IBM Software Group

SQL…Can't Live with it…Can't Live without it

An IBM System z Software Teleconference

ON DEMAND BUSINESS

COMPUTER PROGRAMMING

COBOL

PL1

ASSEMBLER

RPG II

WHAT IS DIFFERENT?

SQL

Java

STATIC vs DYNAMIC SQL
How the statements are coded

- STATIC
  - SQL statements are built into the program
  - Compile / Link Edit / Bind
  - DBRM
  - Plan / Package
- DYNAMIC
  - Keyed in by a user sitting at a terminal or embedded in host language
  - Assembled at execution time
  - Explicitly prepared

SQL

- Embedded Static SQL
  - SQL statements are included in the host language
    - EXEC SQL
    - #sql
  - Pre-compile, link-edit, and bind
  - Dynamic capabilities
    - Using host variables
      - Execute the same statement repetitively using different values
      - The statement itself is fixed
      - Each program using this method has its own statements
      - Each must have its own DBRM (bound)

SQL

- Embedded Dynamic SQL
  - PREPARE and EXECUTE
  - Pre-compile / link-edit / bind
  - The dynamic statements are prepared and executed when invoked at runtime
- Interactive Dynamic SQL
  - SPUFI
  - QMF
  - Report Writers

SQL

- Deferred Embedded Dynamic SQL
  - Used by DB2 private protocol to access remote data
  - Alias or use of 3 part name
    - DSNB.PROD.EMPLOYEE
  - SQL has characteristics of STATIC in that it is embedded and hard coded
  - SQL is like DYNAMIC because it is prepared at runtime prior to execution
  - No plans / packages on the target subsystem
  - Recommendation is to migrate from private protocol to DRDA
- DB2 processes SQL accessing declared temporary tables the same way it does using the private protocol

SQLJ - Java

- SQLJ – API for embedded static SQL in Java programs
- SQLJ translator – transforms SQL into calls for the runtime environment
- Profile files – used at execution time; can be used to create DBRM's
- Compile / Link-edit / Bind
- Dynamic
  - Uses host expressions (vs. host variables)
  - No syntax to handle dynamic SQL statements – rather use JDBC – the 'static' code acts as a dynamic SQL statement
    - Transparent to the user

CALLABLE INTERFACES

- ODBC – available for C and C++
  - Uses function calls
  - Does not require pre-compilation or binding
- JDBC – can access any relational data base
  - Uses function calls
  - Does not require pre-compilation or binding
- REXX
  - Uses RRS or CAF attach to connect to DB2
  - No pre-compilation
  - Comes with a pre-defined set of packages

STATIC vs DYNAMIC SQL vs Combined
By the way the statements are executed

- STATIC
  - Requires a DBRM (Database Request Module)
  - Statements are executed directly without preparation
- DYNAMIC
  - SQL statements have not been bound
  - No executable form of the statement exists
- COMBINED
  - Deferred embedded SQL
  - SQLJ coming in through JDBC interface

STATIC vs DYNAMIC SQL

- STATIC
  - Performance
    - Non-uniformed distribution – the path chosen at bind time may be sub-optimal
  - Authorization benefits
    - User needs EXECUTE privilege on plan / package
  - Auditing
    - Code exists in source libraries
    - Easy to manage change management
- DYNAMIC
  - Performance
    - Overhead of doing the prepare
    - Caching of SQL helps
    - If data is skewed, dynamic SQL may perform better
  - Authorization
    - Users will need explicit authorizations on the object(s) (could use DYNAMICRULES(BIND) on embedded dynamic SQL)
    - Stored Procedure alternative
  - Auditing – not possible to determine if the statement was modified before execution

IBM Software Group

Identifying Problematic SQL using IBM DB2 Query Monitor for z/OS

ON DEMAND BUSINESS

How Do You Find SQL To Tune?

- DB2 subsystem monitors
  - IBM Tivoli OMEGAMON XE for DB2 Performance Expert
  - IBM Tivoli OMEGAMON XE for DB2 Performance Monitor
  - ISV tools
- SQL activity trace
- SQL activity monitors
  - IBM DB2 Query Monitor for z/OS
  - ISV tools

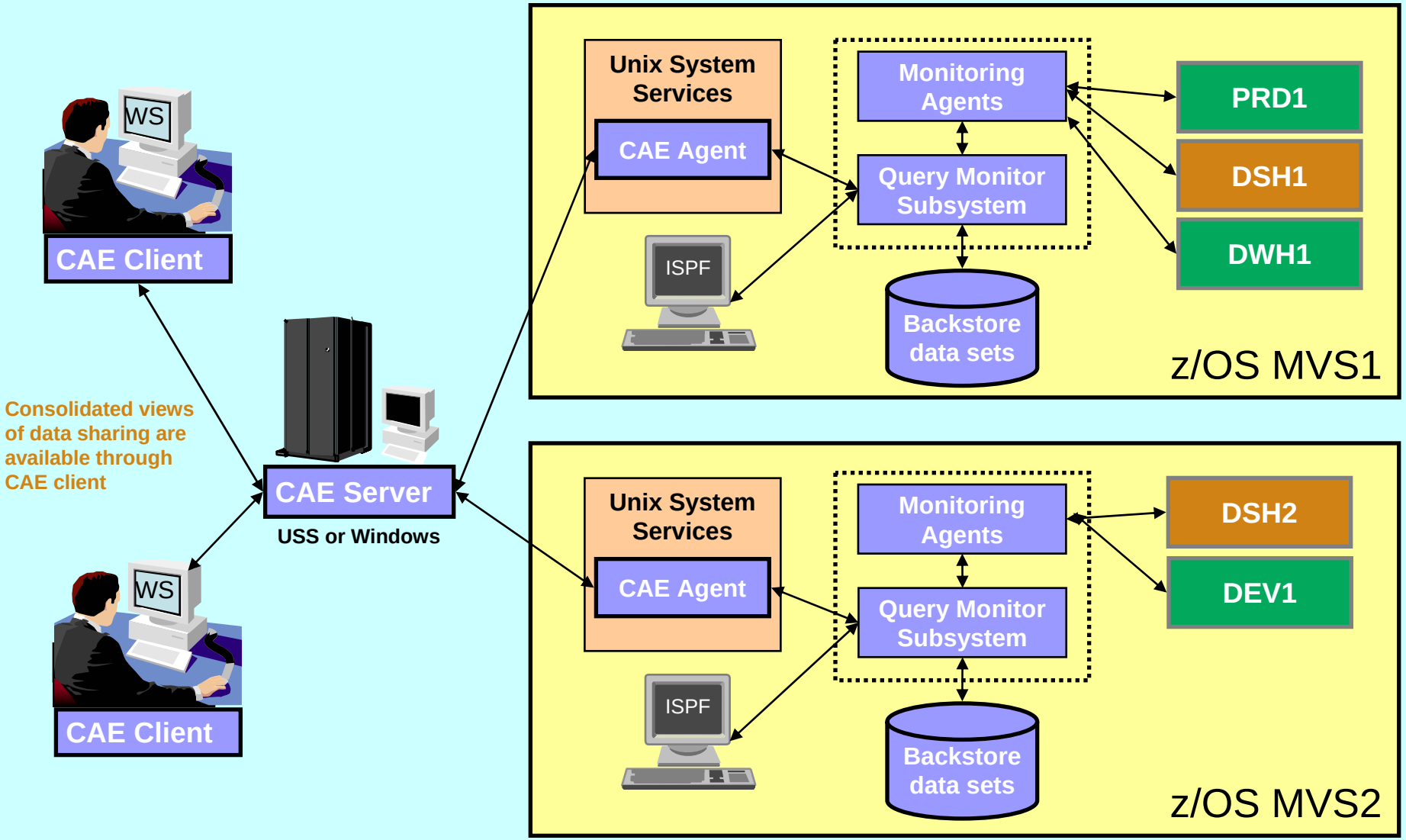
About DB2 Query Monitor for z/OS

- What it is
  - An SQL statement monitor
    - Real-time and historical data
    - Low overhead
      - Traces ACCTG(1,3) and STATS(1,3,4)
    - Easy to use
      - Appropriate for DBAs and application programmers
    - A complementary tool to an existing DB2 subsystem monitor
- What it isn't
  - A DB2 subsystem monitor
  - A DB2 thread monitor
  - A post-processor of trace data

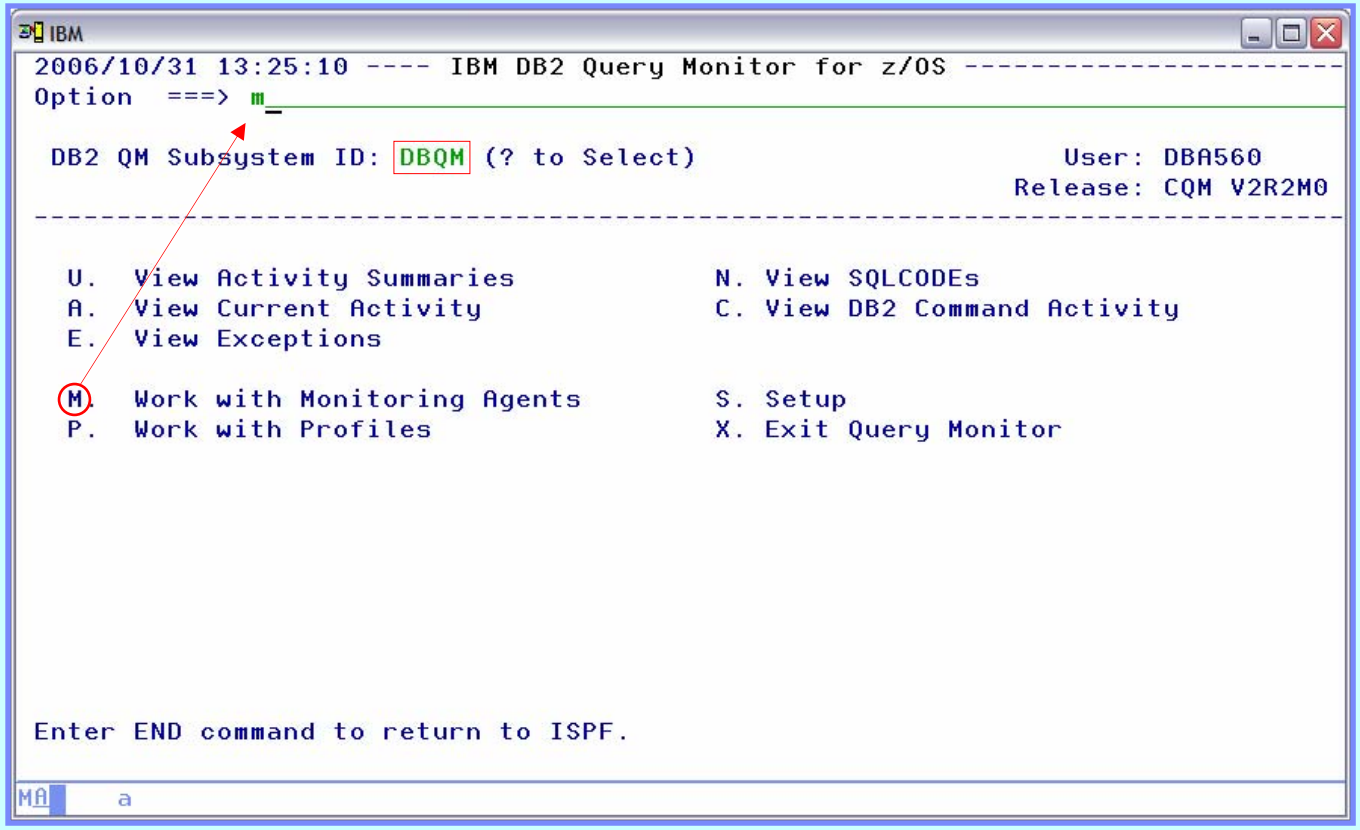
What Does Query Monitor Do?

- Enables quick and easy identification of problem SQL
  - View by plan, package, authid, SQL statement, etc.
  - Key metrics - CPU, elapsed time, GETPAGES
- Allows you to proactively manage DB2 resources
  - Exceptions and alerts by workload
  - Automated actions for alerts
- Allows you to react quickly and effectively to DB2 performance problems
  - ISPF, GUI and web interfaces

DB2QM – Architecture

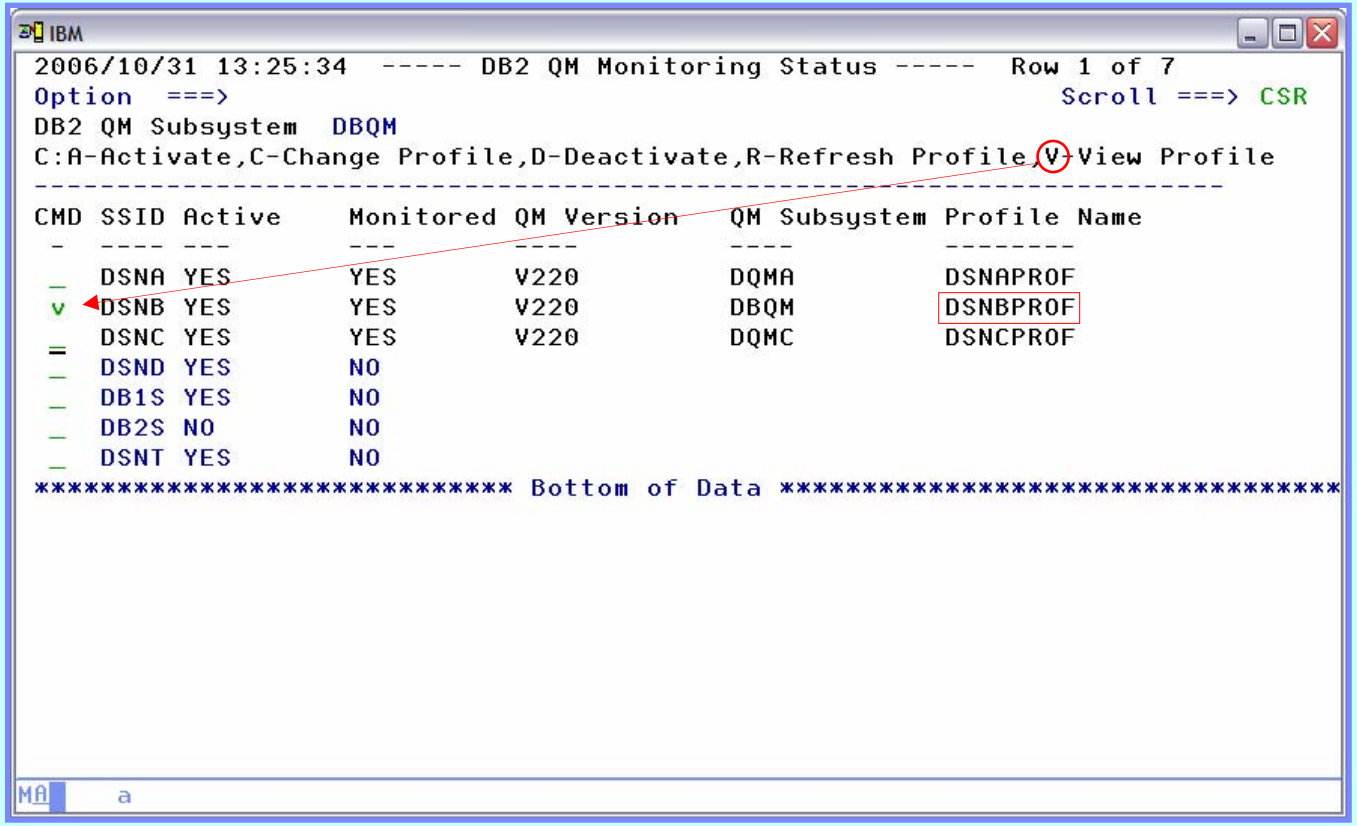


DB2QM – Main Menu



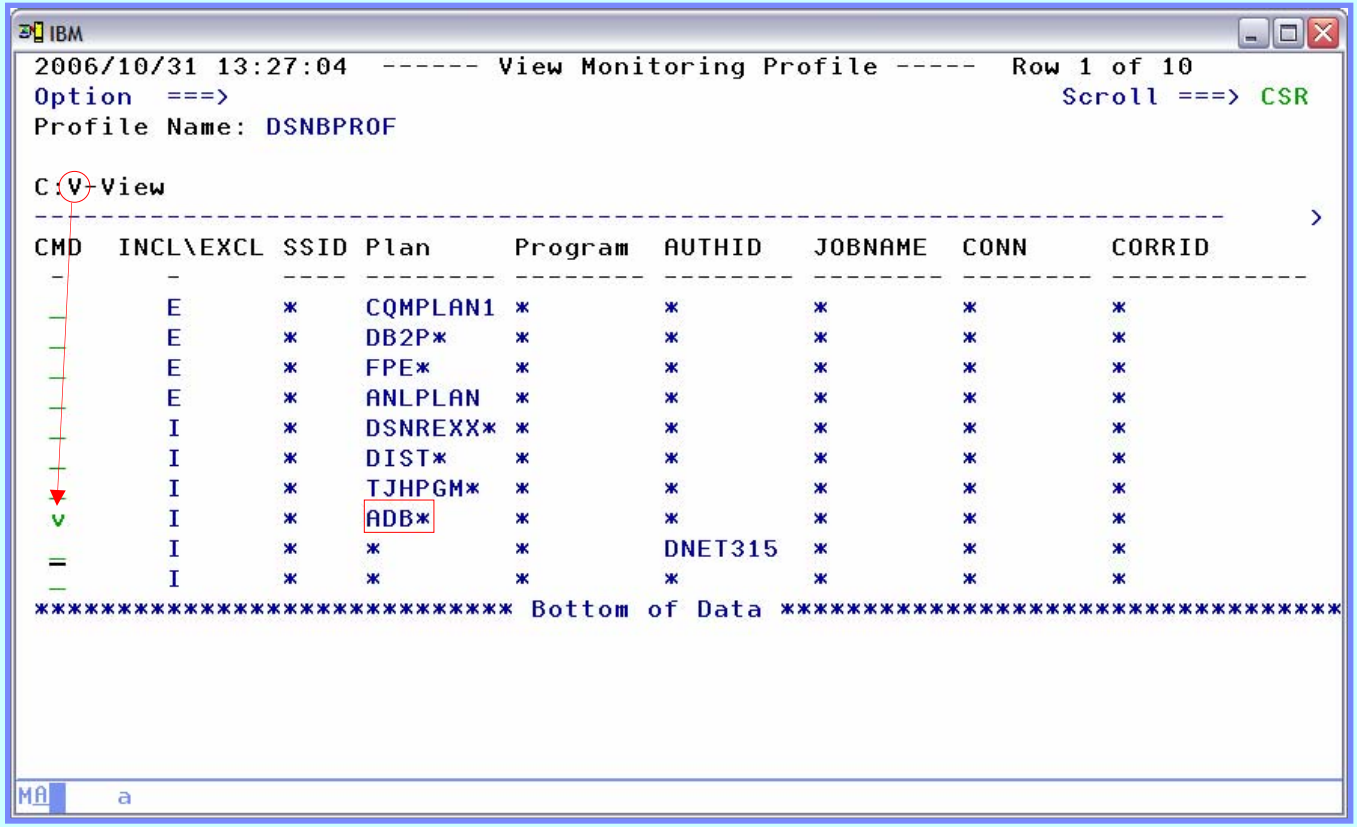
- 'DBQM' is the name of our QM collector
- Use option 'M' to see the monitoring status of all DB2 subsystems on this LPAR
- Options 'U' and 'E' can be used to find problem SQL statements
- Option 'N' can be used to find SQL errors
- Option 'C' can be used to view all commands issued to monitored DB2 subsystems

DB2QM – Monitoring Status



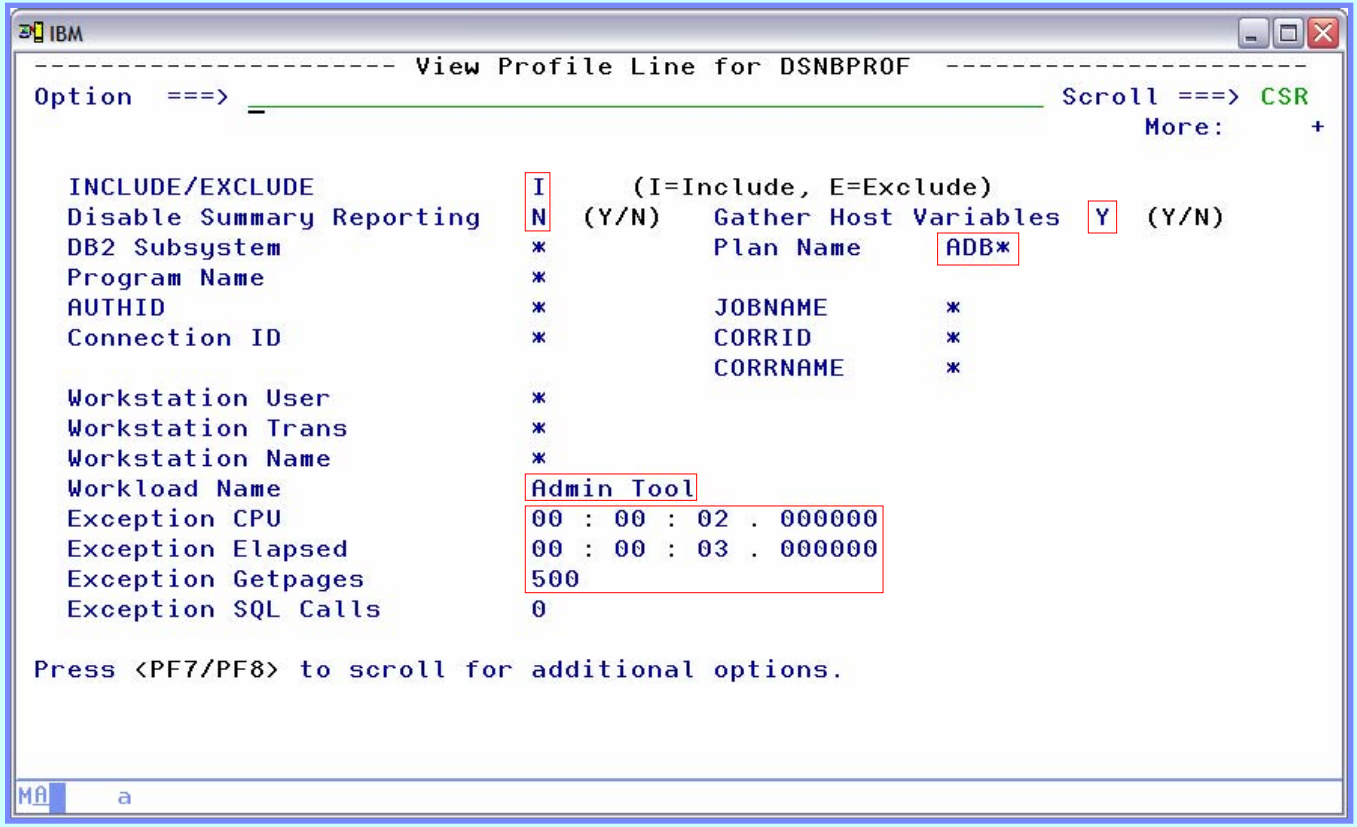
- Monitoring profiles may be dynamically changed or refreshed
- The monitoring agent may be dynamically activated and deactivated
- Use option 'V' to see the details of a monitoring profile
- Note: this is not a typical configuration – generally there will be 1 QM subsystem per LPAR

DB2QM – Monitoring Profile Summary



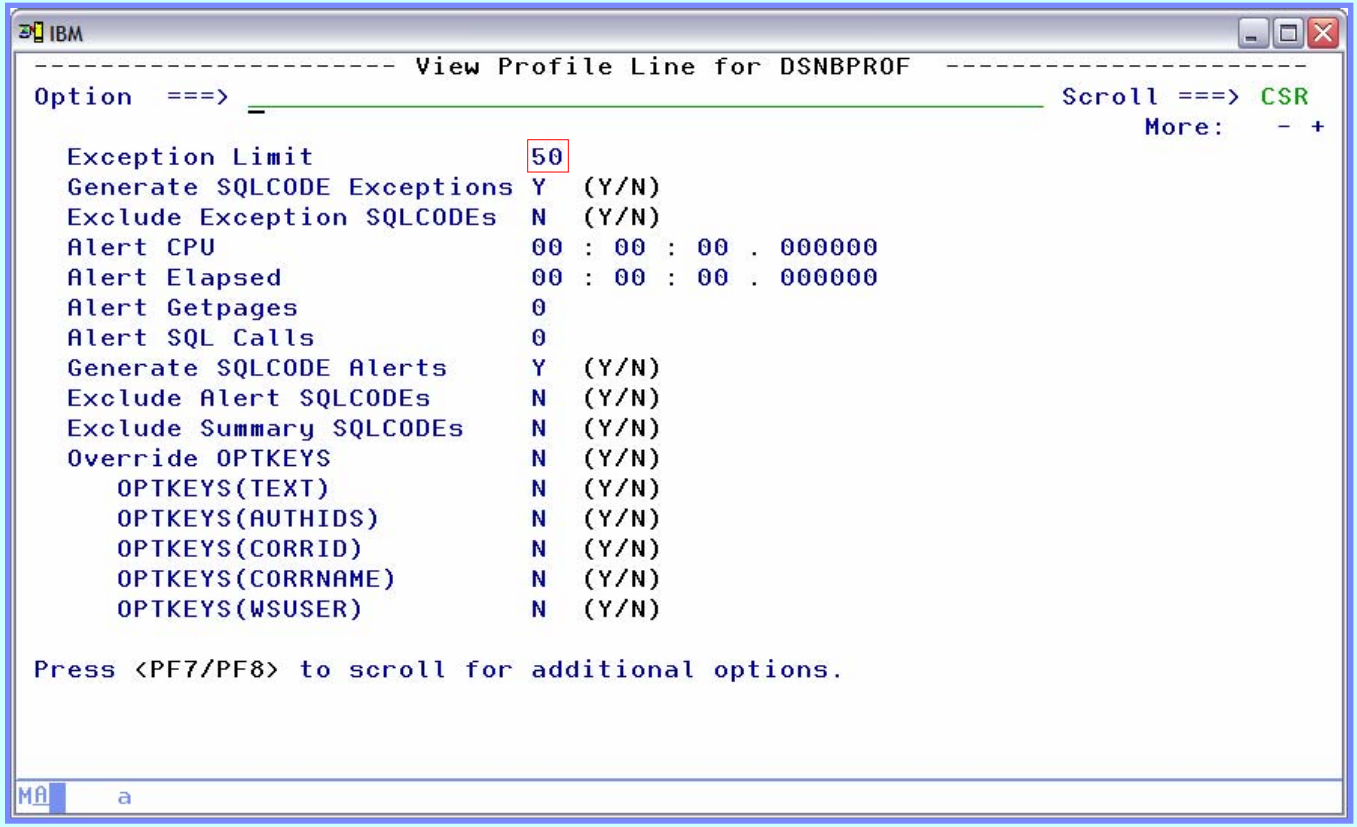
- Workload is assigned to the first matching line in the profile
- Use wildcarding, when possible, to limit number of entries to search
- Specify excludes first
- Specify a ‘catchall’ definition at end for unmatched workload

