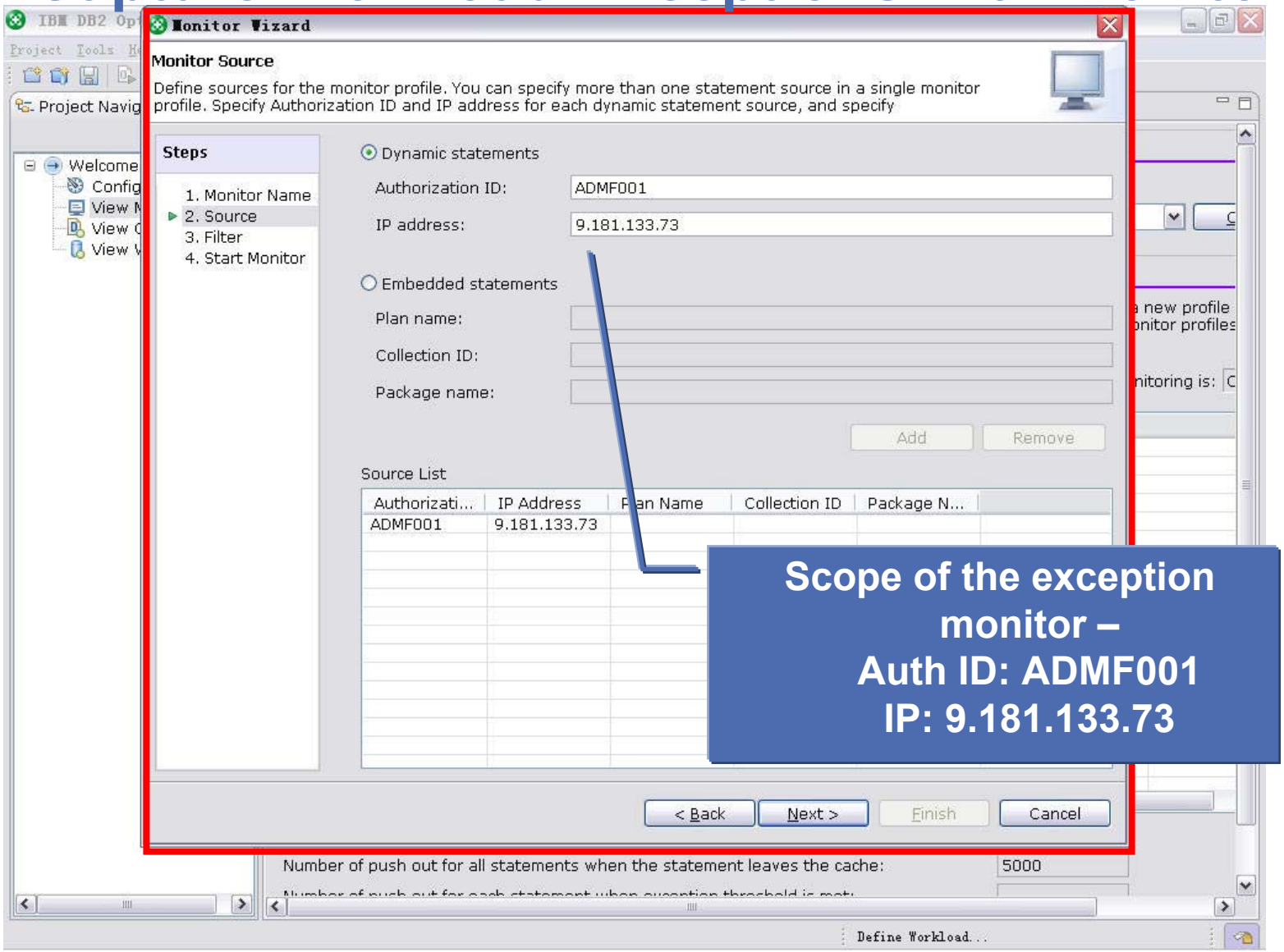
Defining Performance Exceptions

- Two kinds of exceptions
  - **CPU time exception**
  - **CPU spike exception**
- Notify performance exception as soon as it occurs