DB2QM – Monitoring Profile Detail (1)



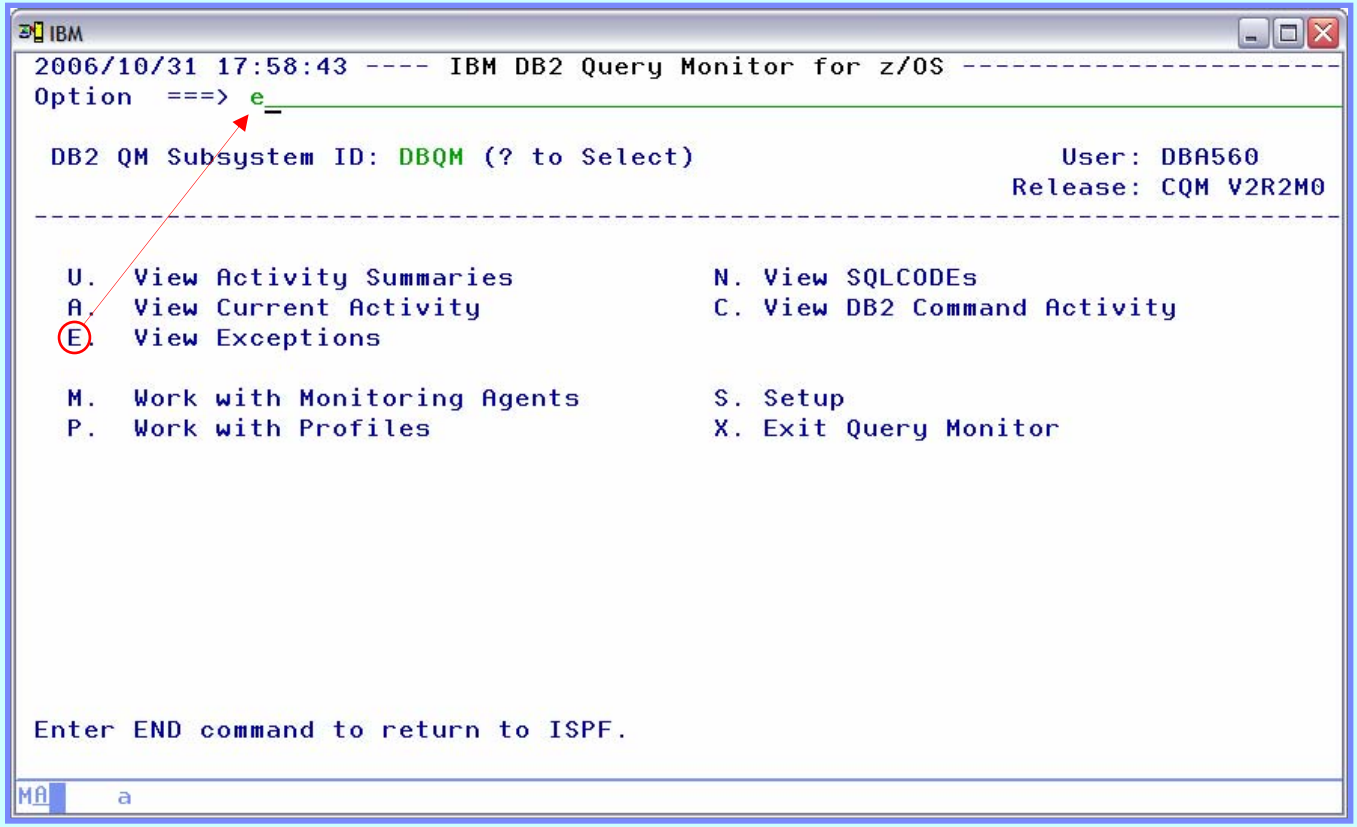
- Definition lines can be used to exclude workloads from summary and exception reporting
- Host variables may be collected
- An ‘and’ condition is applied to filters, i.e. – all must match
- The workload name is shown on the exceptions display
- An ‘or’ condition is applied to thresholds
- A value of ‘0’ means “don’t test”

DB2QM – Monitoring Profile Detail (2)



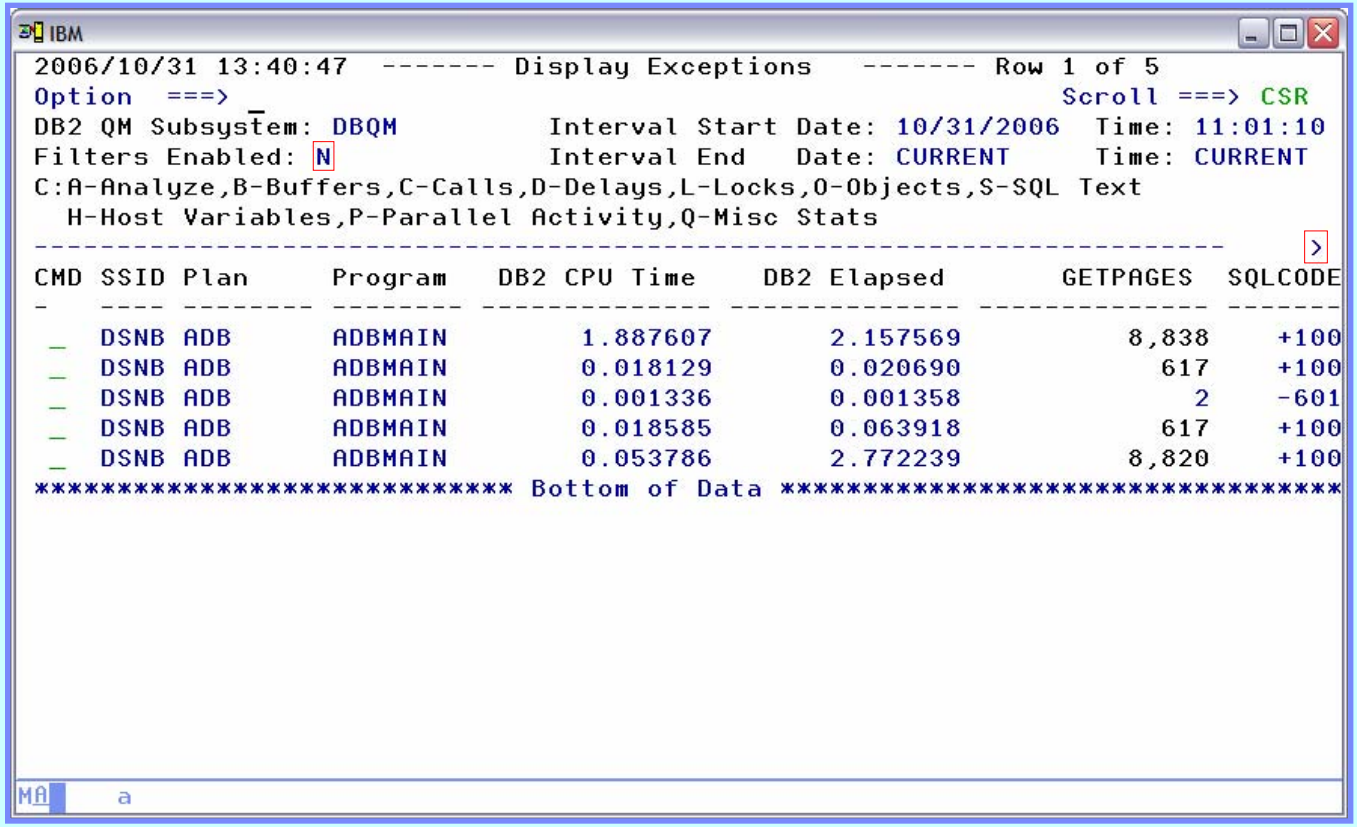
- The exception limit should be a reasonable value to maintain low overhead for collection
- Alert thresholds should be higher than exceptions
- Alerts are sent to the CAE server
- Optional summarization can be controlled at the workload level via the OPTKEYS override

DB2QM – View Exceptions



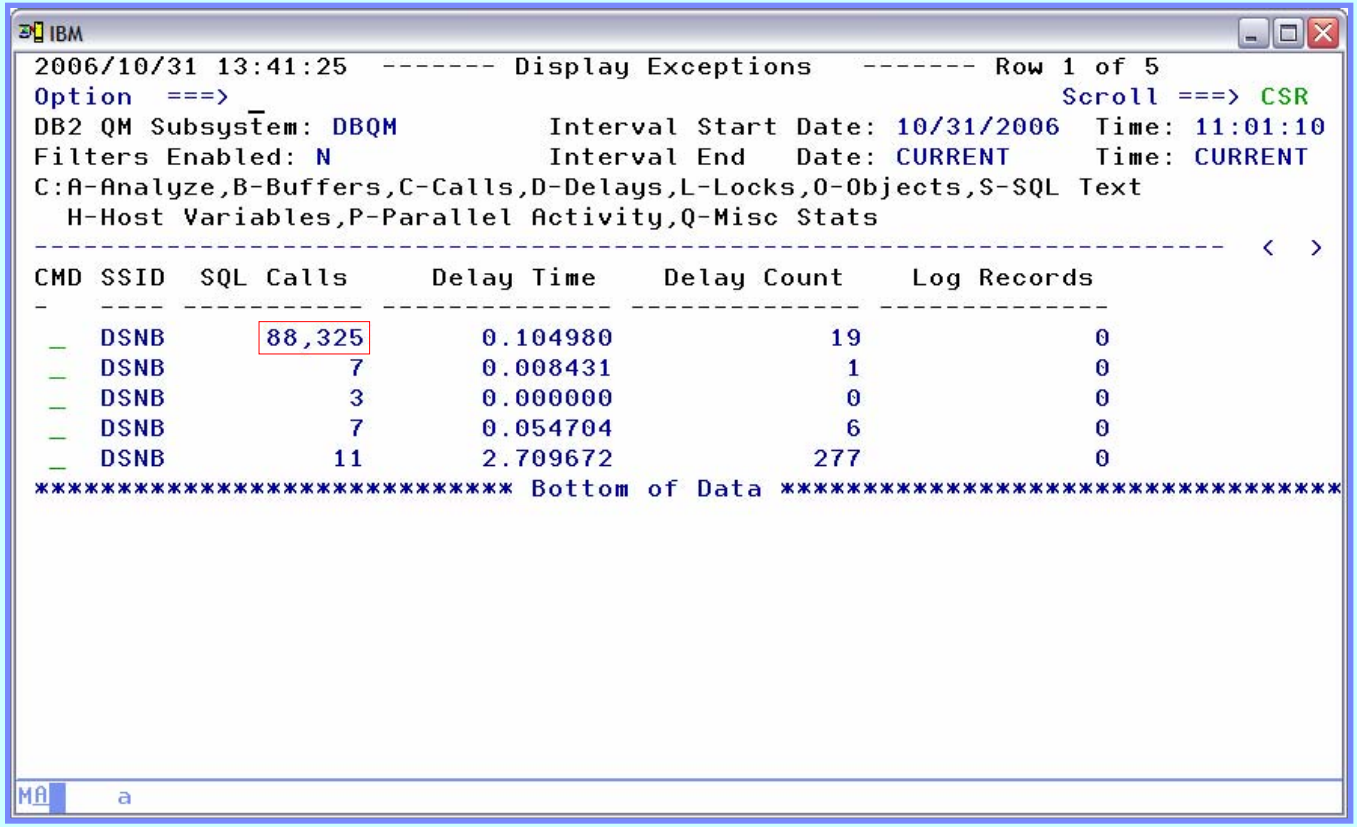
- Exceptions are a good place to start looking for poor performing SQL

DB2QM – Exceptions (1)



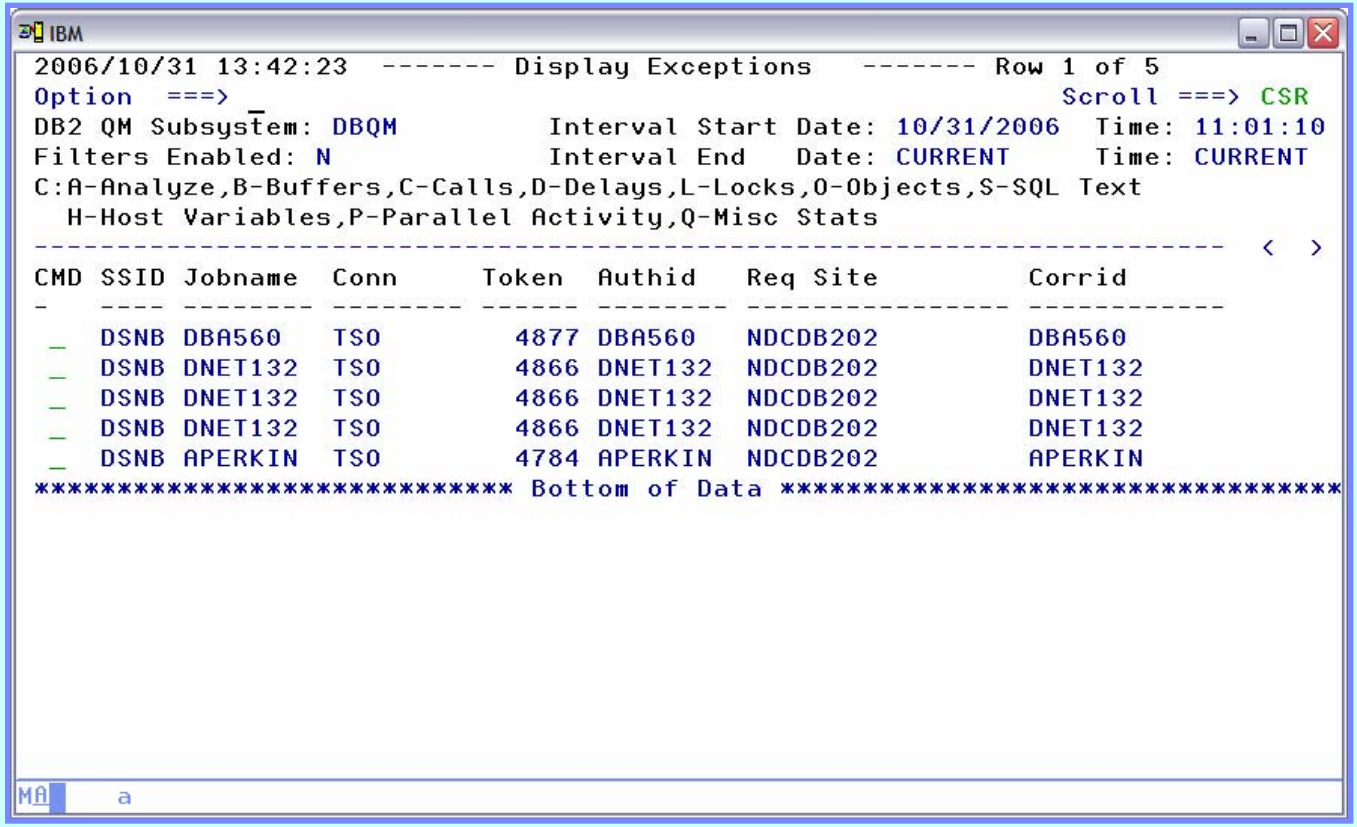
- One line is displayed per exception event
- Significant data is available to the right (PF11) and through drill-down commands
- The key performance indicators (CPU, elapsed, and GETPAGES) are displayed on the first panel
- The triggering threshold is displayed in an alternate color – the GETPAGES count (lines 1,2,4,5) is highlighted
- Filtering is available to limit displayed data by plan, package, authid, etc.

DB2QM – Exceptions (2)



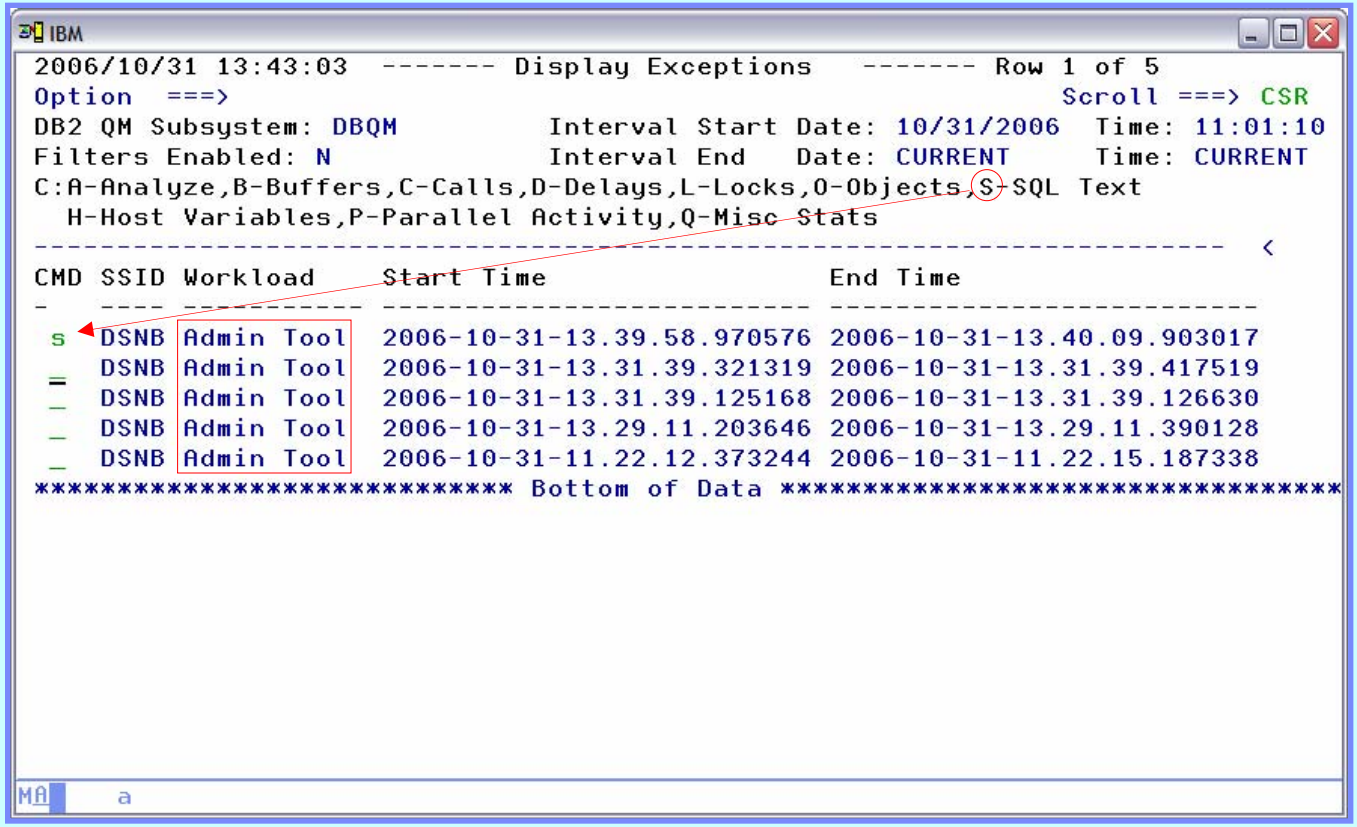
- Additional data is available as we scroll to the right
- The high number of SQL calls and high delay times of the first and last exception may be worthy of further investigation

DB2QM – Exceptions (3)



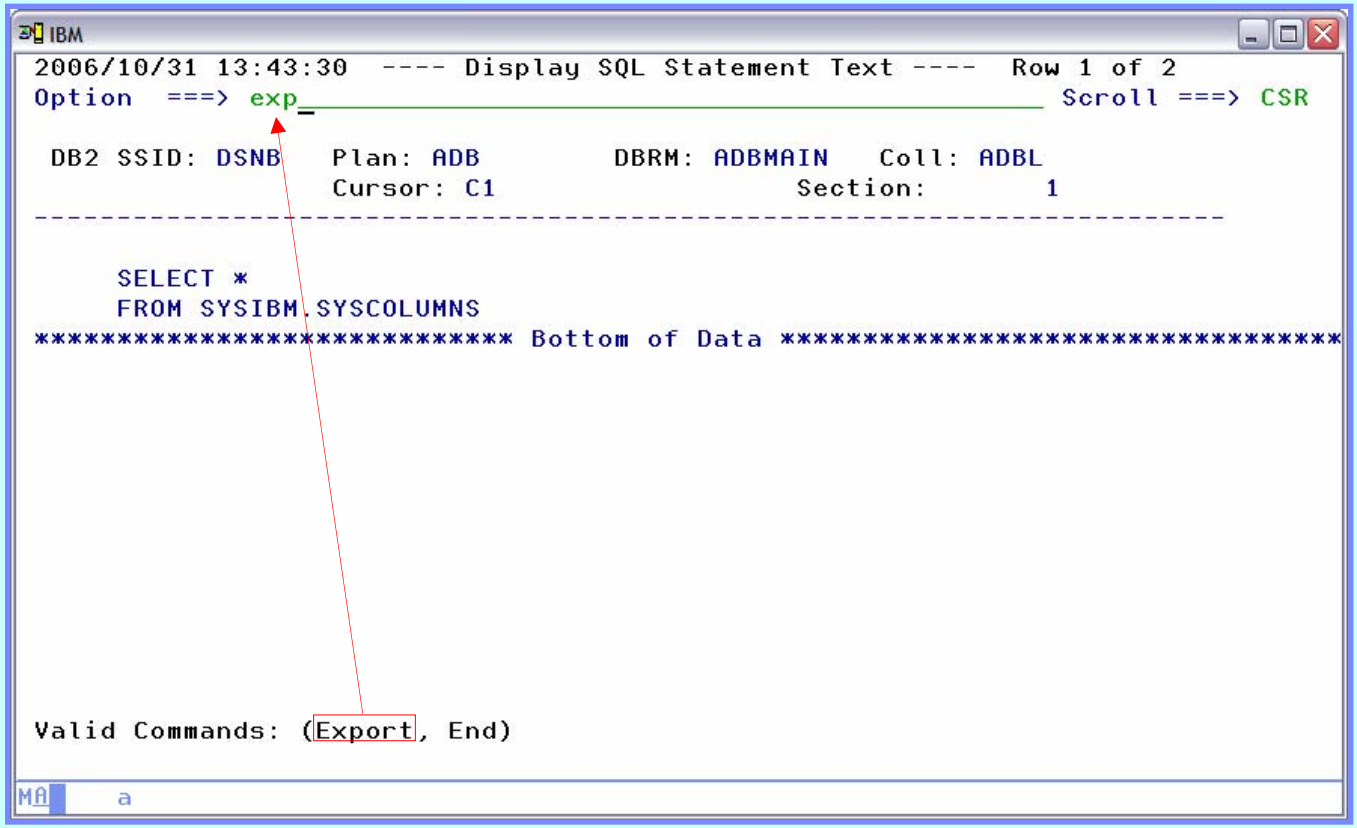
- Key data to identify the user that caused the exception is also available
- The workstation identifiers are available for distributed applications such as SAP and PeopleSoft by scrolling to the right

DB2QM – Exceptions (4)



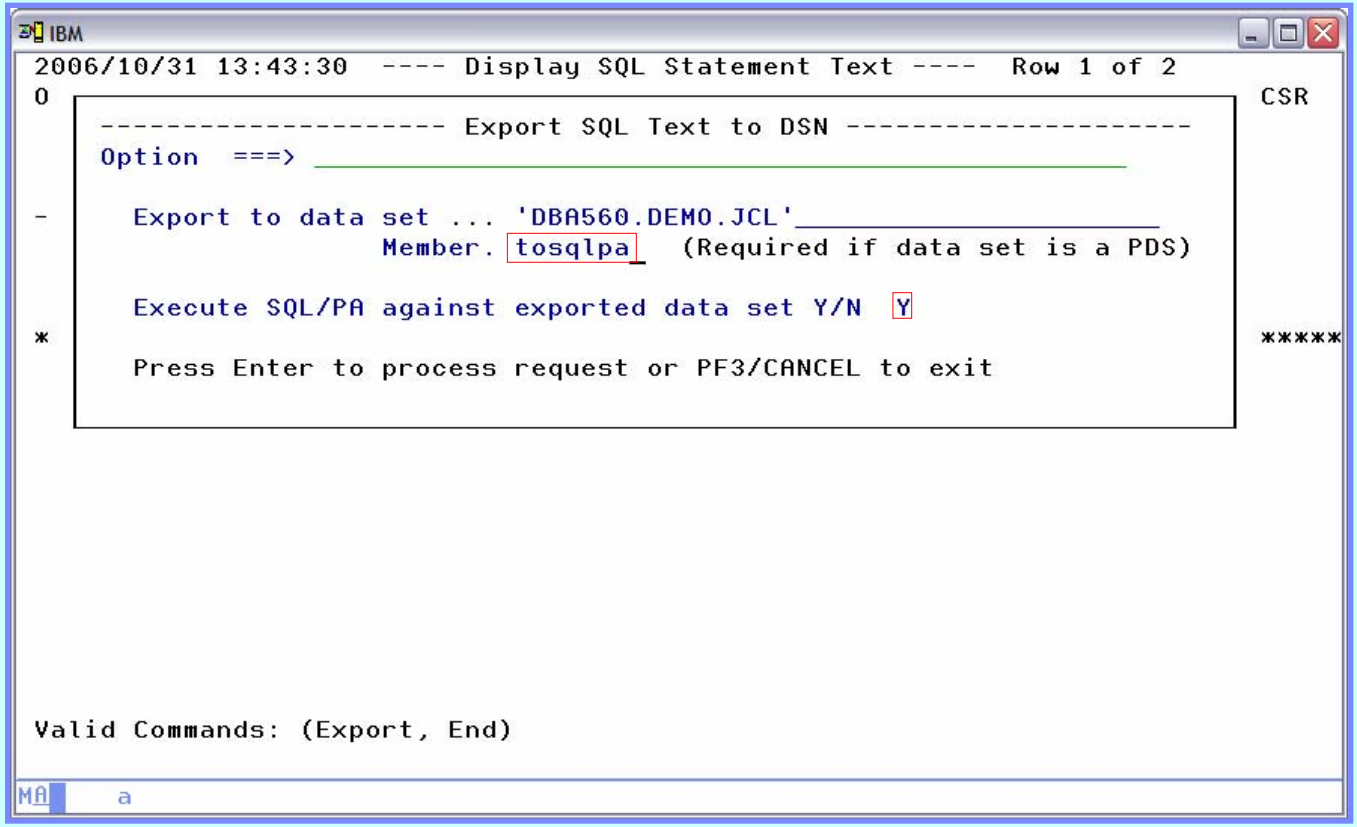
- Notice the workload name for each exception
- Let's use the 'S' line command to look at the SQL text of the statement with a high number of calls

DB2QM – SQL Text Display



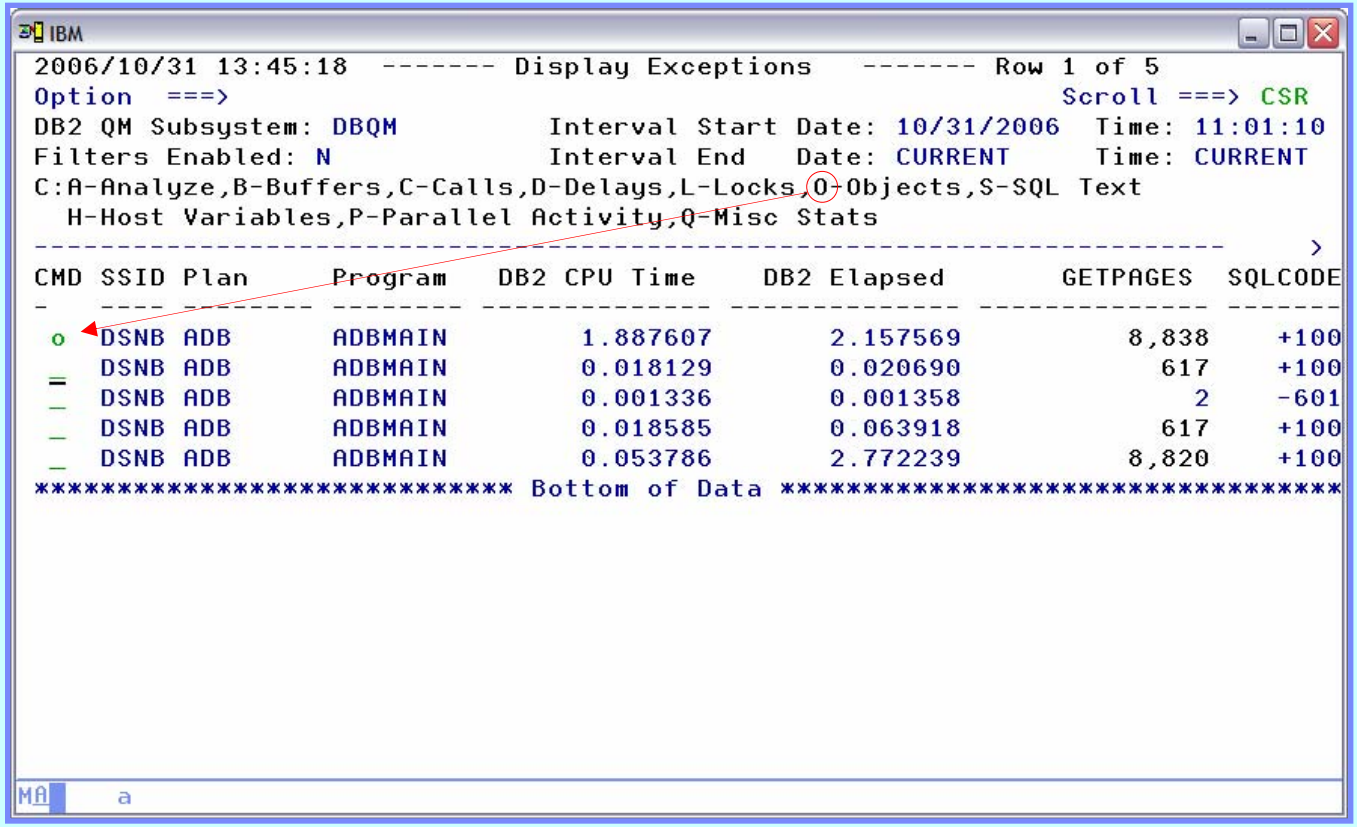
- Not a complex query – but the user probably didn't consider the amount of data they were requesting
- The 'Export' command may be used to save the query and optionally pass it to IBM DB2 SQL Performance Analyzer
- The complete text of the SQL statement is displayed – 32KB for V7 and 2MB for V8

DB2QM – Export SQL Text



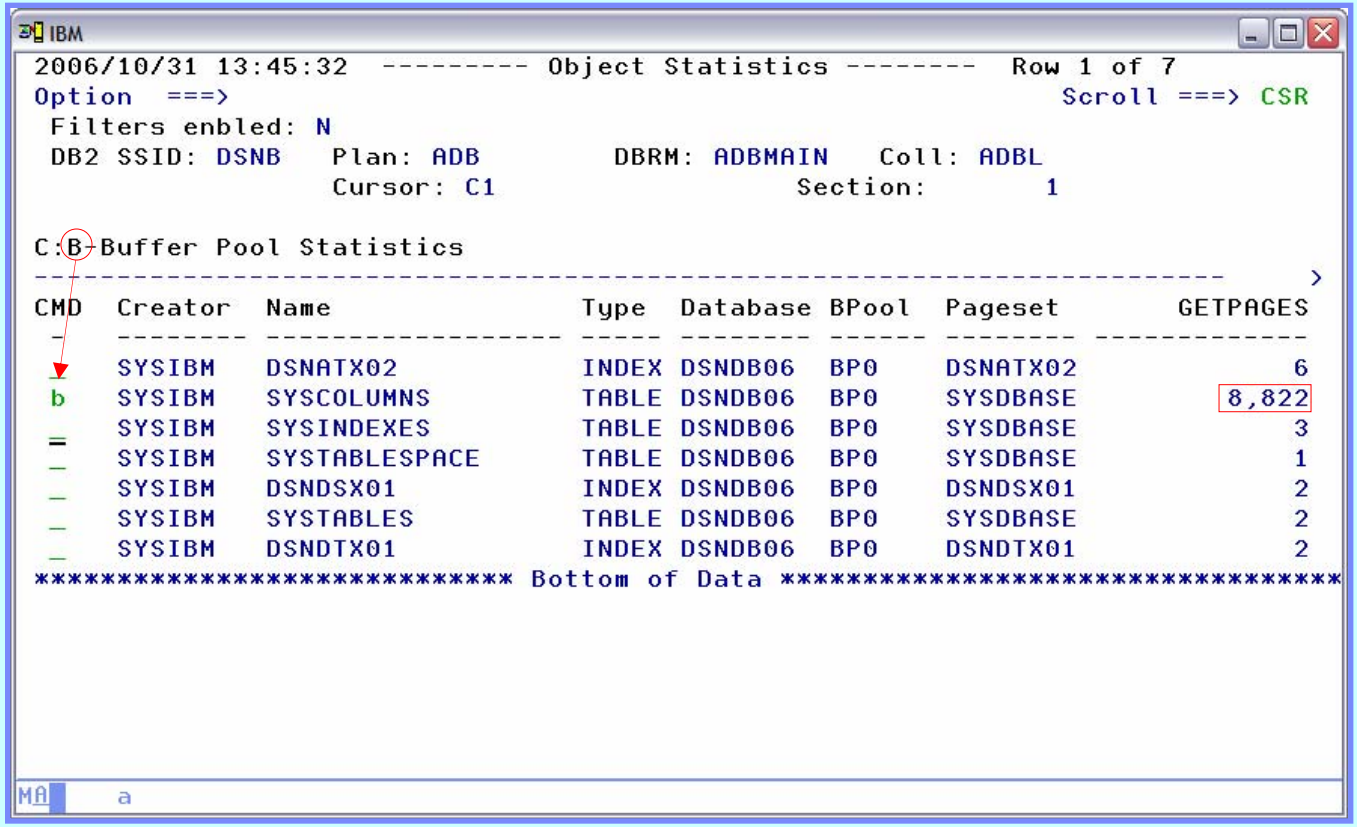
- The export pop-up requests a location to store the SQL text, and whether or not to invoke SQL PA
- The 'A' line command may be used from the 'Display Exceptions' panel to directly invoke SQL PA without first exporting to a data set

DB2QM – View Objects Related to Exception



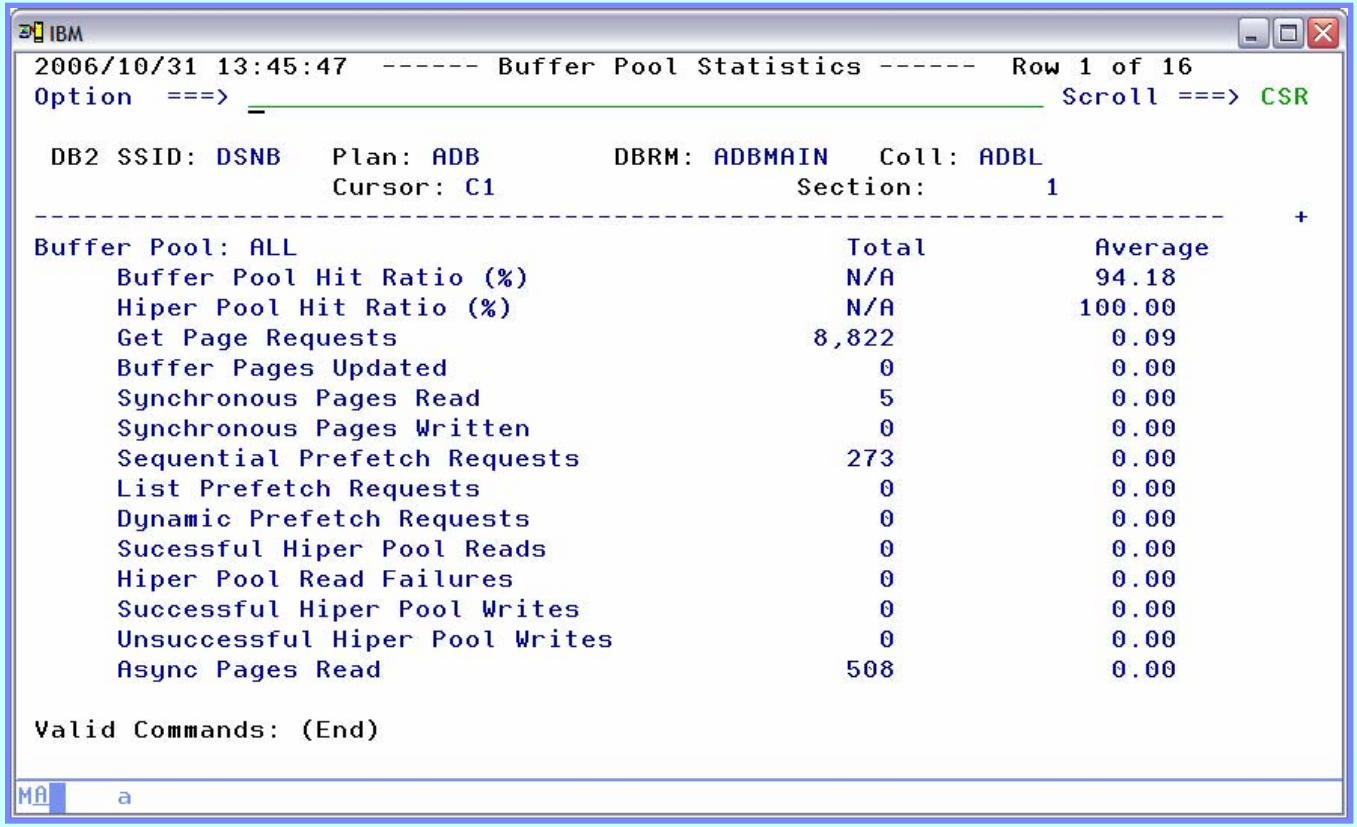
- We can drill-down to see detailed metrics related to buffer pools, delays, locks, etc.
- We can also look at the objects used by the query

DB2QM – Object Statistics



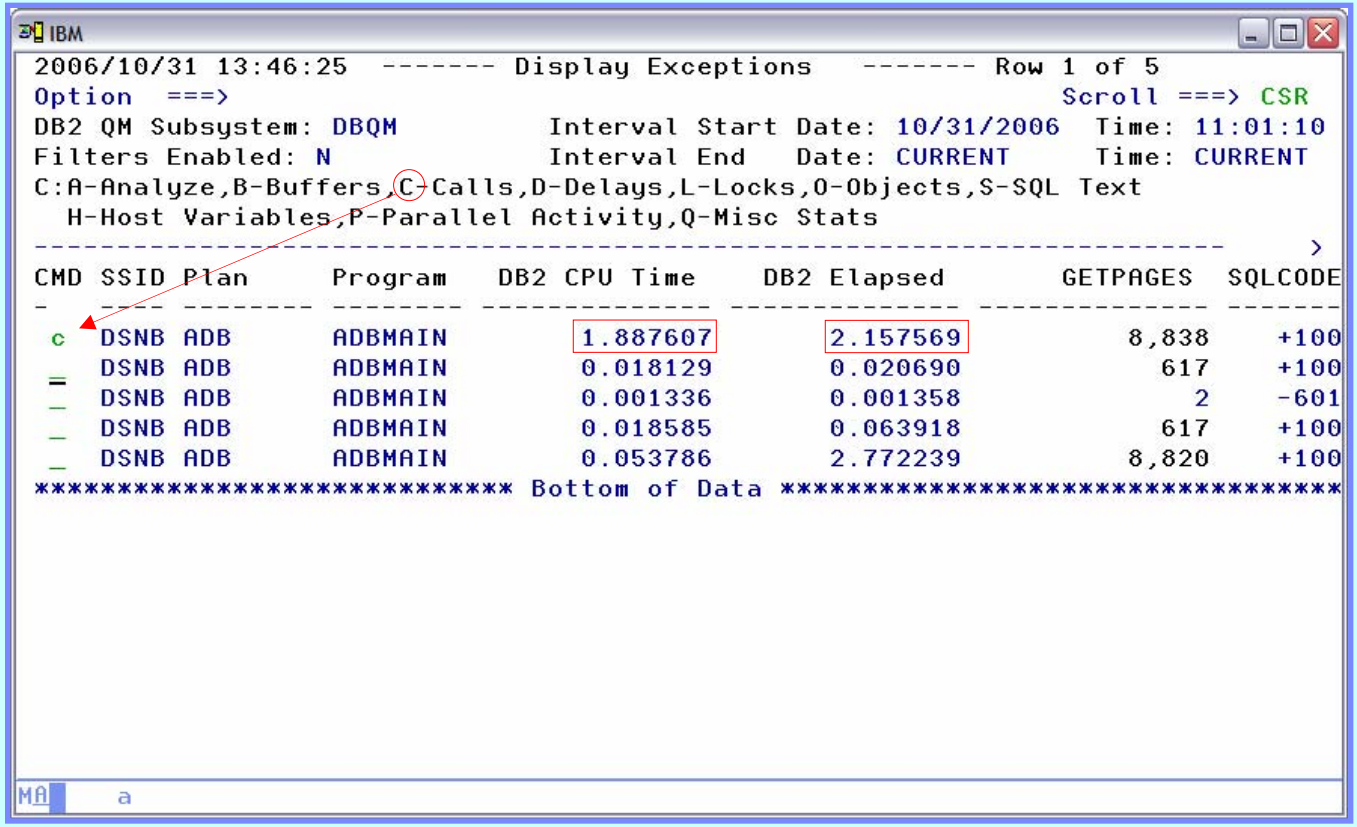
- Tables and indexes related to the exception are displayed
- We can scroll to the right for more detail and use the ‘B’ line command to display buffer pool details for each object

DB2QM – Buffer Pool Statistics



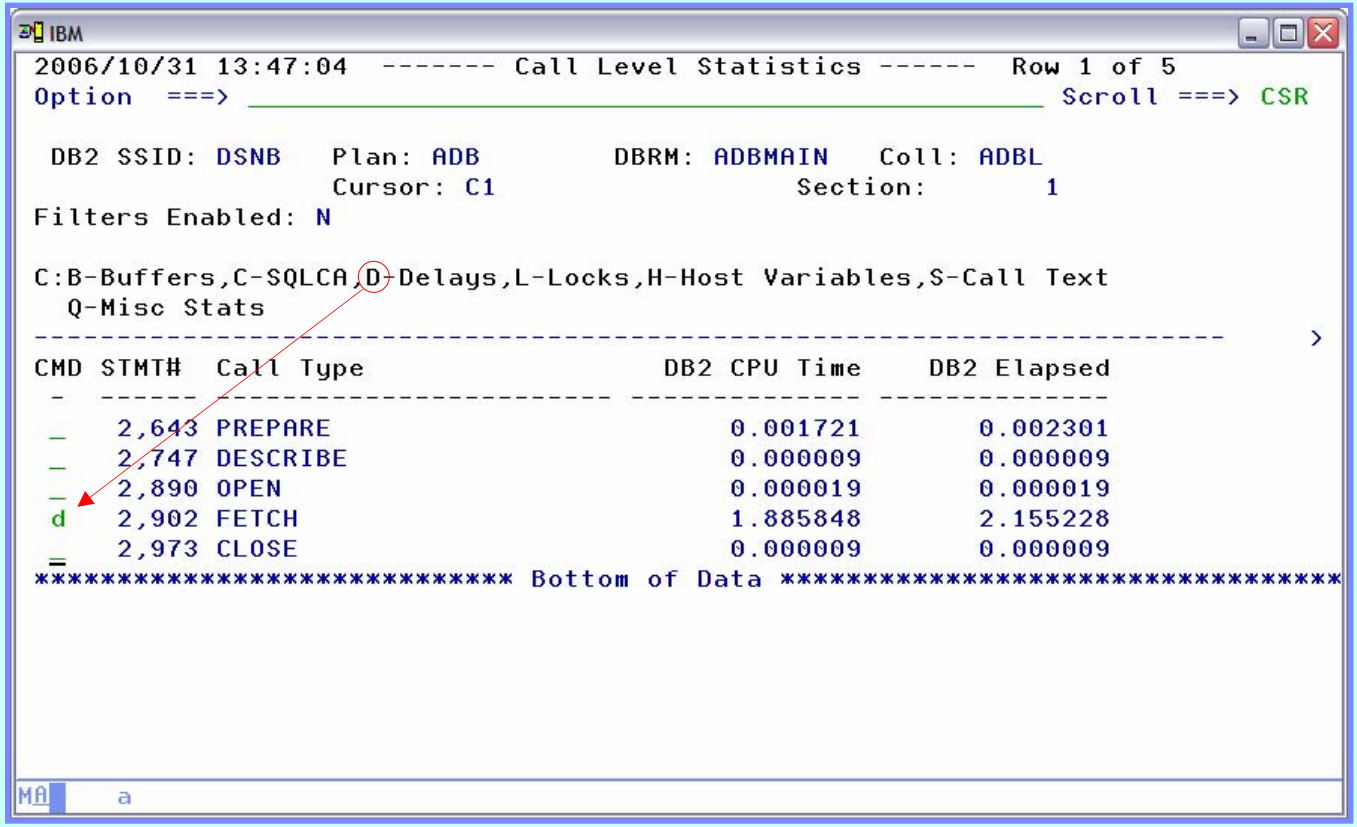
- We now see the buffer pool statistics, for table SYSCOLUMNS, related to the query that triggered the exception

DB2QM – View Calls for an Exception



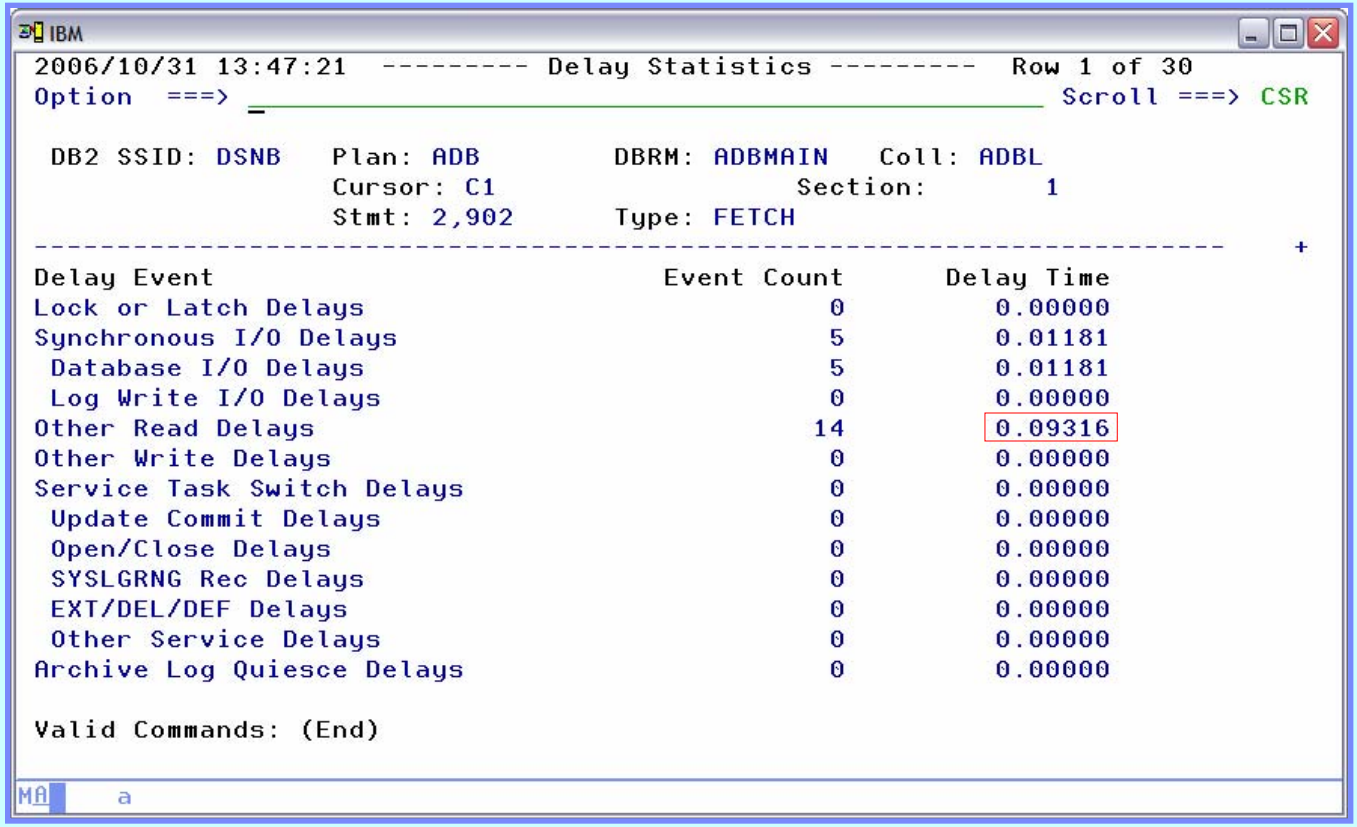
- The ‘C’ line command may be used to display statements by type
- Calls is an optional summary bucket controlled by the OPTKEYS setting

DB2QM – View Delays for a Call



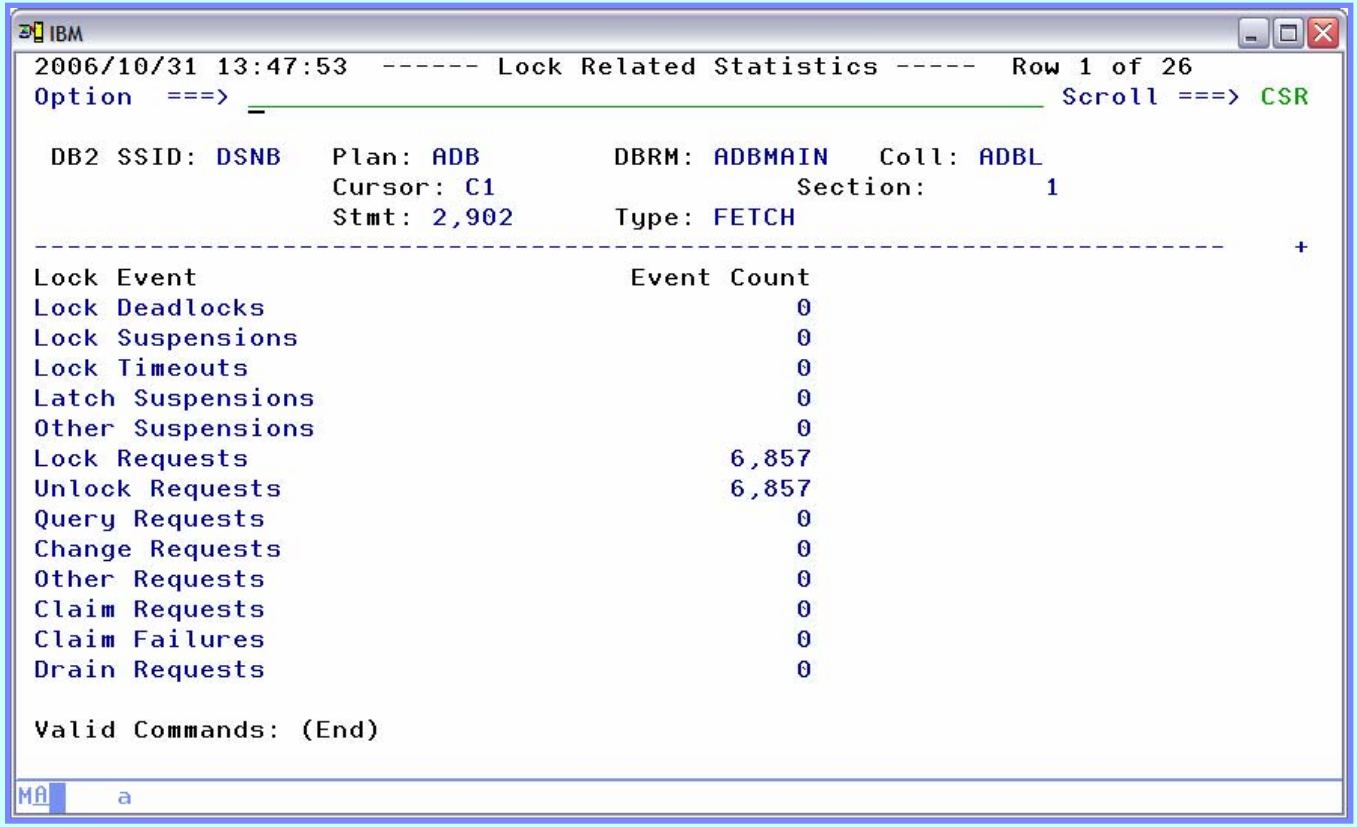
- Call level statistics allows us to view metrics related to the individual parts of a query
- Delays, locks and miscellaneous stats may be useful to diagnose performance issues