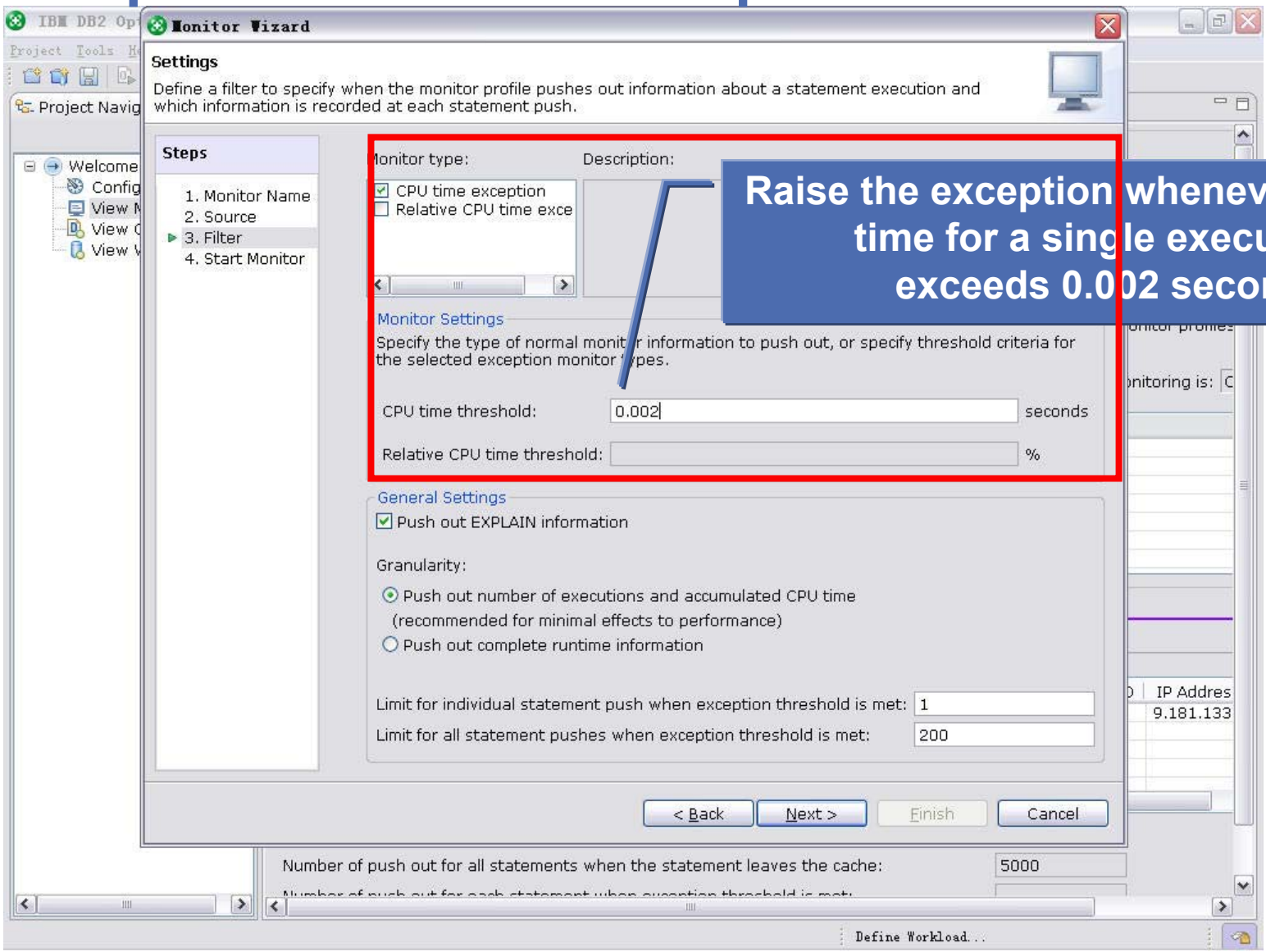
Capture Workload Exceptions with Monitor