DB2QM – Delay Statistics (1)



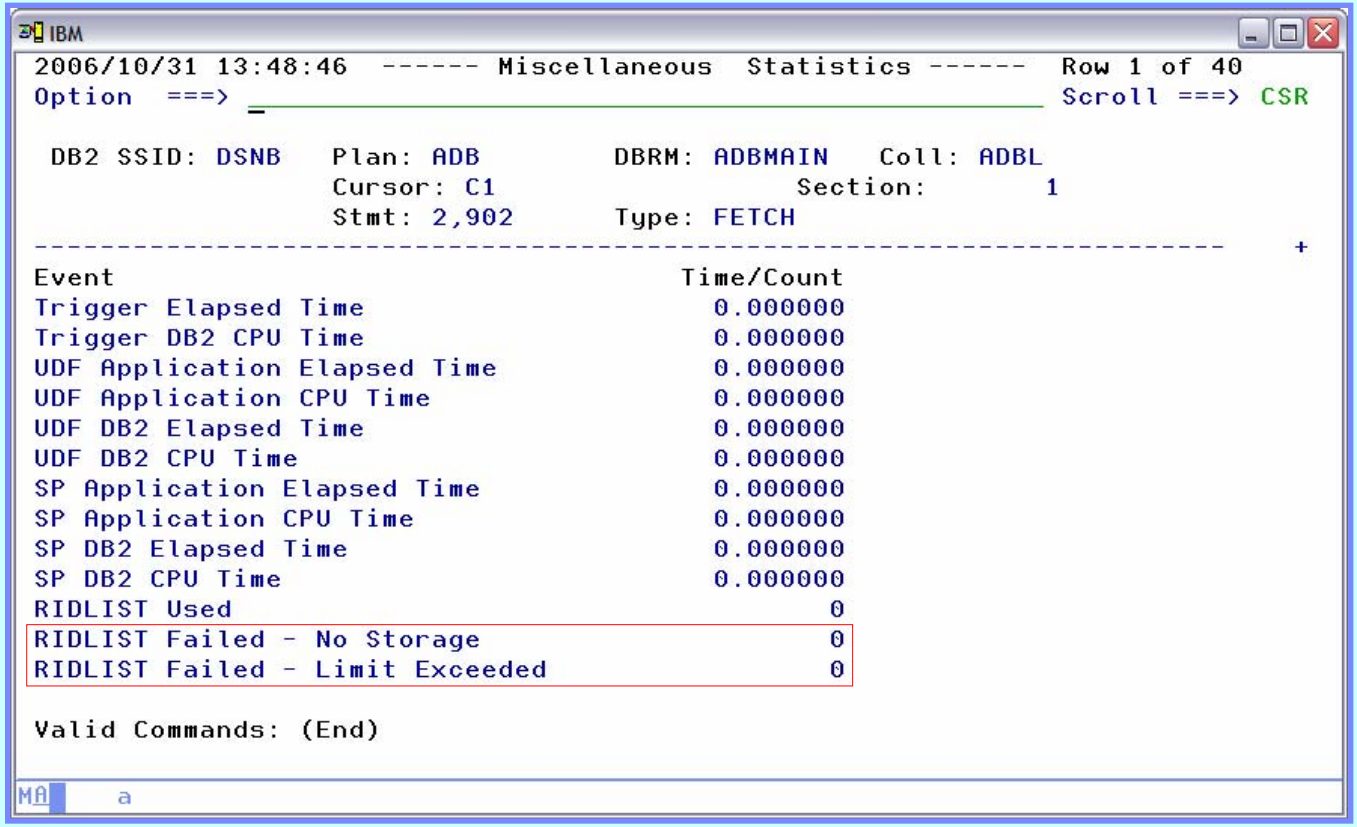
- Delays elongate the elapsed time of a query and should be investigated

DB2QM – Lock Related Statistics (2)



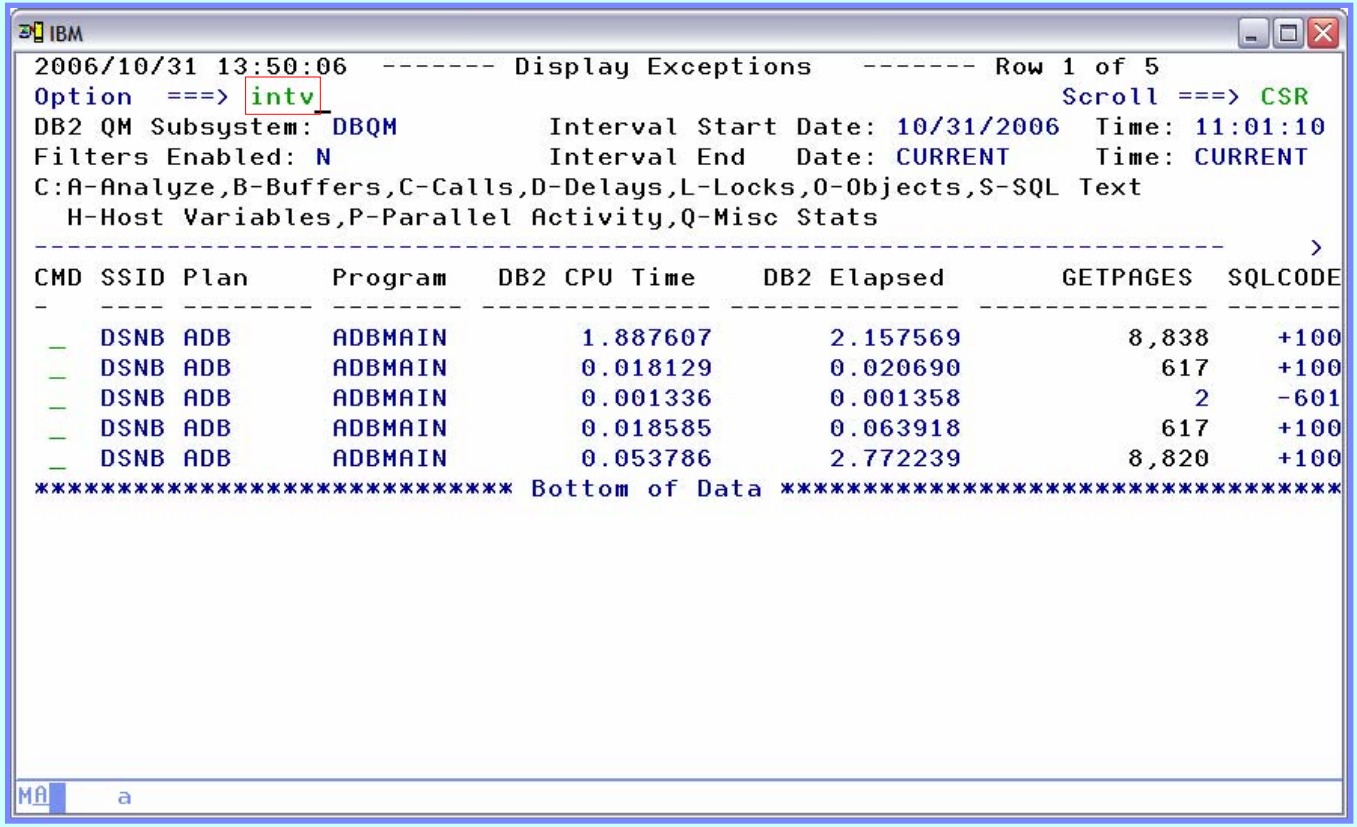
- High lock requests should be investigated – be sure the isolation level requested is appropriate

DB2QM – Miscellaneous Statistics



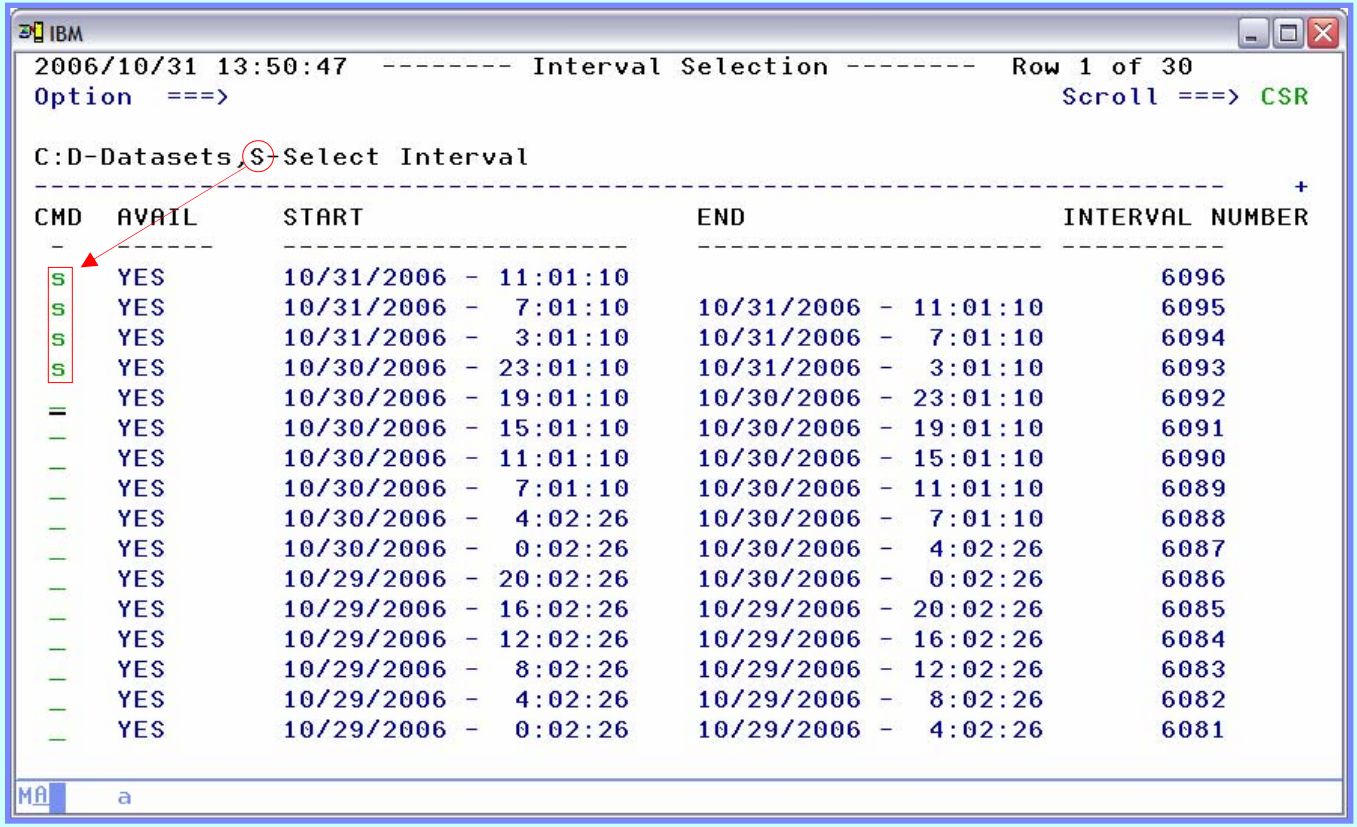
- Miscellaneous statistics provide us with details on triggers, user-defined functions, stored procedures and more
- RIDLIST failures can have a significant impact on query performance and warrant further investigation

DB2QM – View Historical Data



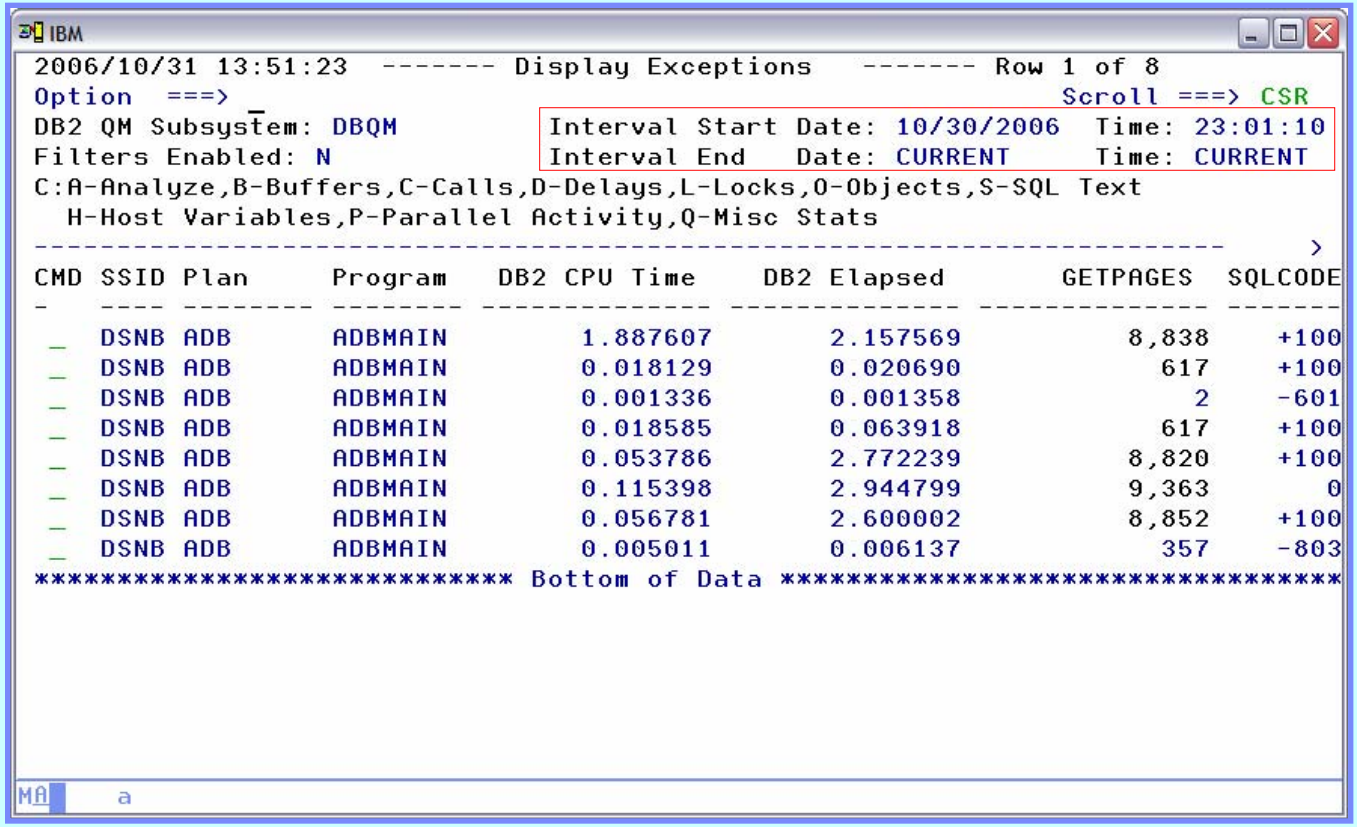
- Historical data may be accessed by using PF4/PF6 or the 'INTV' primary command

DB2QM – Interval Selection of Historical Data



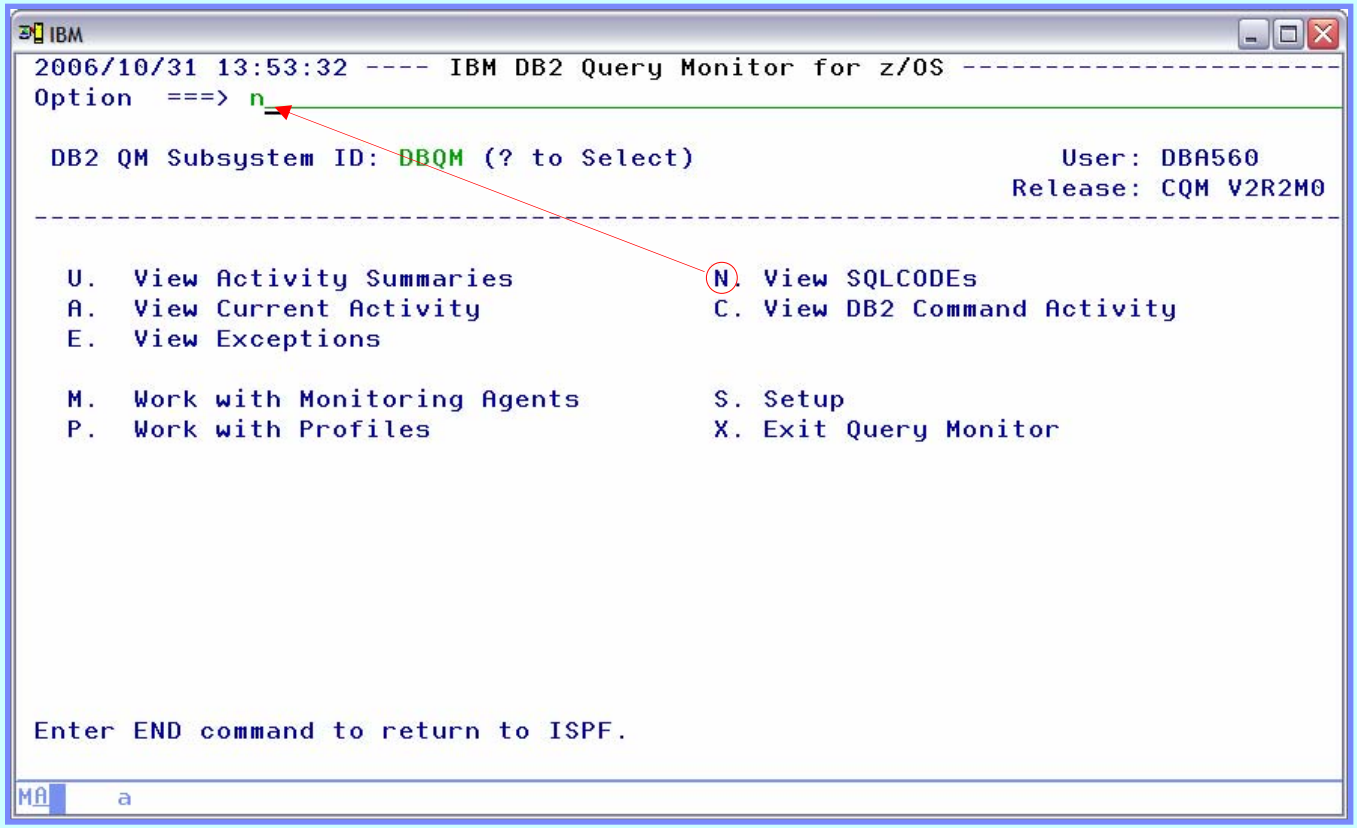
- The length of an interval and the number of intervals to keep is controlled by a start-up parameter
- Multiple intervals may be selected for viewing
- Data from the underlying VSAM data sets may be loaded into DB2 tables for analysis using SQL

DB2QM – Multiple Intervals Displayed



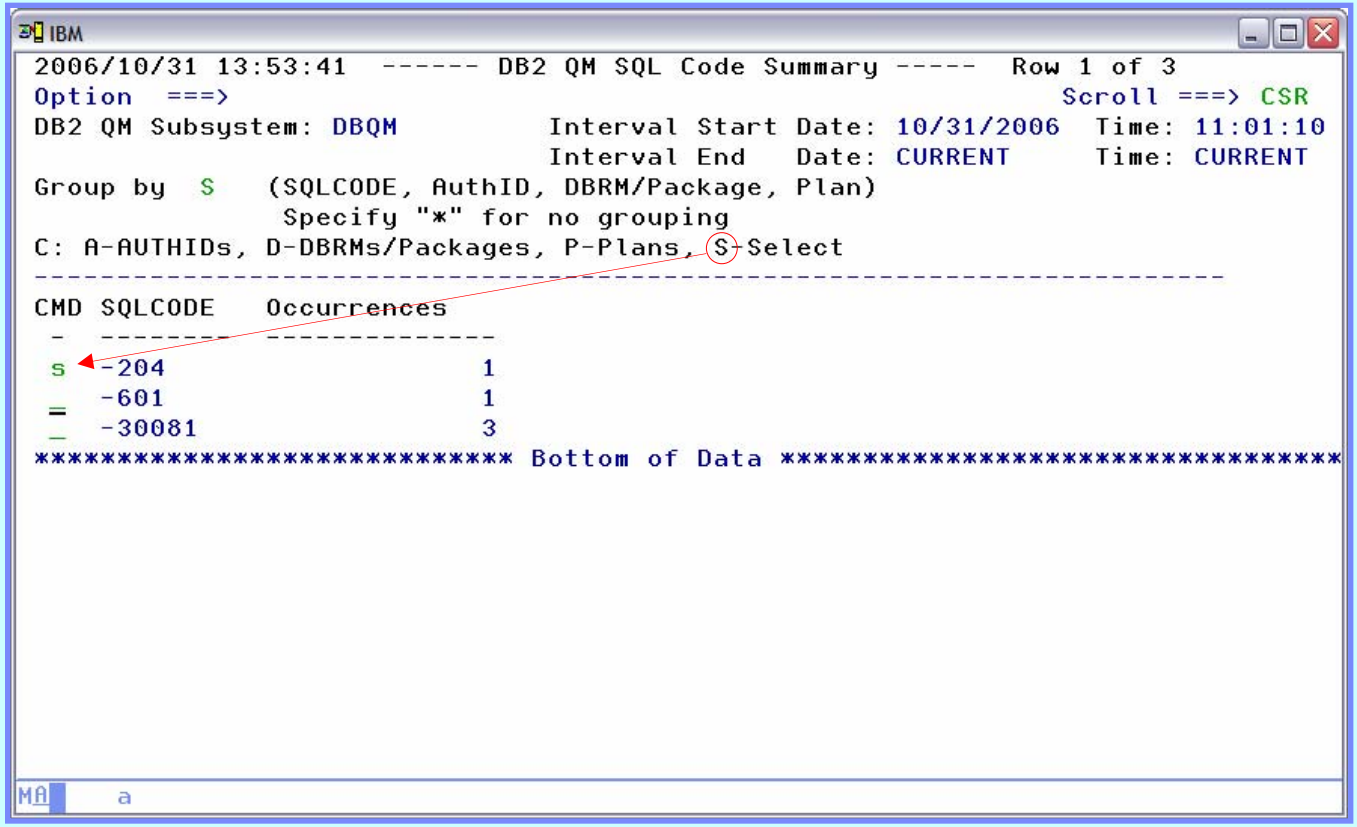
- We're now seeing data for the current and 3 previous intervals
- The 'CURRENT' primary command may be used to return to the current interval data

DB2QM – View Negative SQL Codes



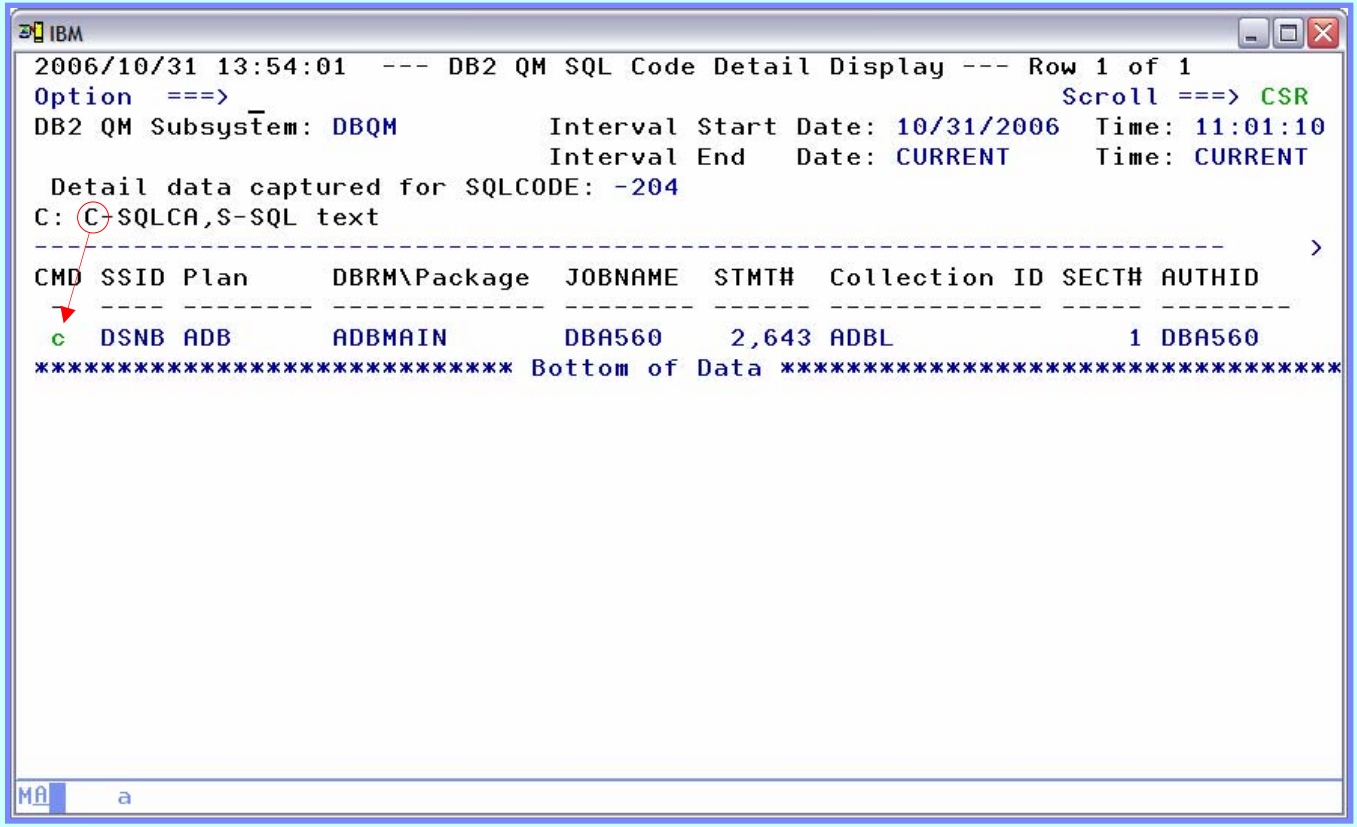
- Negative SQL codes may represent significant wasted resources and should be investigated
- Use option 'N' to view negative SQL codes
- The monitoring profile may be used to exclude the collection of SQL codes which represent common programming techniques – e.g., Singleton select (-811)

DB2QM – View SQL Code Details



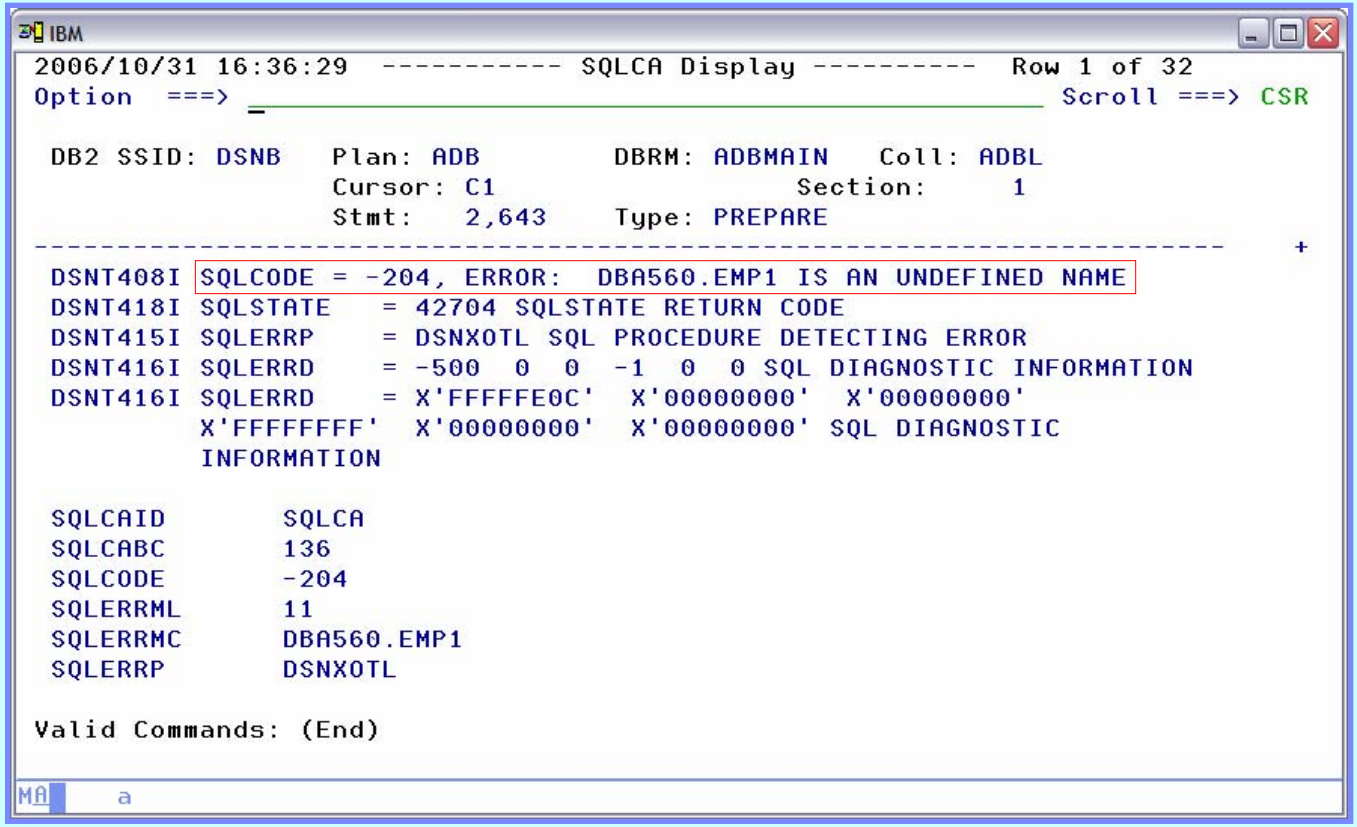
- Use ‘Group by’ option to view by code, user, etc.
- Start-up parameters limit the number of discrete codes captured per interval as well as the number of occurrences per code
- An ‘*’ next to the occurrence count indicates the limit for capturing has been exceeded

DB2QM – View SQLCA



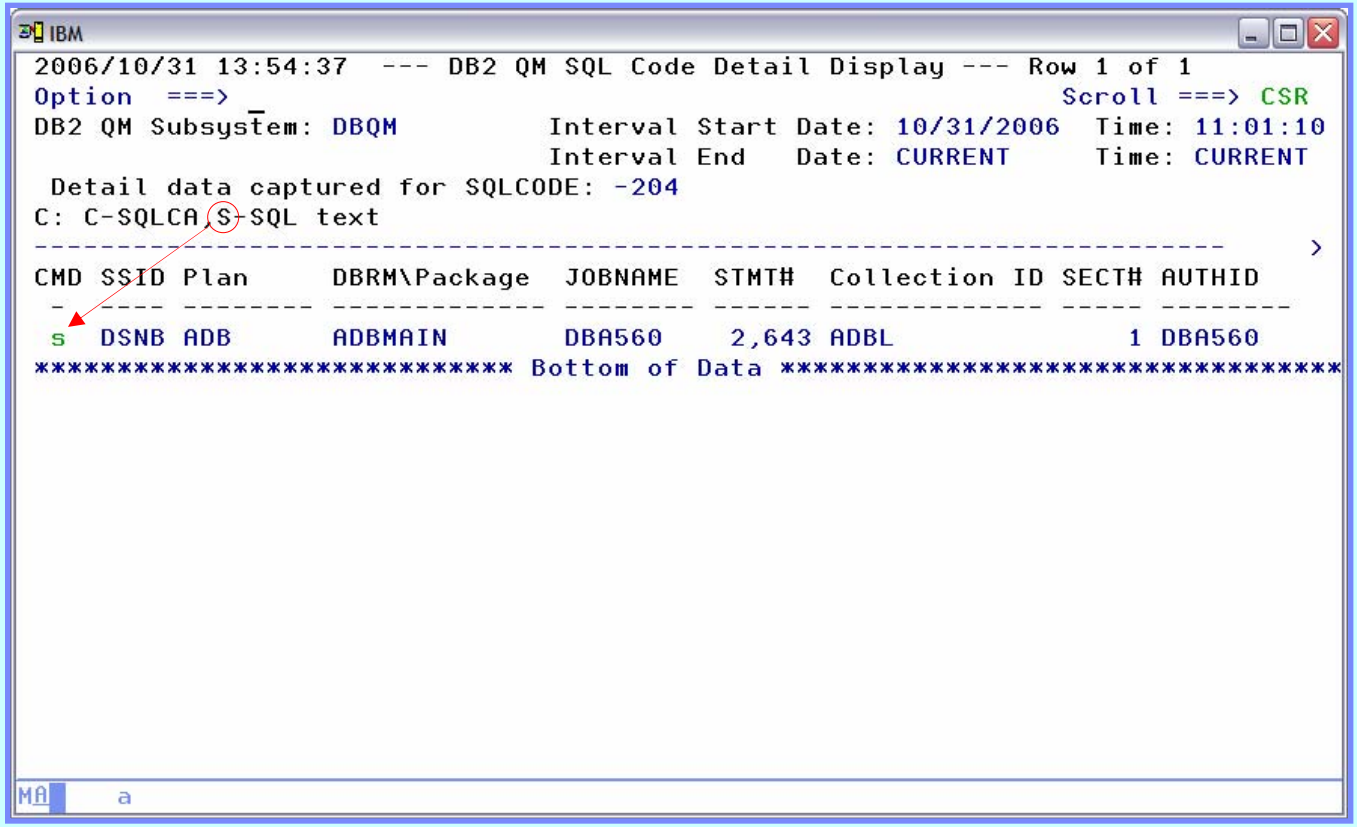
- Use the ‘C’ line command to view the SQL communications area

DB2QM – SQLCA Display



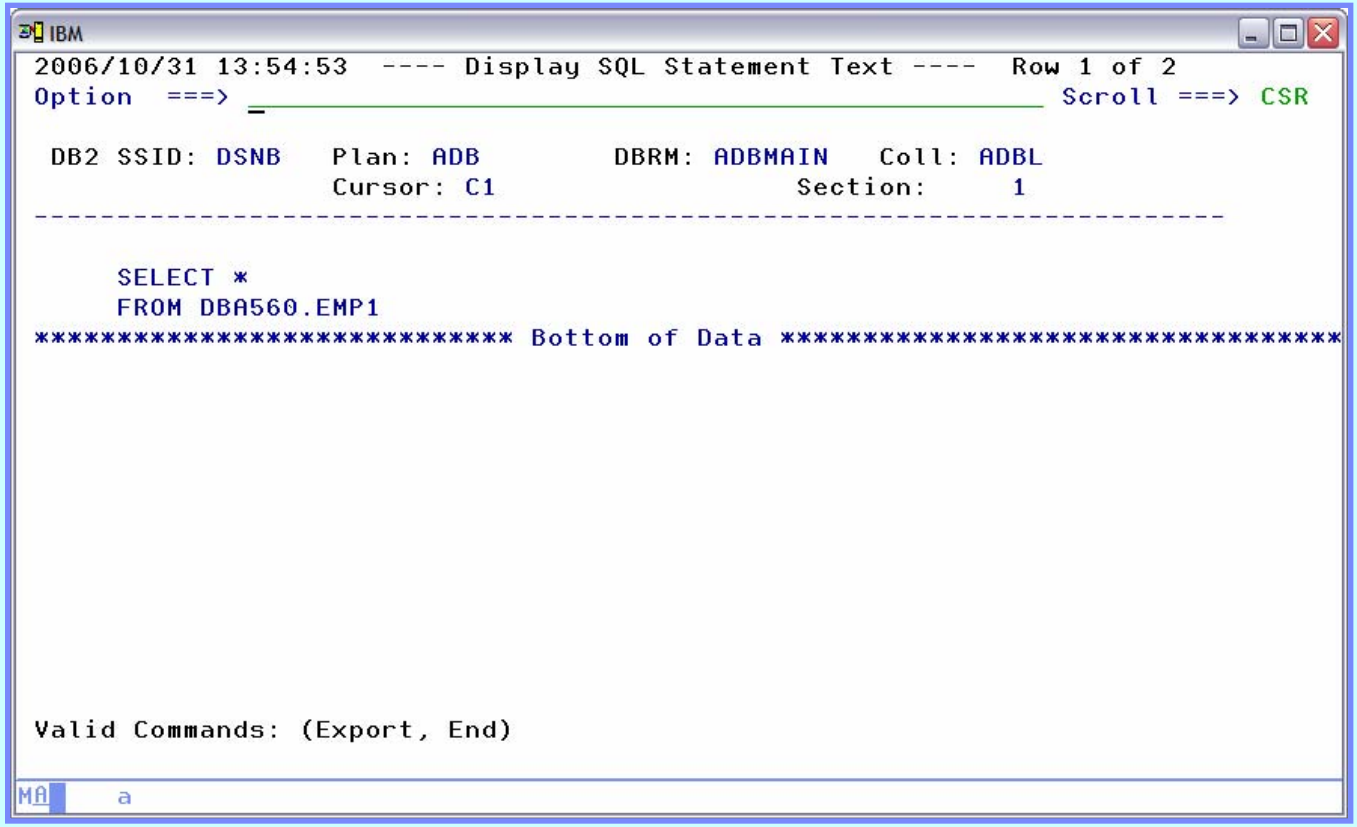
- Examine the SQL error for potential training topics

DB2QM – View SQL Text



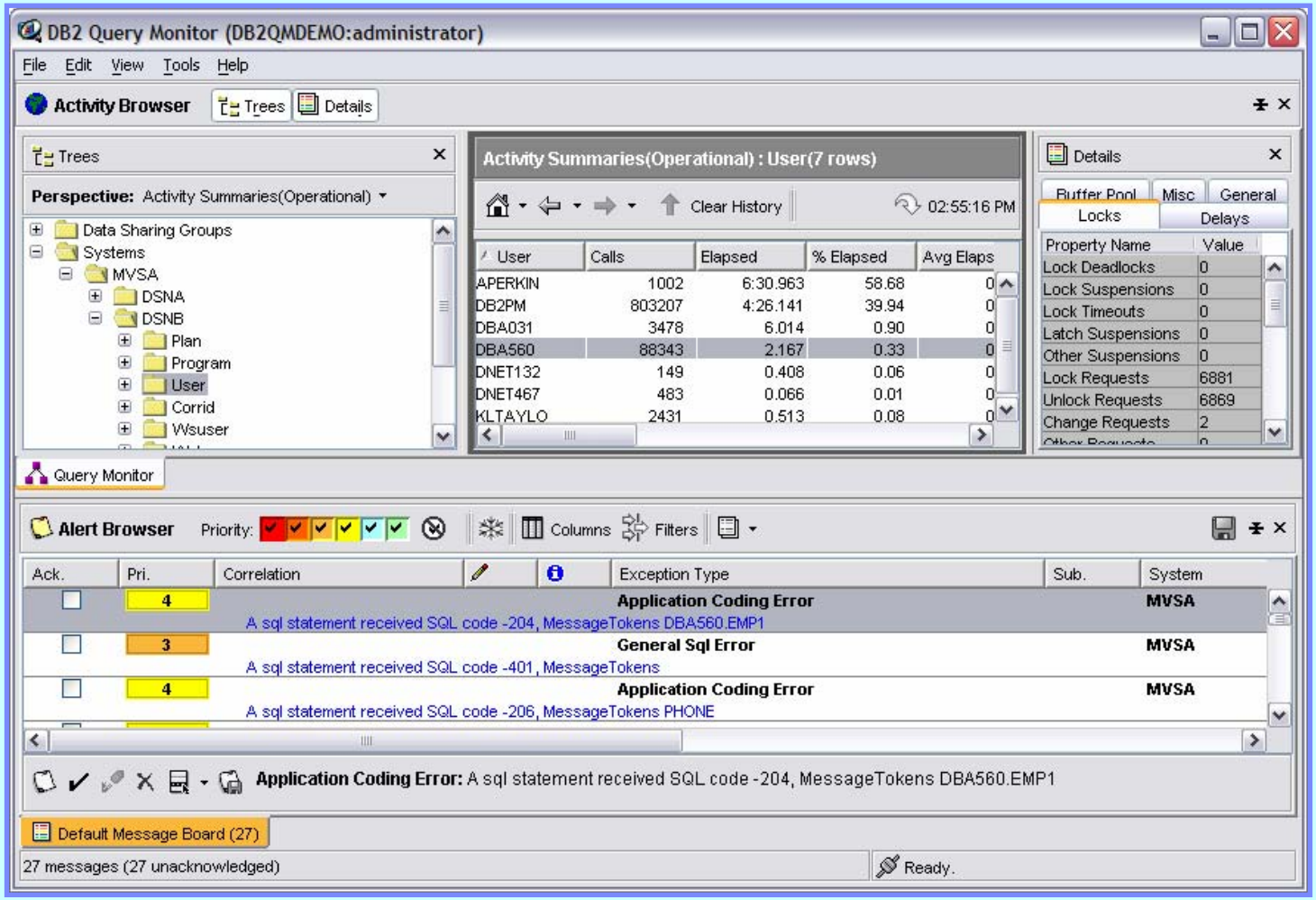
- Use the ‘S’ line command to view the SQL text

DB2QM – SQL Text Display



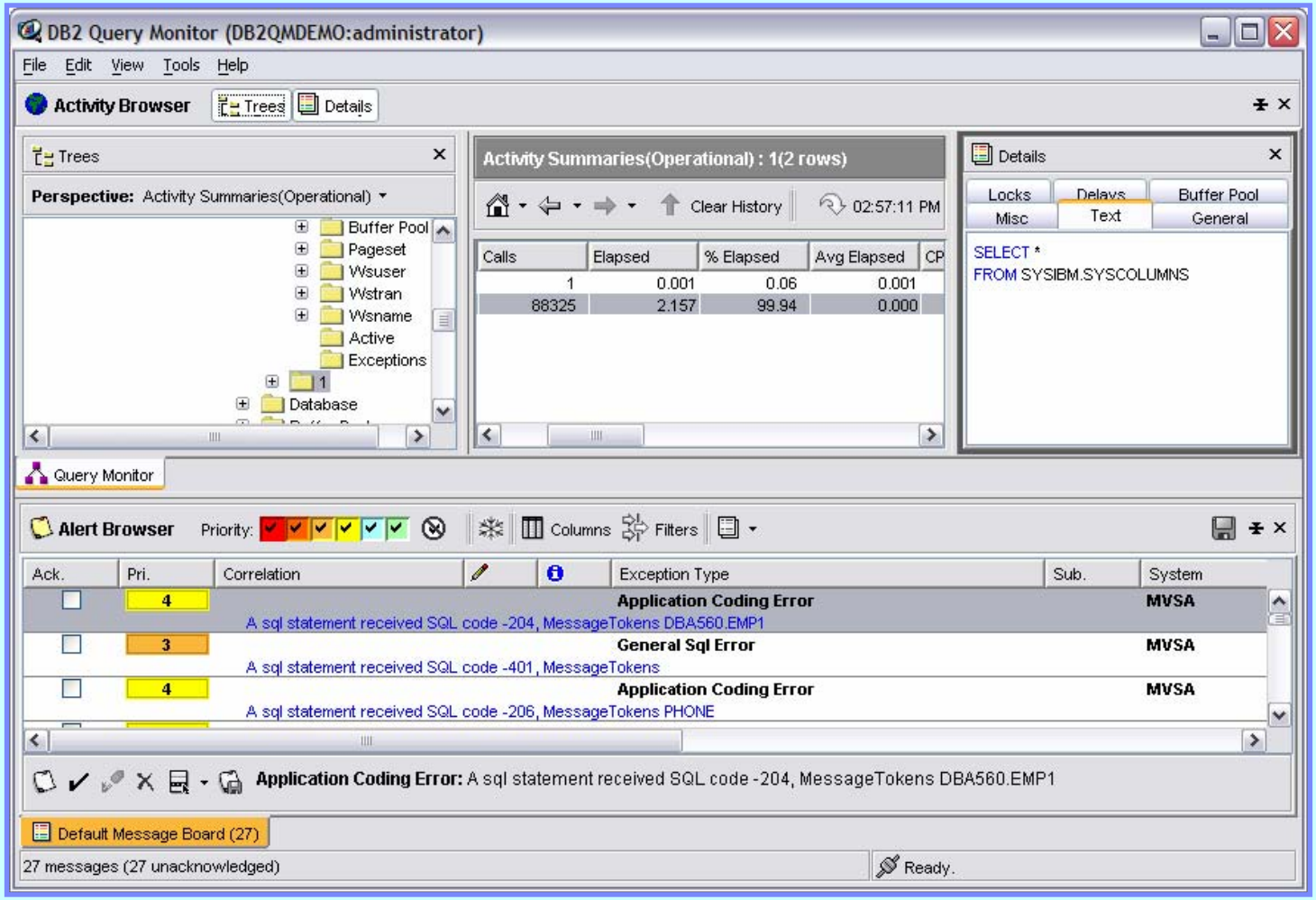
- Examine the SQL text for potential training topics

DB2QM – Java Client (1)



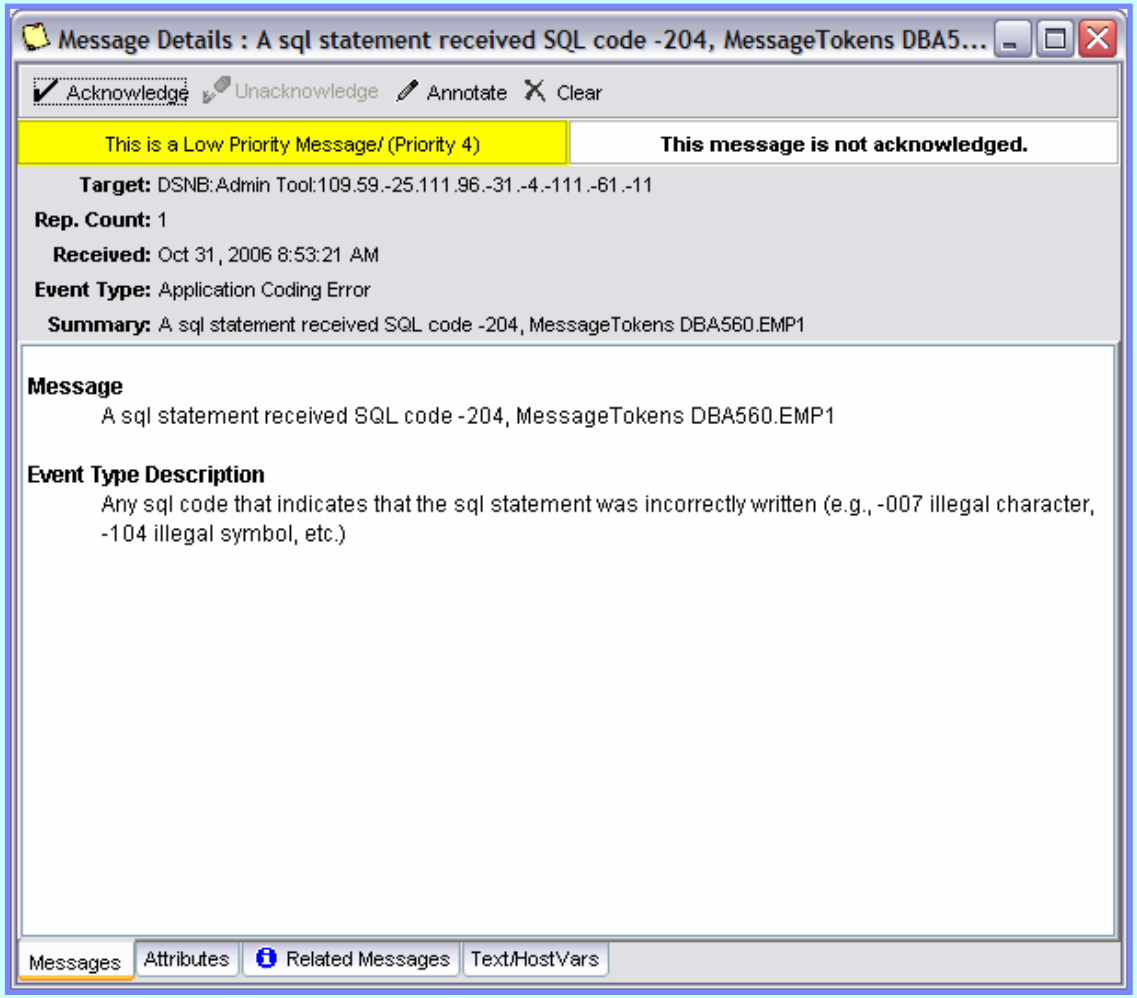
- The Java client may be used as an alternative to the ISPF interface
- The Java client offers a message board for alerts
- Data from the various windows may be saved to a file on the workstation

DB2QM – Java Client (2)



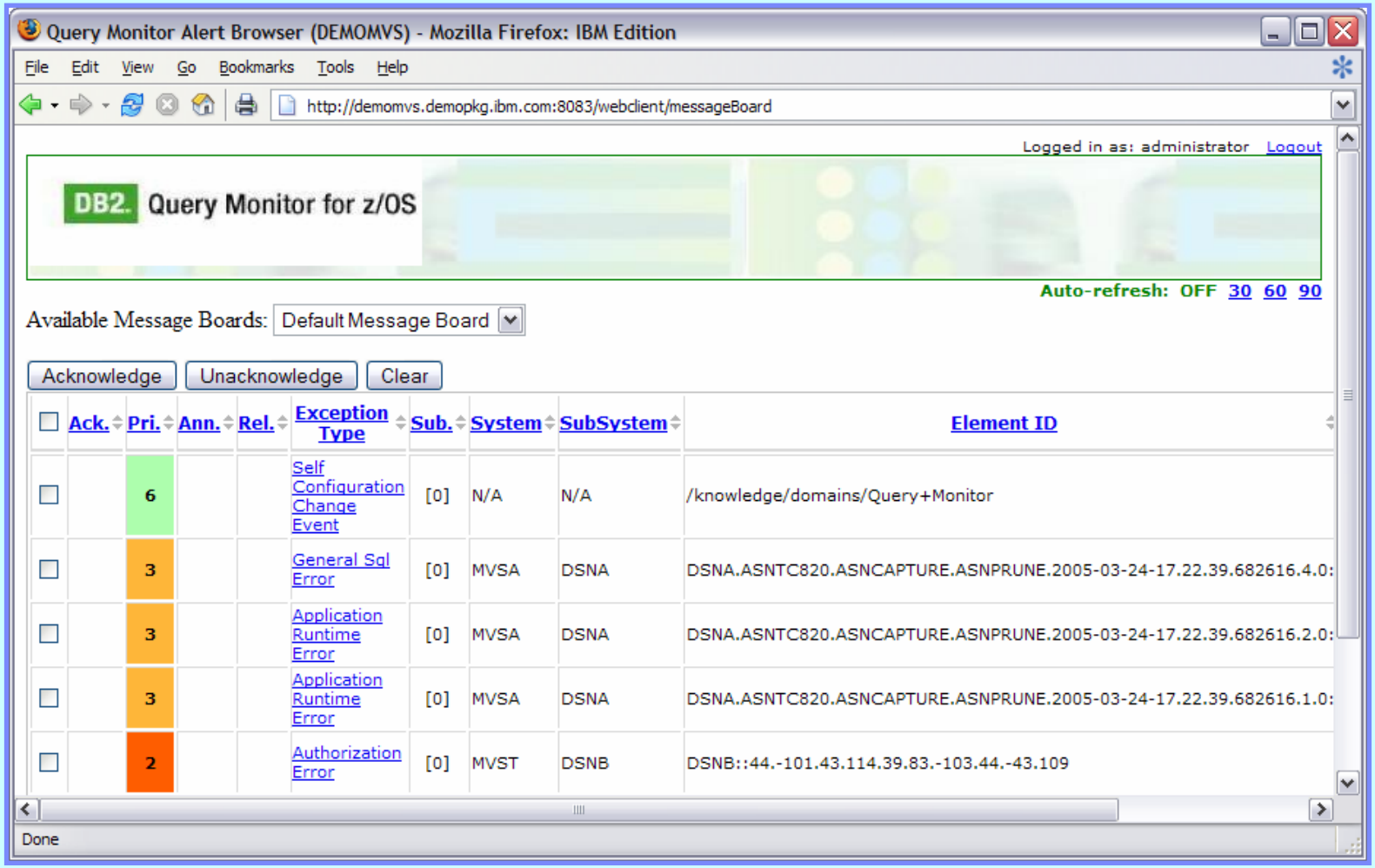
- The data we saw earlier related to an exception is displayed at the top of the interface
- The negative SQL error we investigated is displayed in the message board