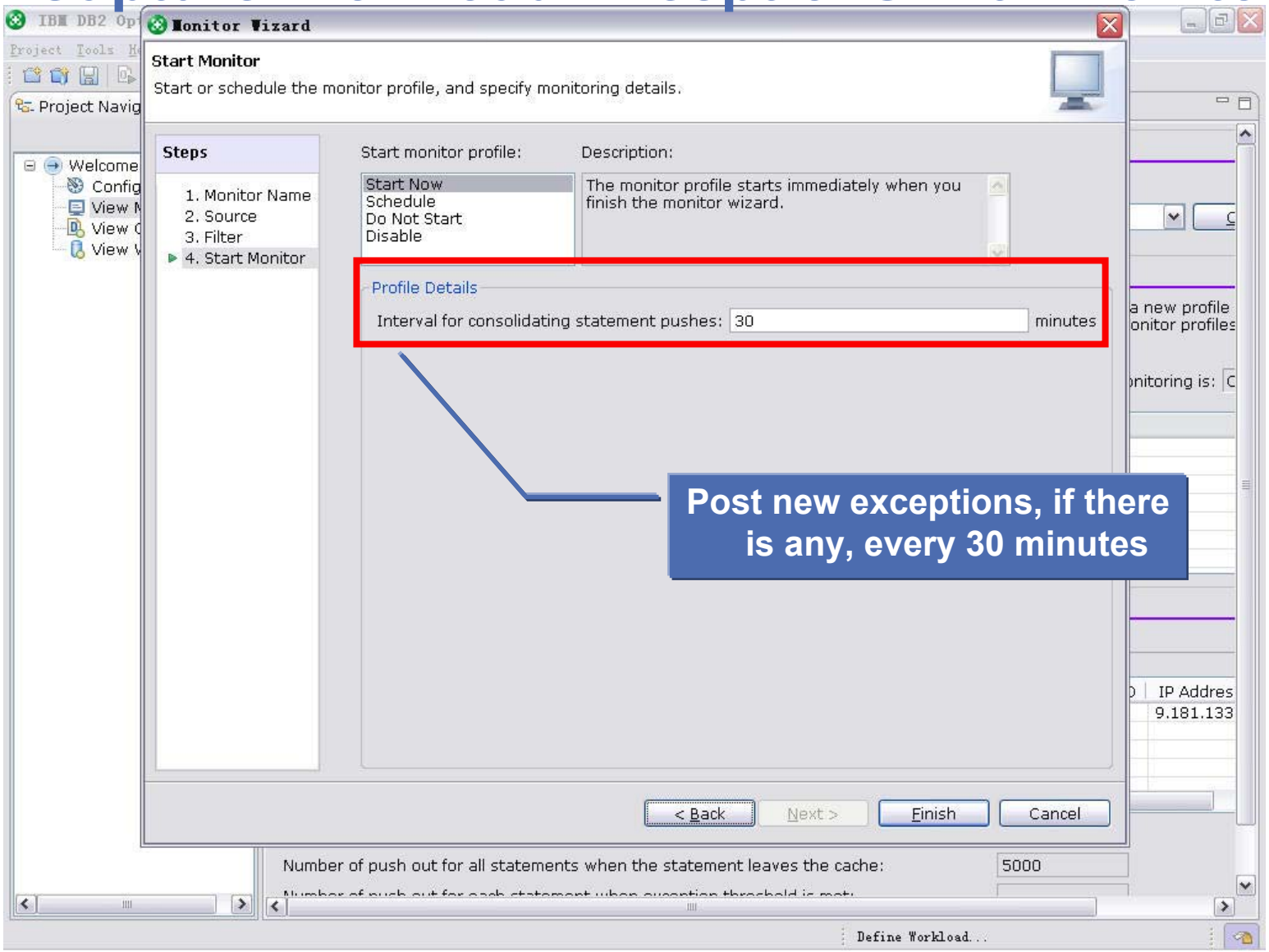
Capture Workload Exceptions with Monitor