DB2QM – Java Client / SQL Error Details



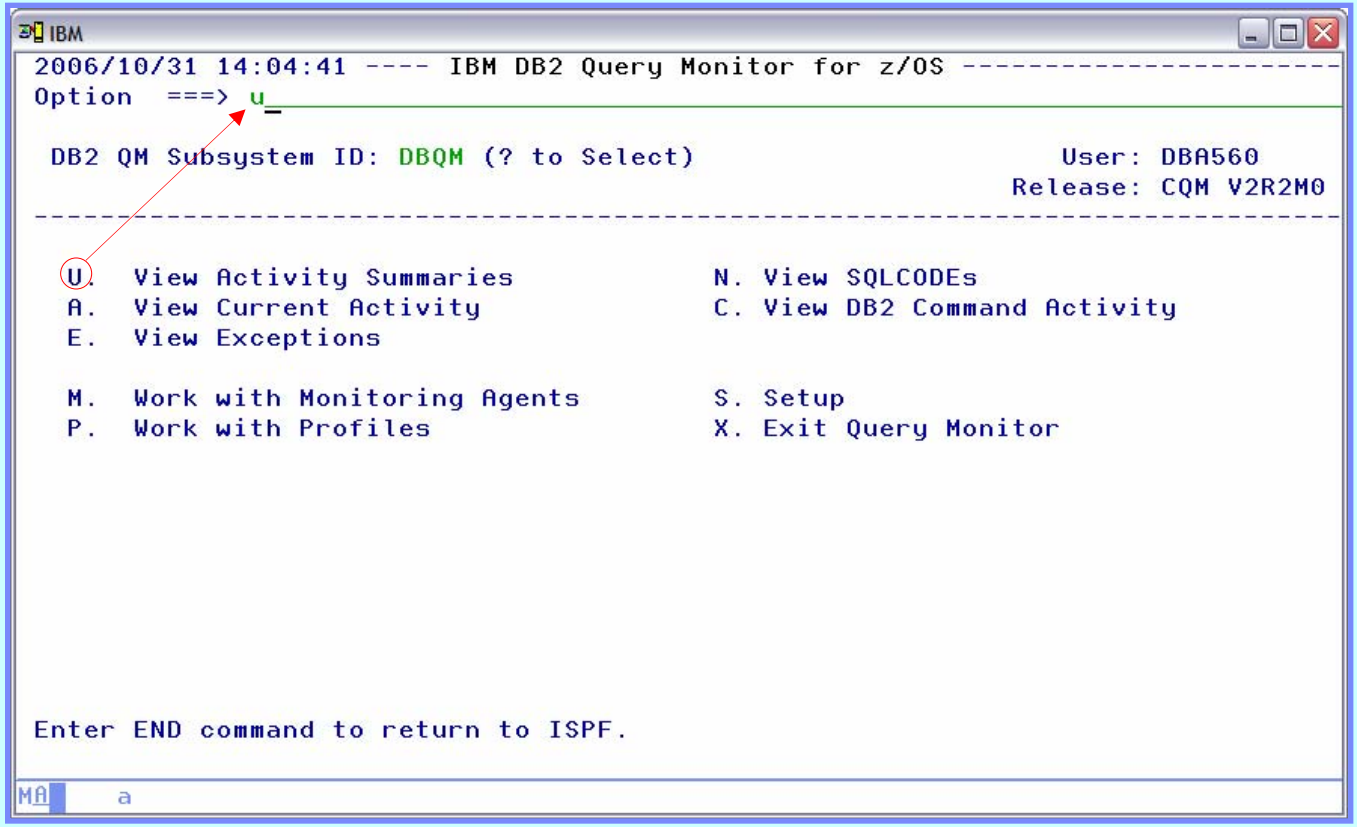
- Double-clicking on an event in the message board opens up a detail window – notice the tabs at the bottom for more details

DB2QM – Web-based Message Board



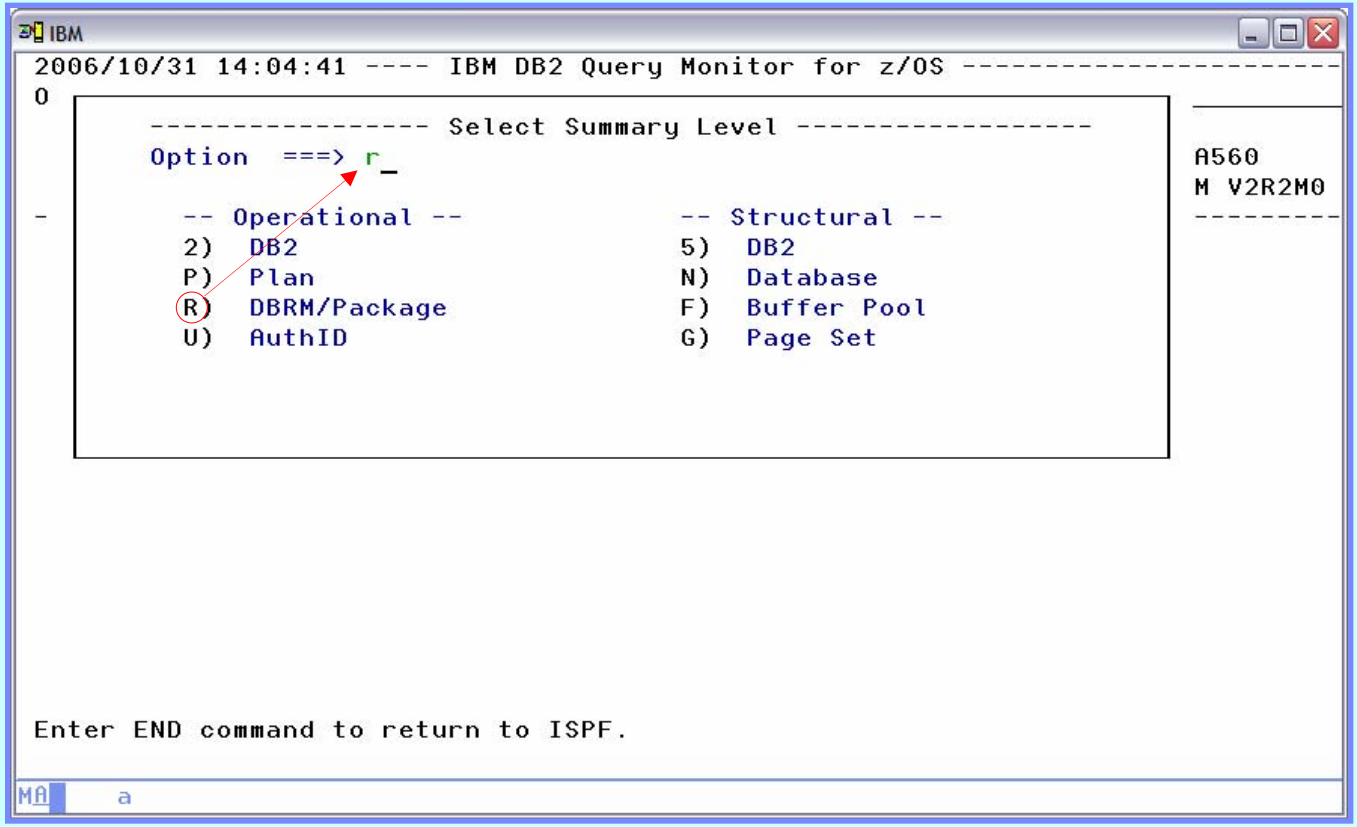
- The message board may be accessed via a web browser
- Alerts go to the message board – exceptions do not
- Clicking on a hyperlink will display the details for the event
- Notice the auto-refresh option at the top right

DB2QM – View Summarized Data



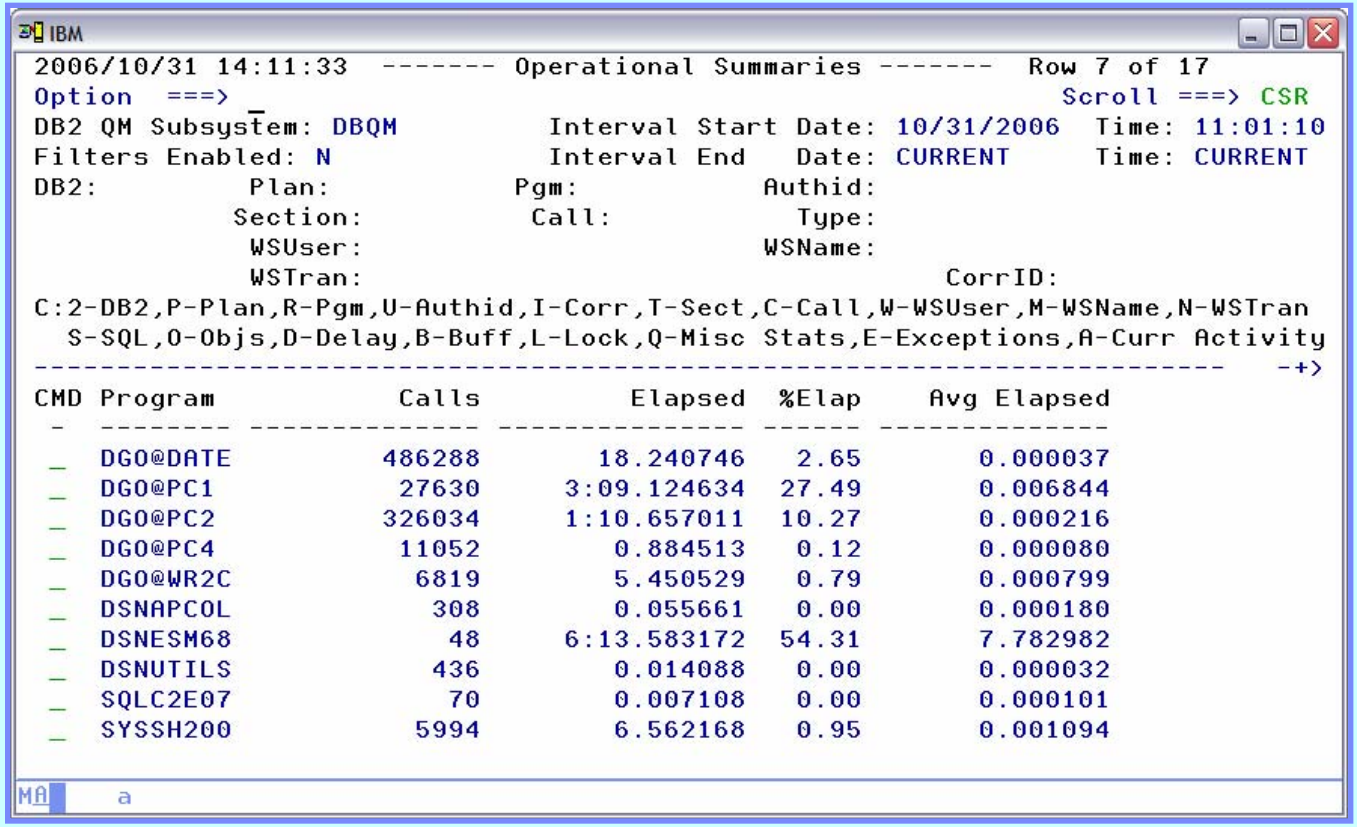
- Summarized data does not show individual SQL statement executions, but may be a good way to find SQL statements that use a high amount of resources

DB2QM – Select Summary Level



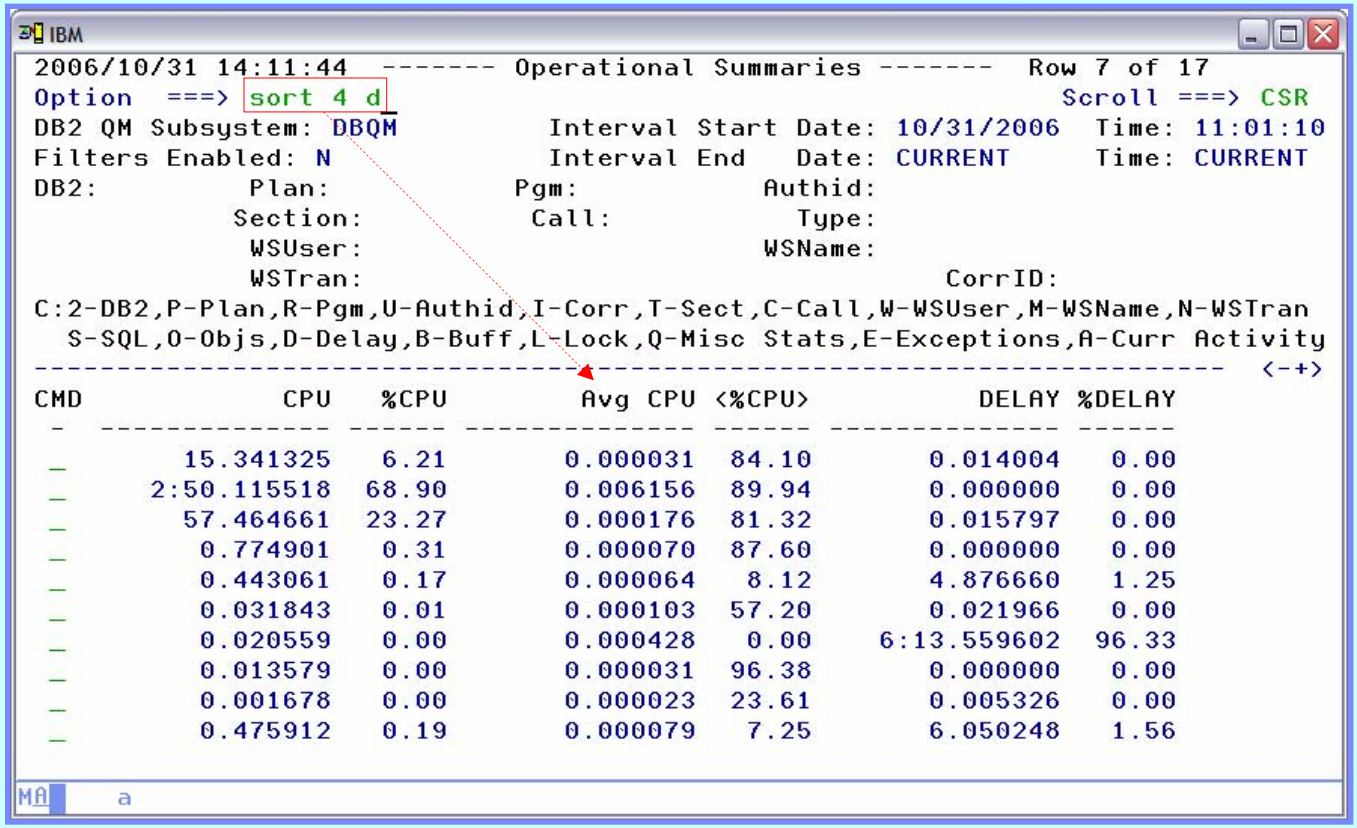
- Query Monitor summarizes data by subsystem, plan and package
- Summarization by AUTHID as well as object data collection is controlled by start-up and profile settings

DB2QM – Package Summary Display



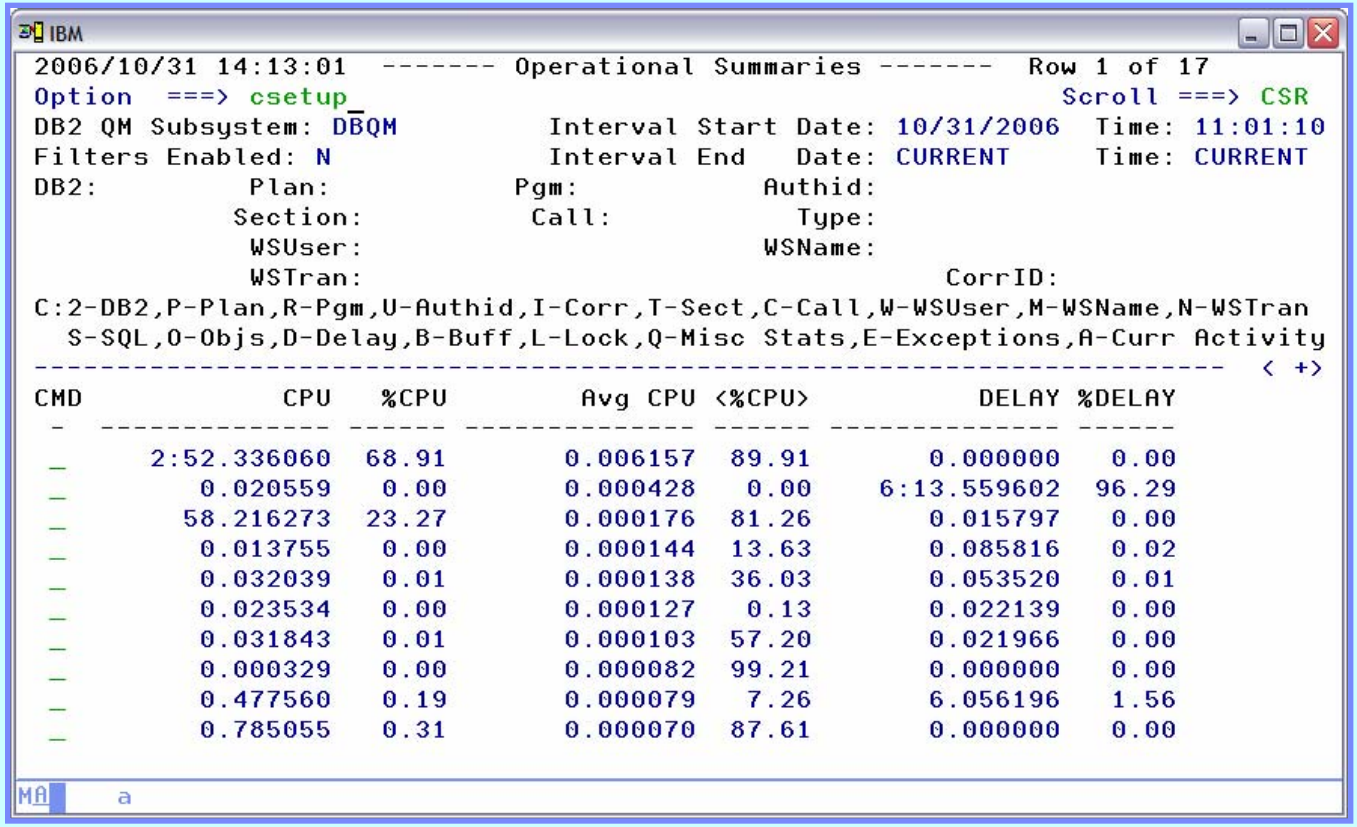
- SQL text is available for summarized data
- Individual executions of an SQL statement cannot be seen in summarized data

DB2QM – Sort Displayed Data



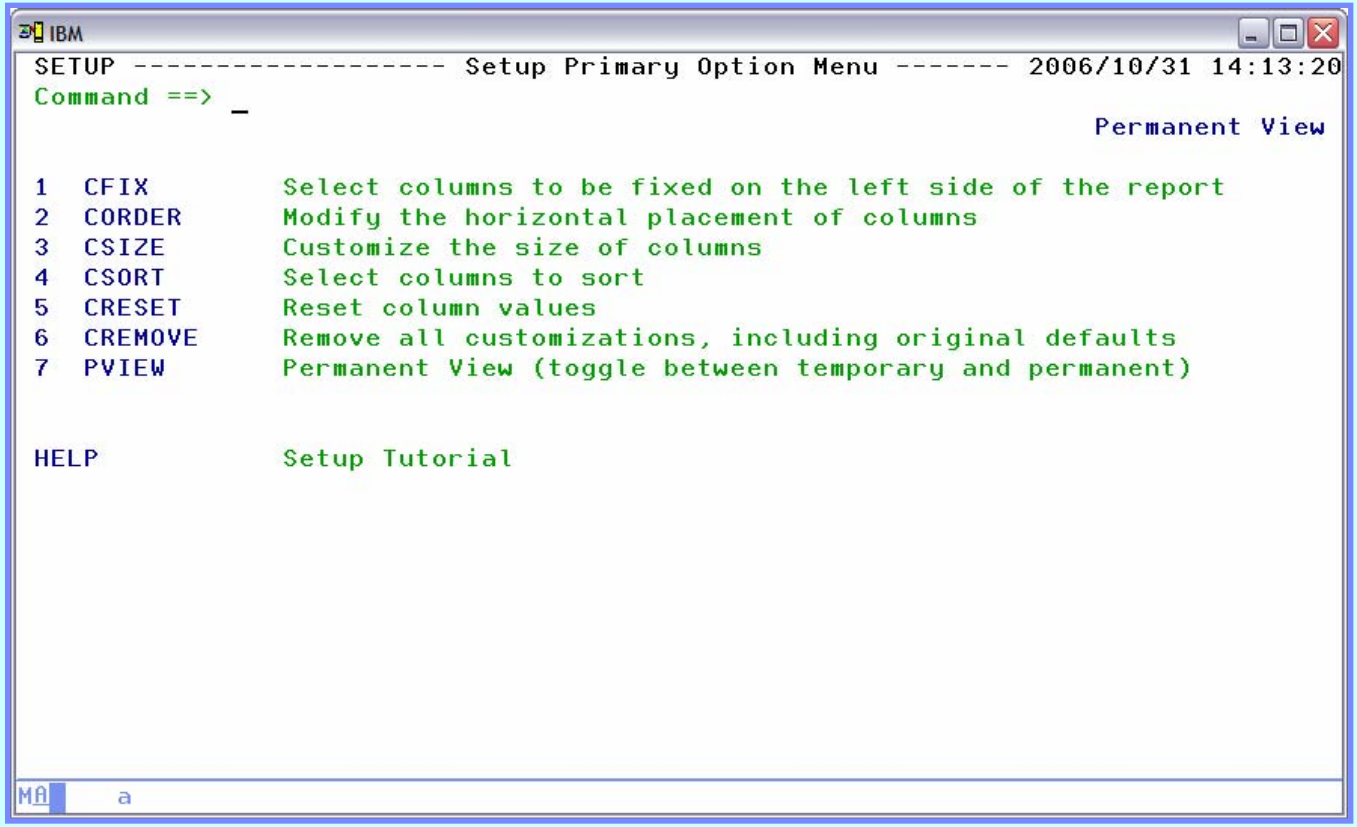
- Any panel may be sorted using the ‘SORT’ or ‘CSORT’ primary commands
- Sorting is based on column number – the ‘CMD’ column is column number 1
- Use ‘CNUM’ to toggle on/off column numbers

DB2QM – Manage Display Characteristics



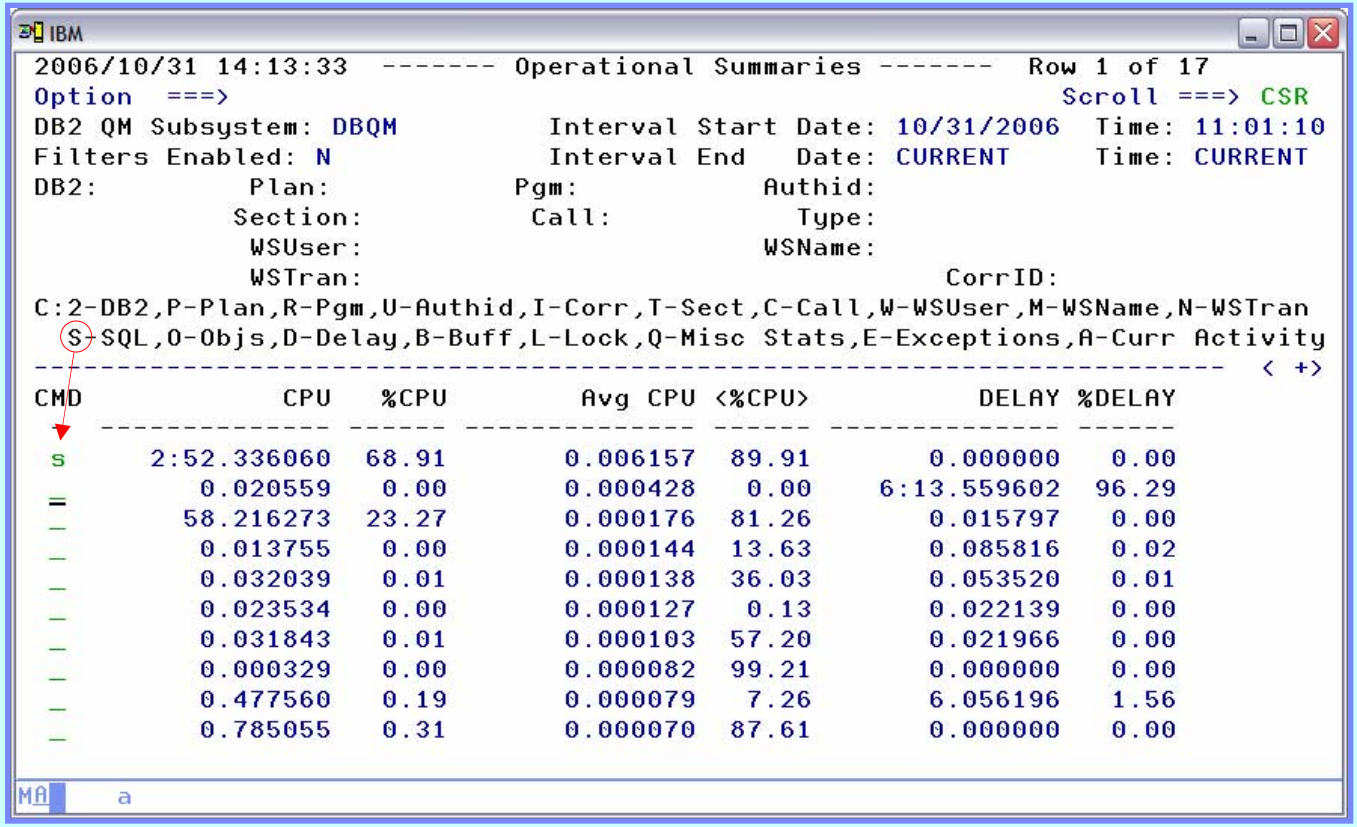
- Use ‘CSETUP’ to change display attributes

DB2QM – Setup Options Menu



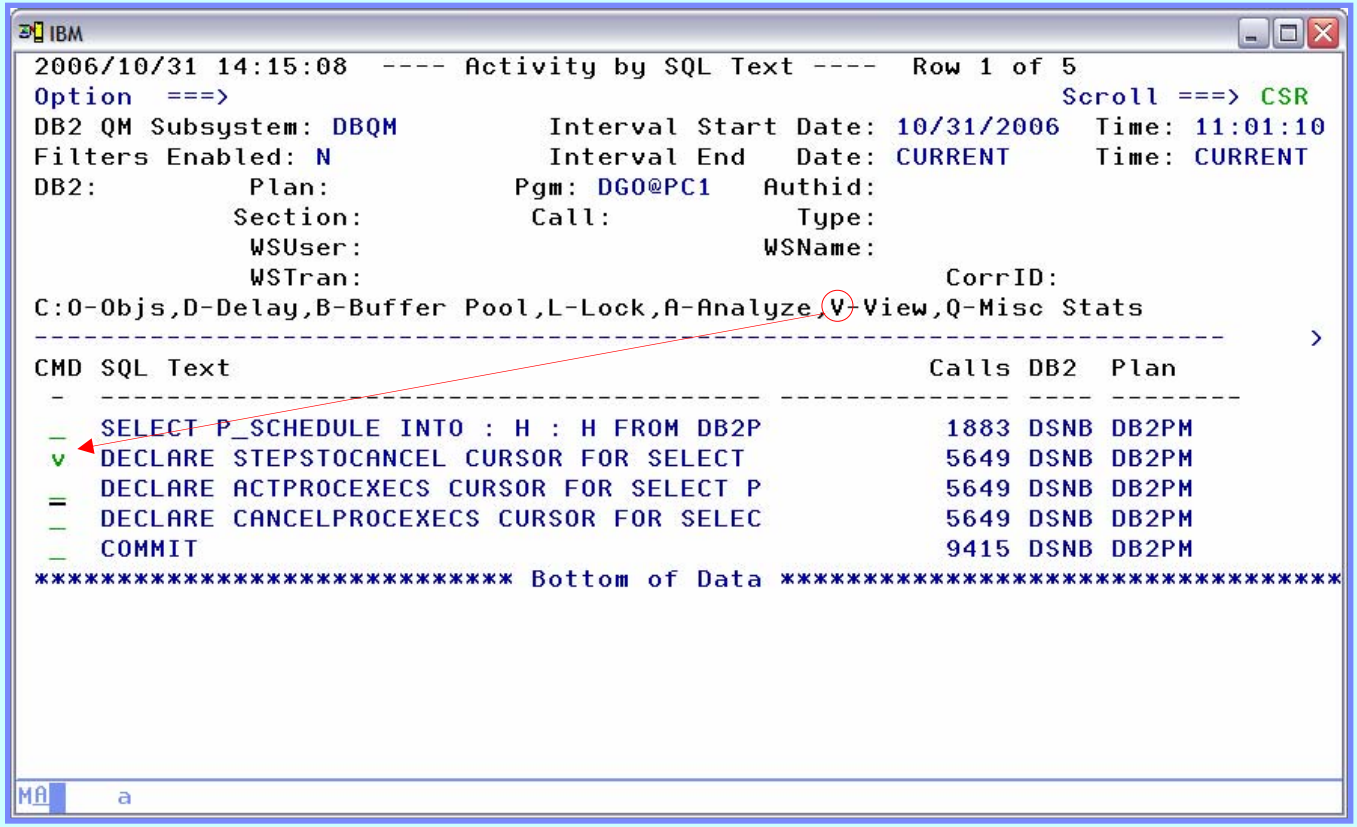
- Options are stored in each users ISPF profile data set
- Commands listed on this panel may be used as primary commands without first issuing 'CSETUP'

DB2QM – View Package SQL Text



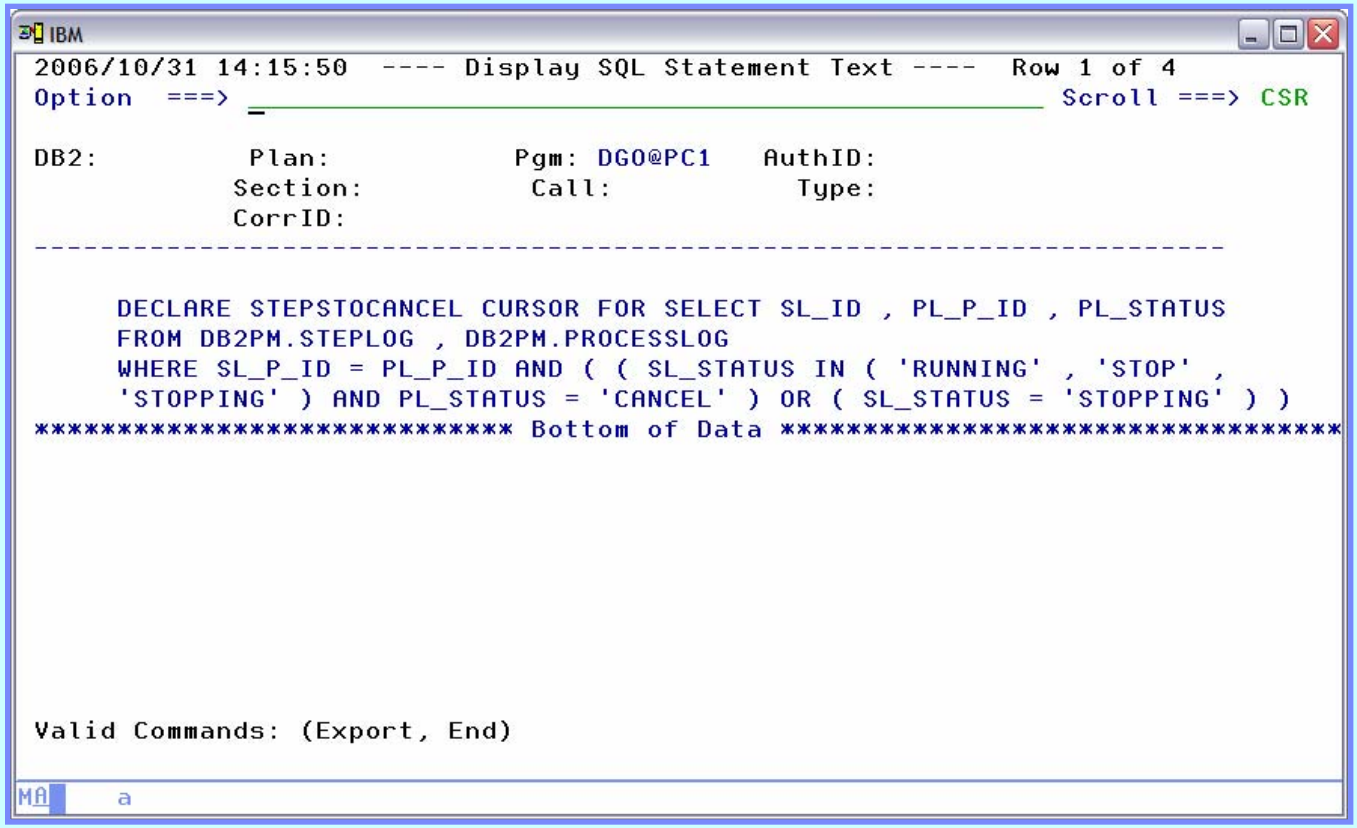
- We've sorted the package summary to bring the highest average user of CPU to the top – we'll use the 'S' line command to view the SQL text

DB2QM – View Full SQL Text



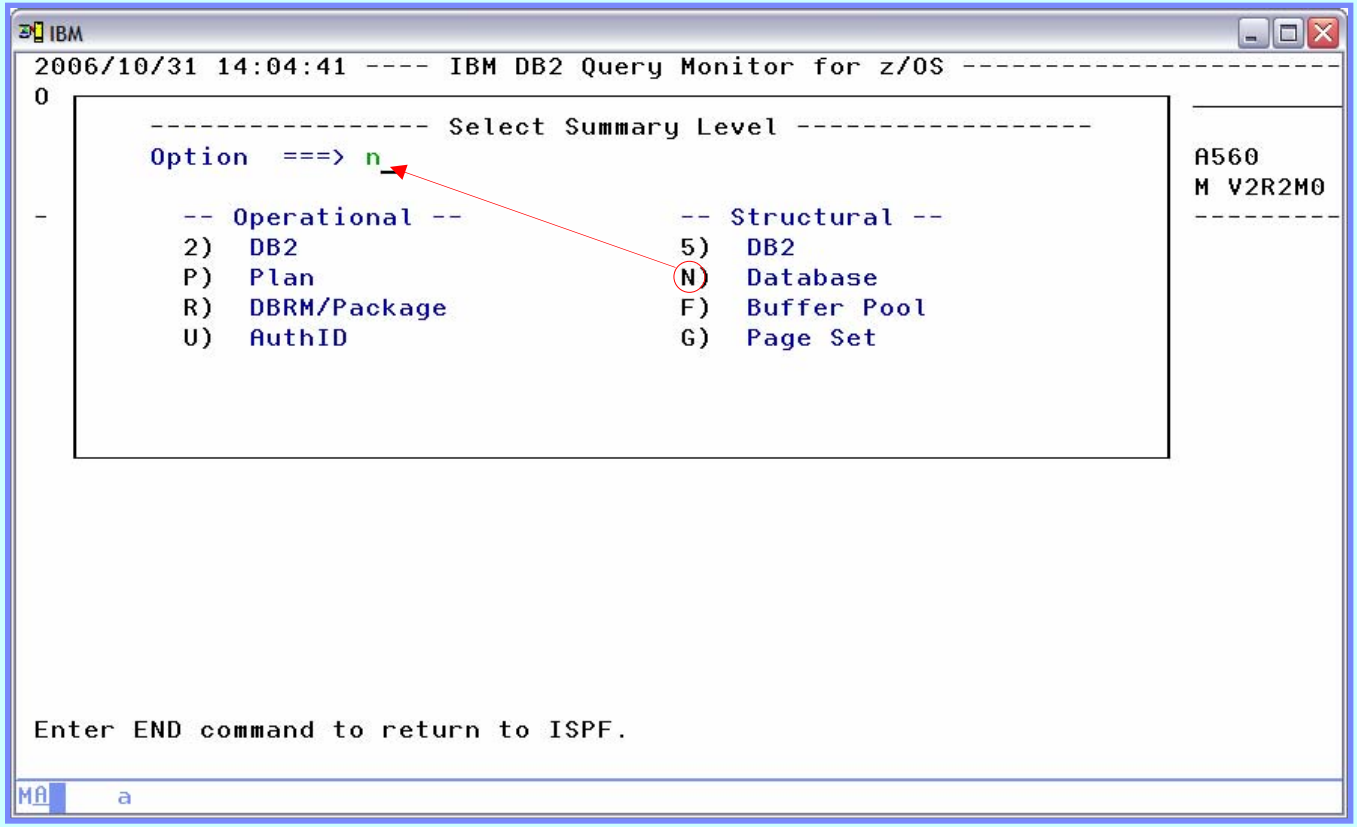
- Use the ‘V’ line command to see the entire SQL statement text

DB2QM – Full SQL Text Display



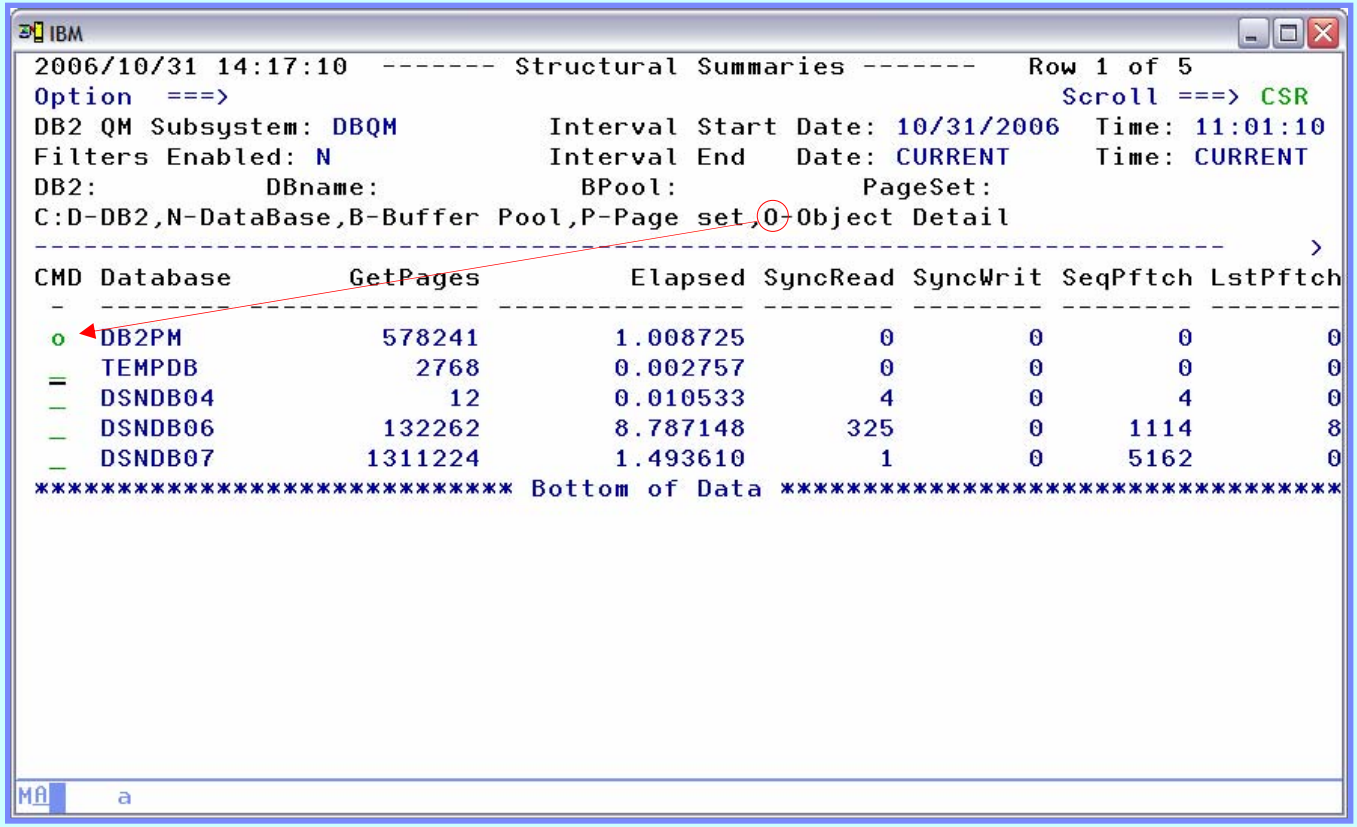
- Query Monitor will display the full 32K (V7) or 2MB (V8) of SQL text

DB2QM – View Database Summary Data



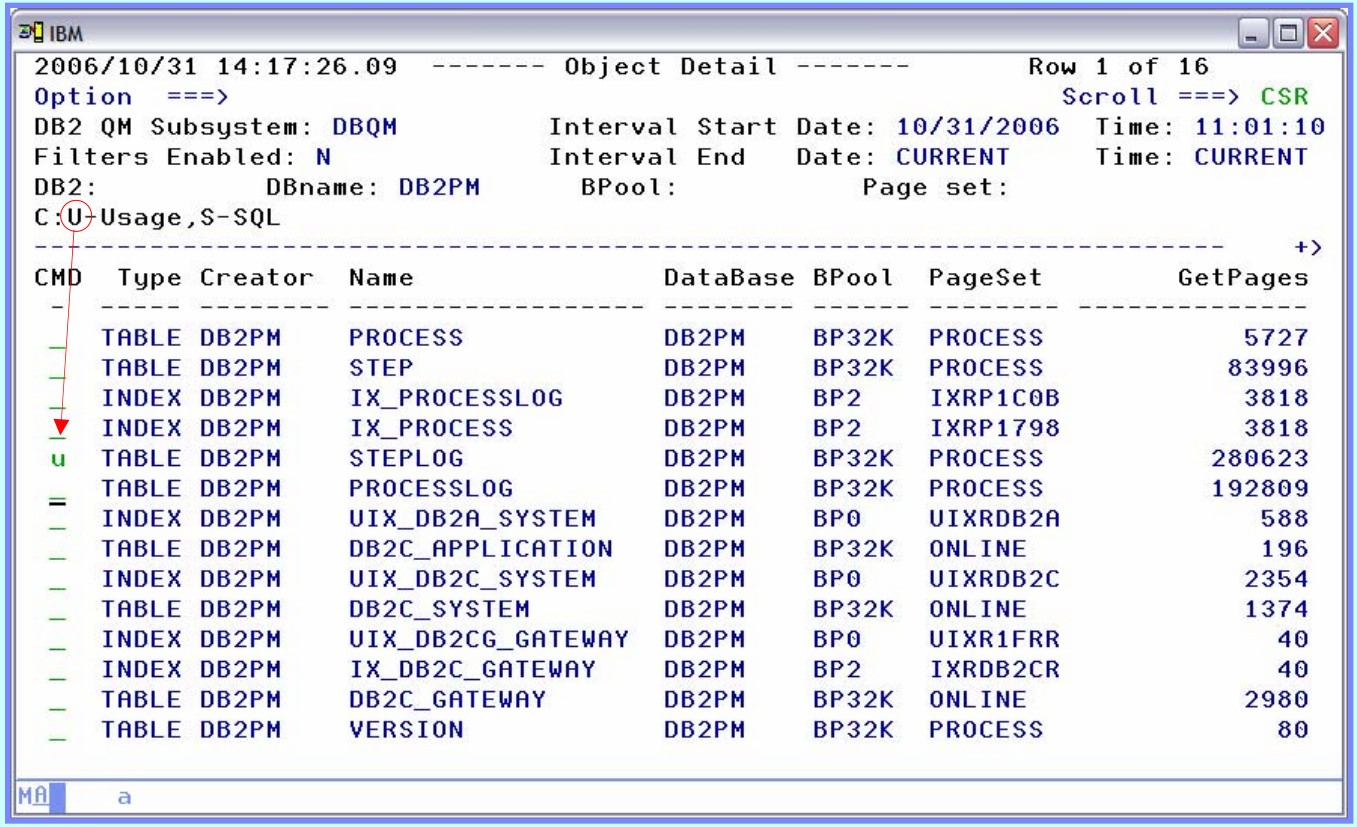
- Structural data is collected based on start-up parameters
- Structural data may be more useful for tuning at the subsystem level, rather than the SQL statement level

DB2QM – View Object Detail



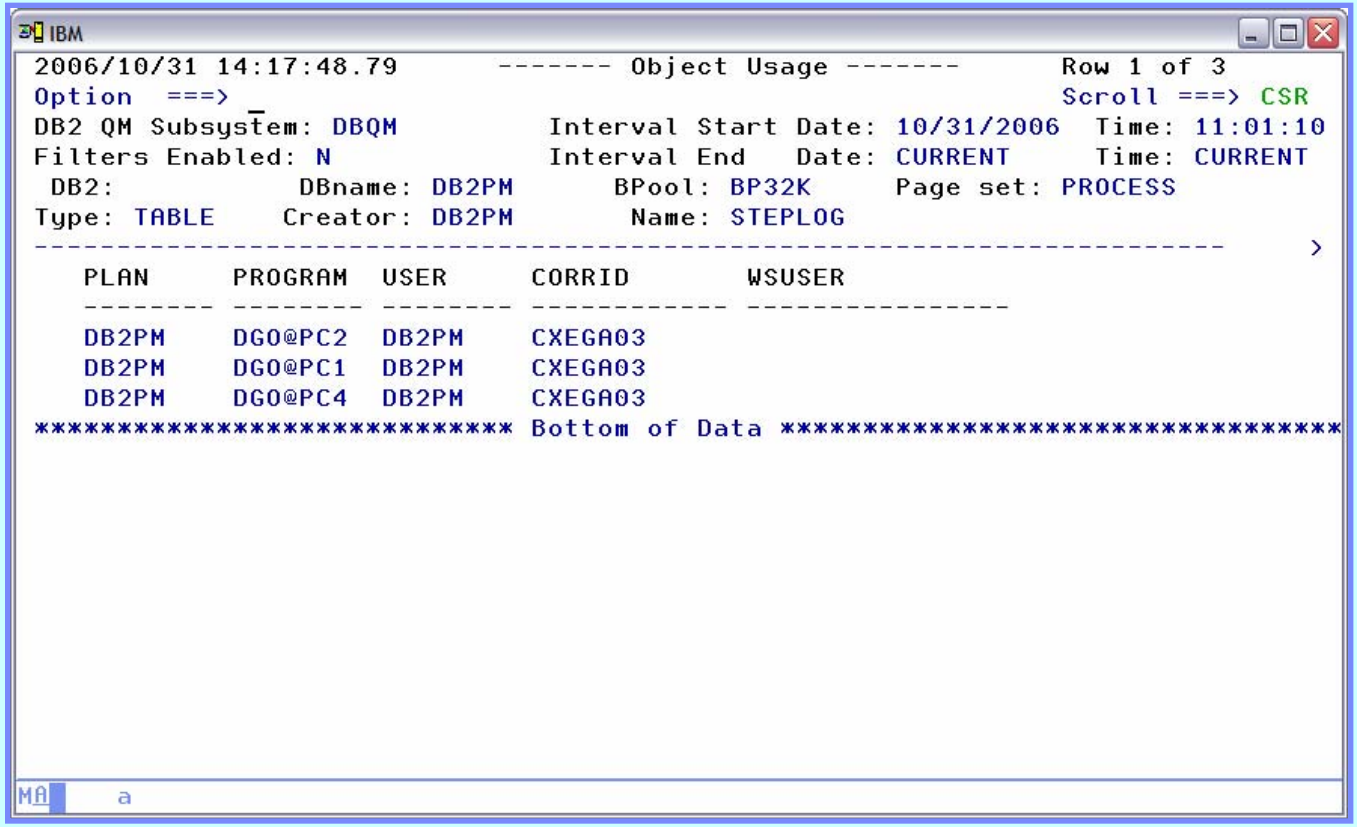
- At the object level we can drill down to get details on buffer pool, page set, and object details

DB2QM – View Object Usage



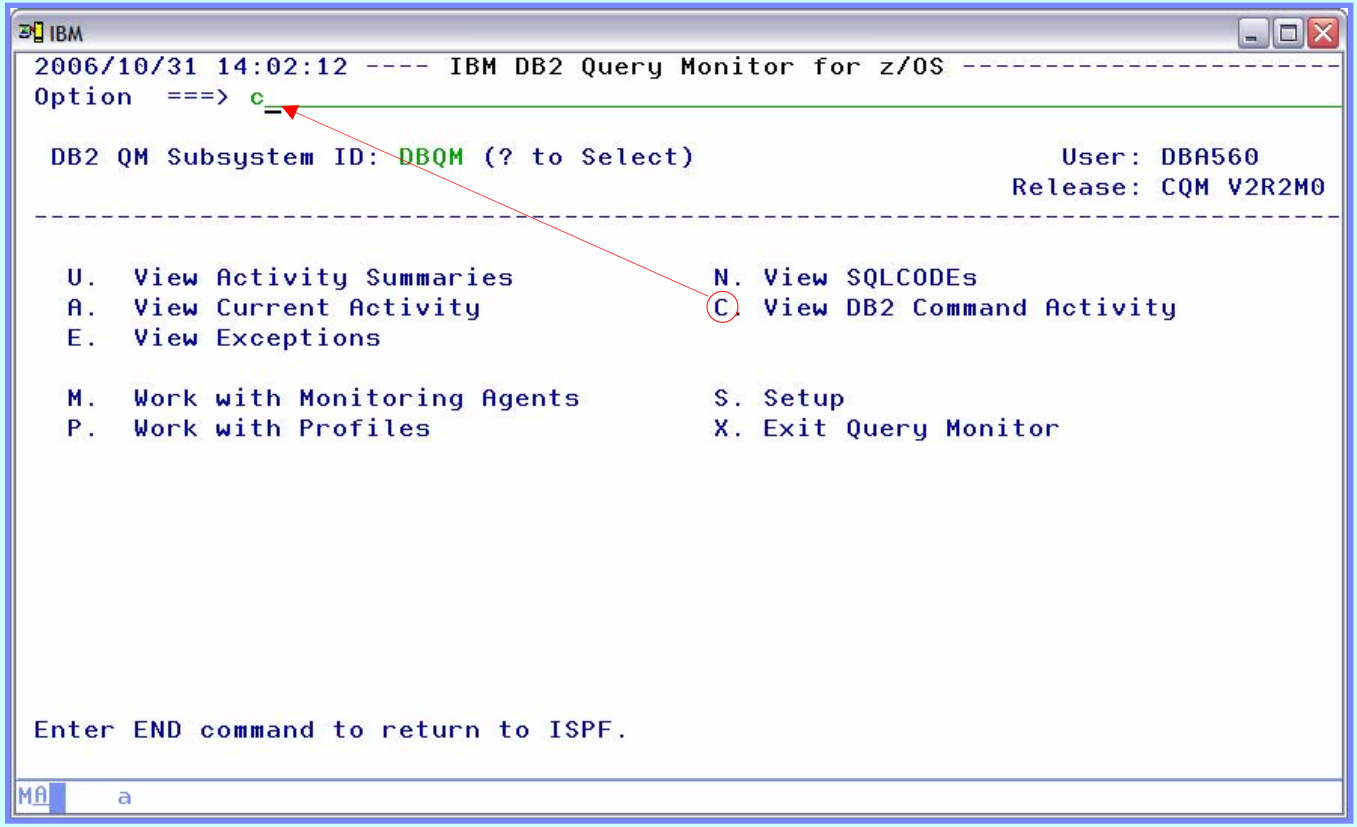
- Tables and indexes are displayed in the object detail
- Drill down is limited to object usage, and SQL accessing objects