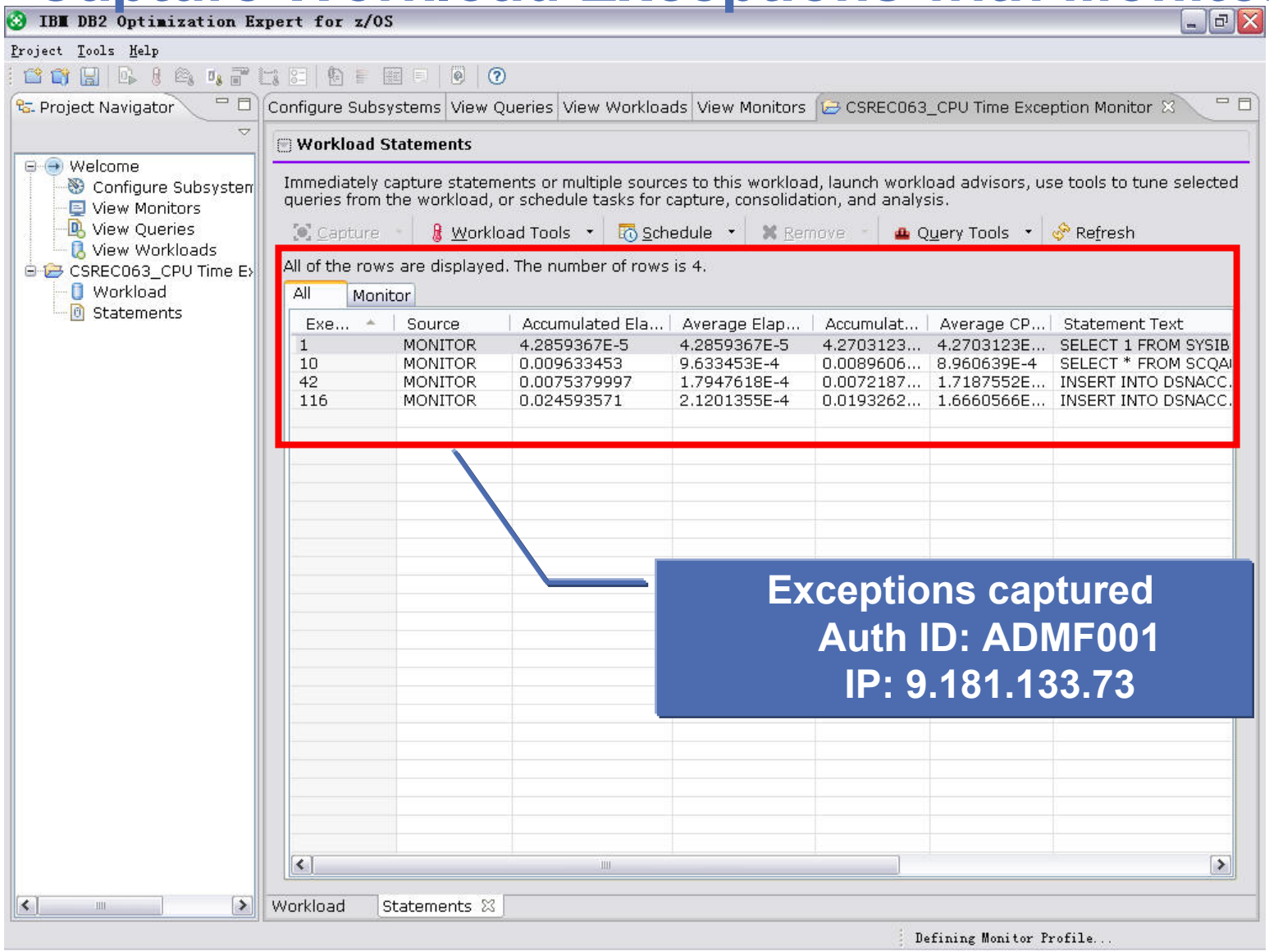
Capture Workload Exceptions with Monitor



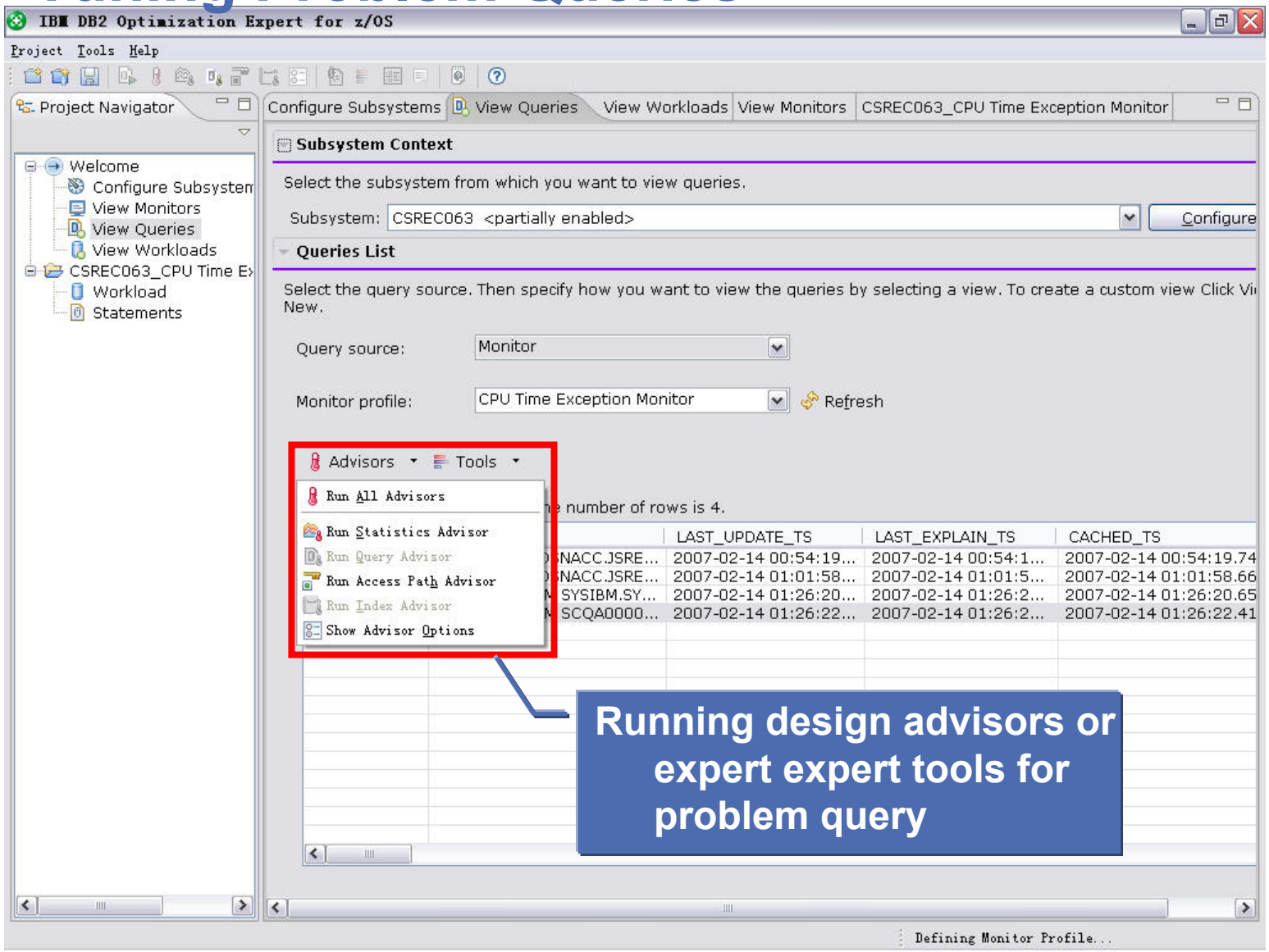
Capture Workload Exceptions with Monitor



Capture Workload Exceptions with Monitor



Tuning Problem Queries



User Feedback Opportunity (UFO) Sessions

- **Data Server Administration Console (DB2 for z/OS, DB2 for LUW, Informix)**
  Provide input on a web-based administration console for the IBM data servers (DB2 for z/OS; DB2 Linux, Unix, Windows; and Informix IDS). We are exploring designs for a health, availability, and troubleshooting user interface.

- **DB2 for z/OS Query Optimization**
  Come and try the latest, generally available IBM Optimization Service Center for DB2 for z/OS (OSC)! This session will focus on the overall user experience of OSC, from Getting Started materials to identifying and analyzing problem queries, reviewing and implementing solution recommendations, and automating tuning across workloads.

- **Getting the Documentation You Want When You Want It**
  Provide feedback on how you use documentation in your shop and how we can make it easier for you to get the information you need when you need it.

- **DB2 for z/OS Futures**
  In small group sessions, help direct the goals of future releases of DB2 for z/OS. Help us understand your role and pain points in using and implementing DB2.

UFO Sessions held in the Willow Glen III Room; stop by to sign up!

Session: C02

Application Performance Tuning in DB2 9 for z/OS

Gene Fuh

IBM Silicon Valley Laboratory

fuh@us.ibm.com