DB2QM – Object Usage Display



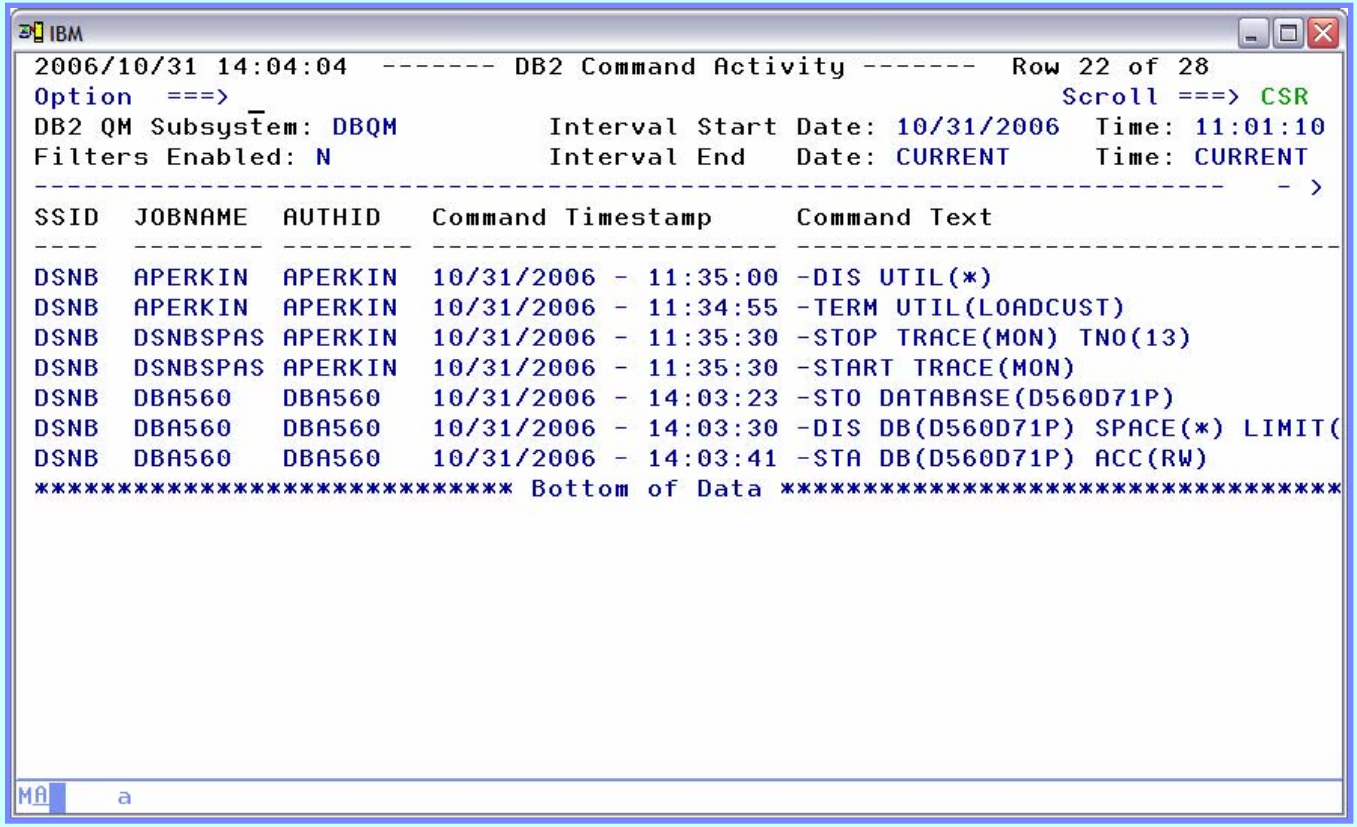
- Object usage allows us to determine the plans, packages and users that have accessed an object

DB2QM – View DB2 Command Activity



- Query Monitor captures all commands issued to DB2

DB2QM – DB2 Commands Display



- Look for commands which could have an impact on user response, or perception of response
- Start trace commands could have a significant impact on subsystem performance

Now that we know how

to find poorly performing SQL

let's see how we can make it better

IBM Software Group

Optimizing the Performance of SQL with IBM Explain Tools

EXPLAIN SOLUTIONS FROM IBM

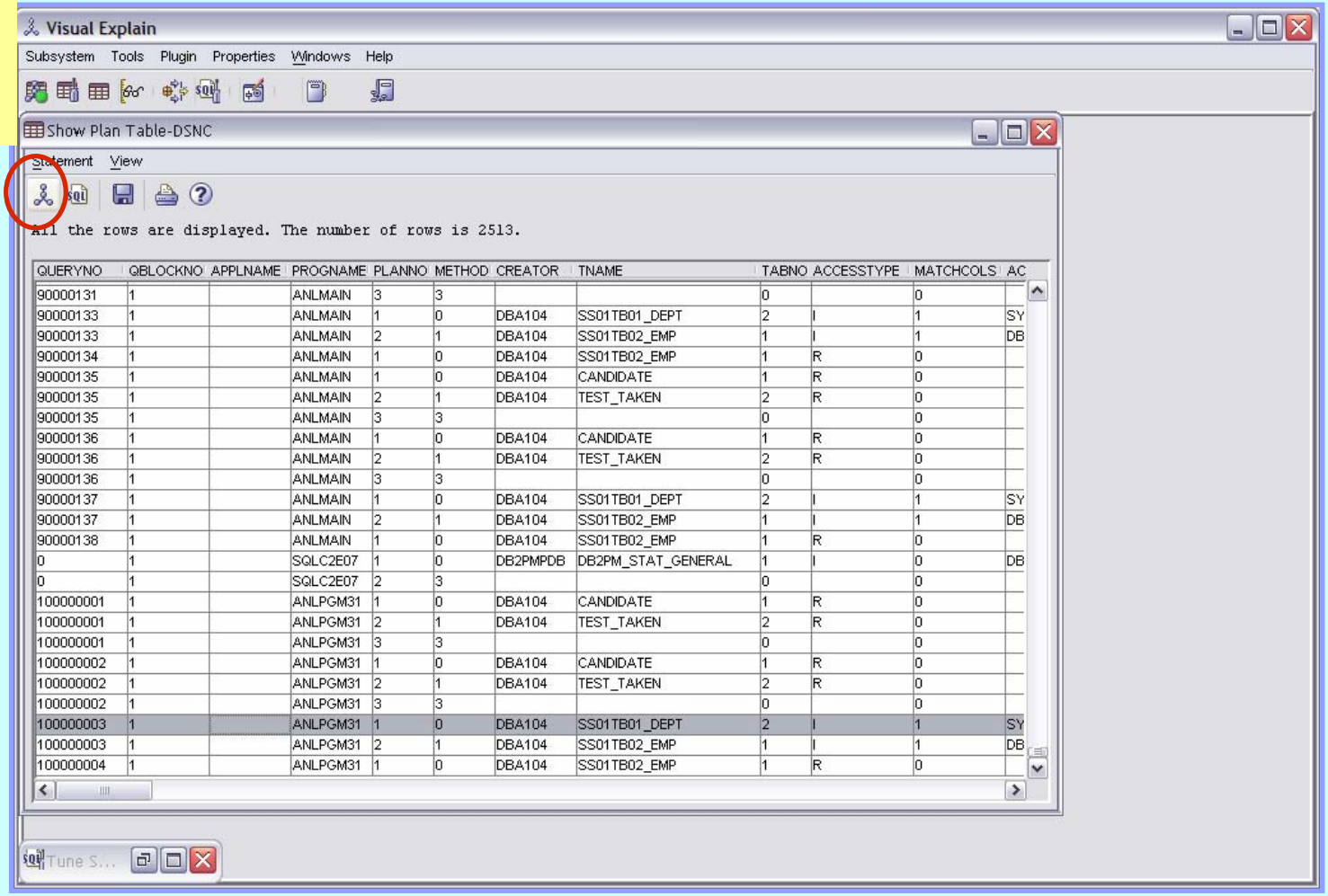
- Visual Explain
- DB2 SQL Performance Analyzer
- DB2 Path Checker

VISUAL EXPLAIN

- GUI interface for explaining SQL on various platforms including z/OS
- Feature of DB2
- Source of SQL
  - Key or cut and paste a SQL statement into the tool
  - Plan / Package
  - Existing entry in a PLAN_TABLE (retro explain)

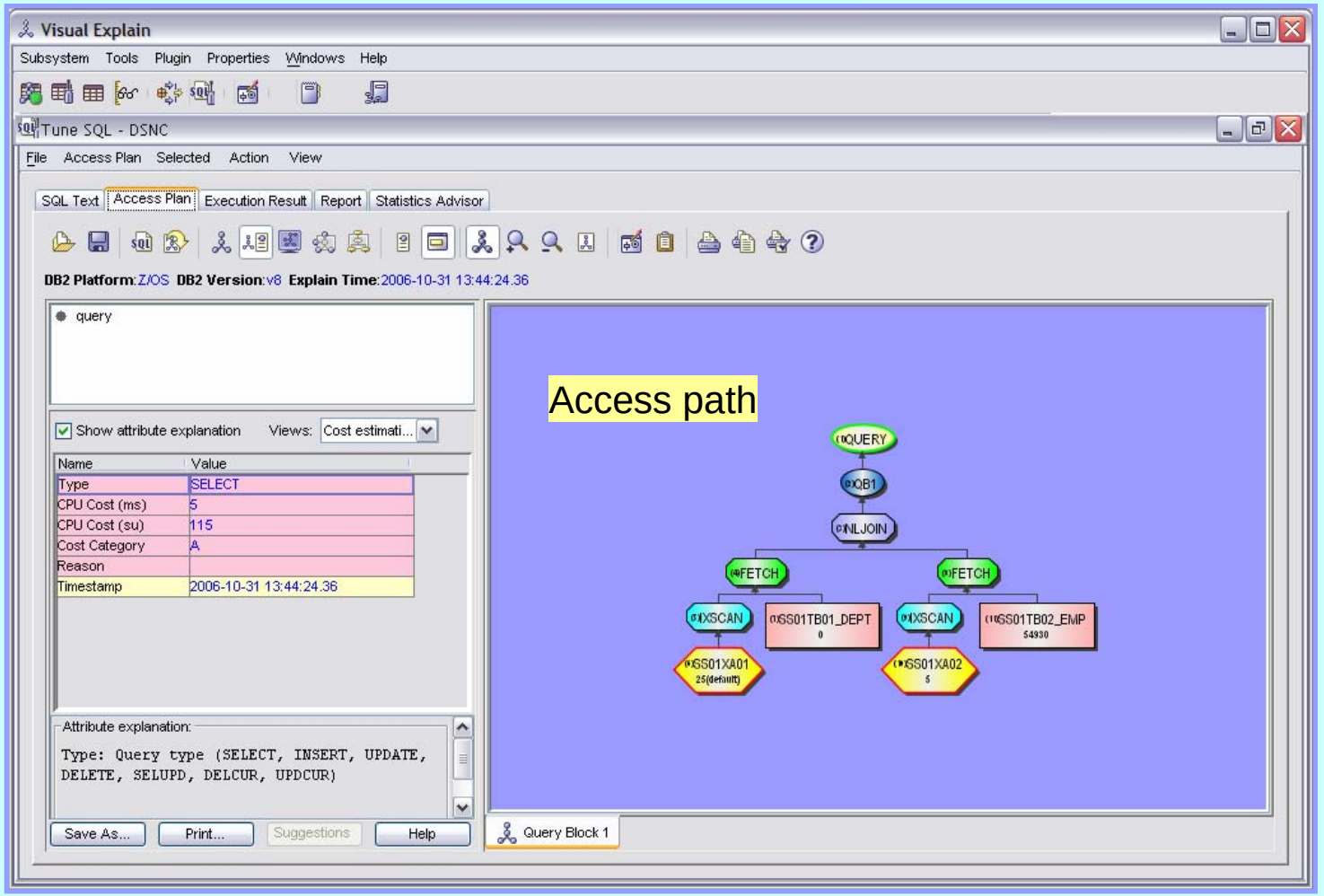
VISUAL EXPLAIN – RETRO EXPLAIN

Click to display a graphical representation of the access path



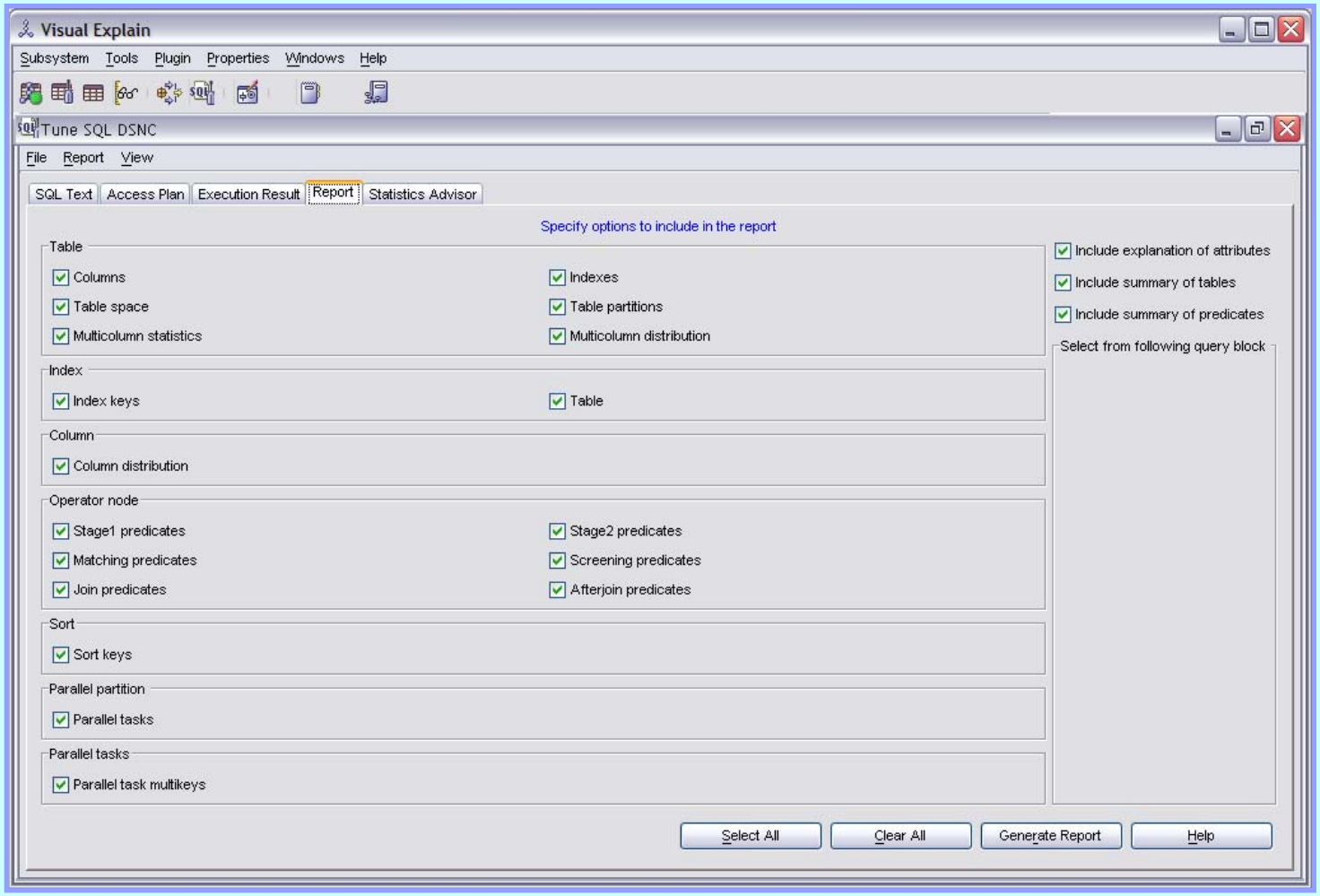
QUERYNO	QBLOCKNO	APPLNAME	PROGNAME	PLANNO	METHOD	CREATOR	TNAME	TABNO	ACCESSTYPE	MATCHCOLS	AC
90000131	1		ANLMAIN	3	3			0		0	
90000133	1		ANLMAIN	1	0	DBA104	SS01TB01_DEPT	2	I	1	SY
90000133	1		ANLMAIN	2	1	DBA104	SS01TB02_EMP	1	I	1	DB
90000134	1		ANLMAIN	1	0	DBA104	SS01TB02_EMP	1	R	0	
90000135	1		ANLMAIN	1	0	DBA104	CANDIDATE	1	R	0	
90000135	1		ANLMAIN	2	1	DBA104	TEST_TAKEN	2	R	0	
90000135	1		ANLMAIN	3	3			0		0	
90000136	1		ANLMAIN	1	0	DBA104	CANDIDATE	1	R	0	
90000136	1		ANLMAIN	2	1	DBA104	TEST_TAKEN	2	R	0	
90000136	1		ANLMAIN	3	3			0		0	
90000137	1		ANLMAIN	1	0	DBA104	SS01TB01_DEPT	2	I	1	SY
90000137	1		ANLMAIN	2	1	DBA104	SS01TB02_EMP	1	I	1	DB
90000138	1		ANLMAIN	1	0	DBA104	SS01TB02_EMP	1	R	0	
0	1		SQLC2E07	1	0	DB2PMPDB	DB2PM_STAT_GENERAL	1	I	0	DB
0	1		SQLC2E07	2	3			0		0	
100000001	1		ANLPGM31	1	0	DBA104	CANDIDATE	1	R	0	
100000001	1		ANLPGM31	2	1	DBA104	TEST_TAKEN	2	R	0	
100000001	1		ANLPGM31	3	3			0		0	
100000002	1		ANLPGM31	1	0	DBA104	CANDIDATE	1	R	0	
100000002	1		ANLPGM31	2	1	DBA104	TEST_TAKEN	2	R	0	
100000002	1		ANLPGM31	3	3			0		0	
100000003	1		ANLPGM31	1	0	DBA104	SS01TB01_DEPT	2	I	1	SY
100000003	1		ANLPGM31	2	1	DBA104	SS01TB02_EMP	1	I	1	DB
100000004	1		ANLPGM31	1	0	DBA104	SS01TB02_EMP	1	R	0	

VISUAL EXPLAIN – GUI Access Path View

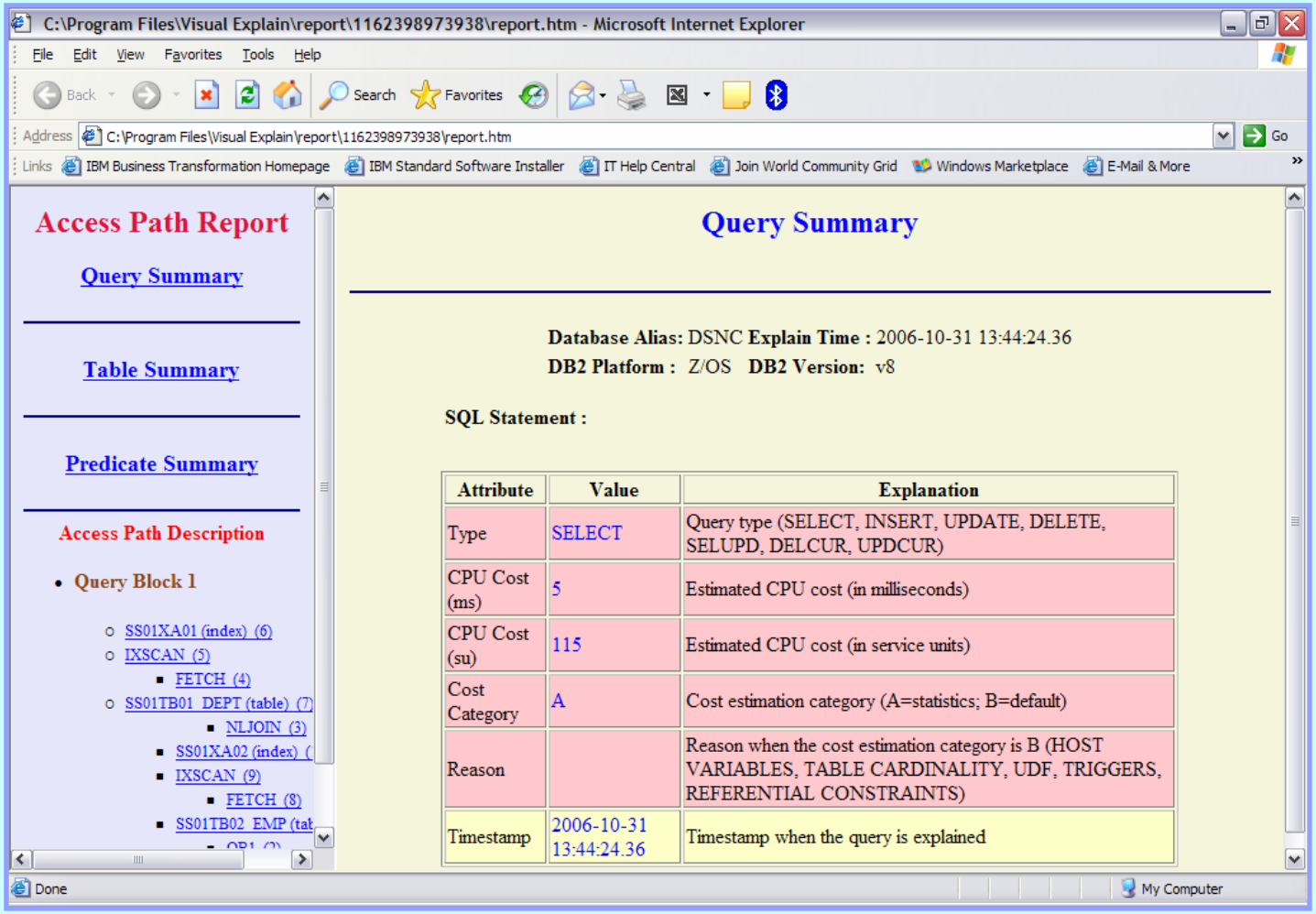


Cost

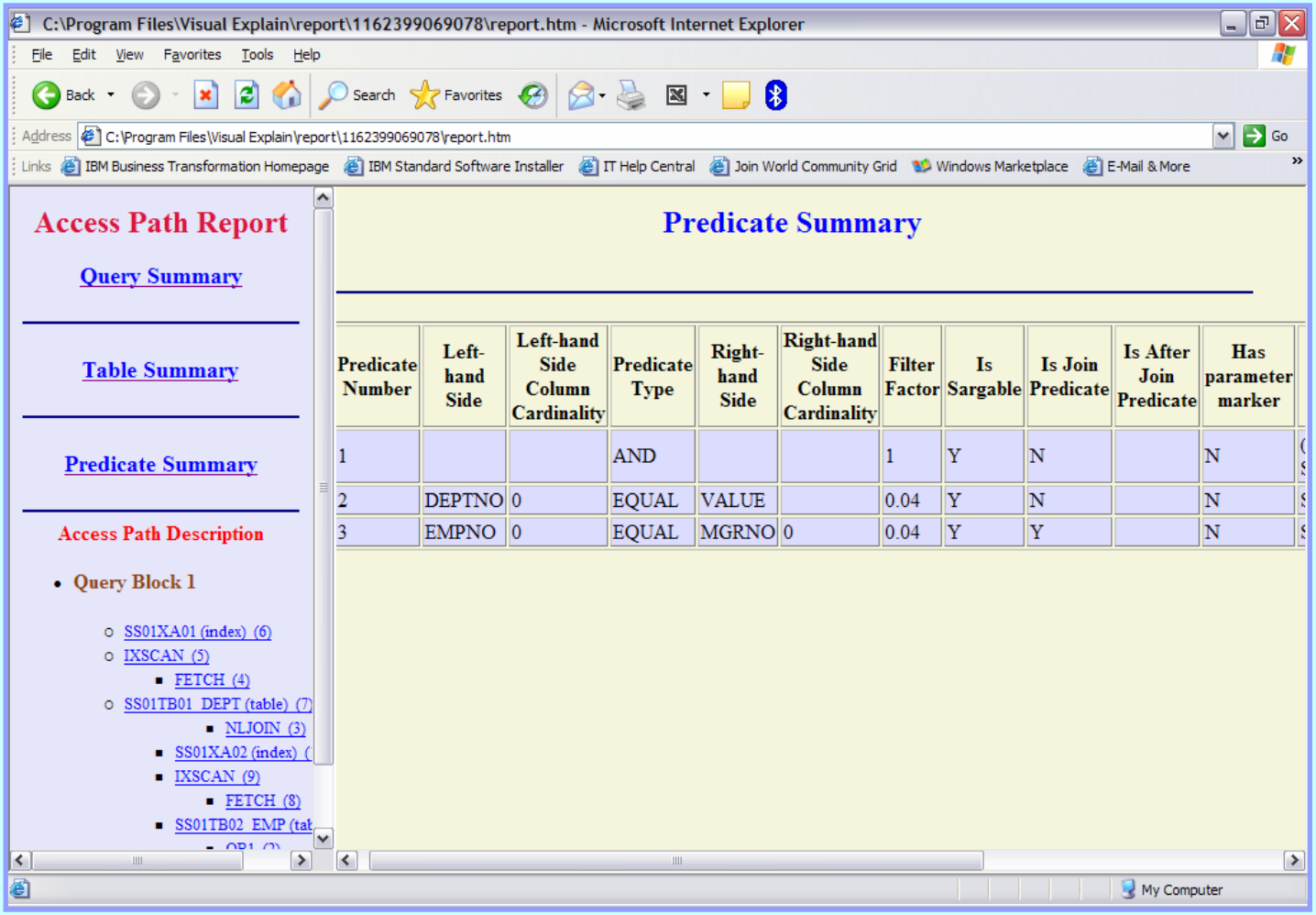
VISUAL EXPLAIN - Report Options



VISUAL EXPLAIN – Query Summary Report

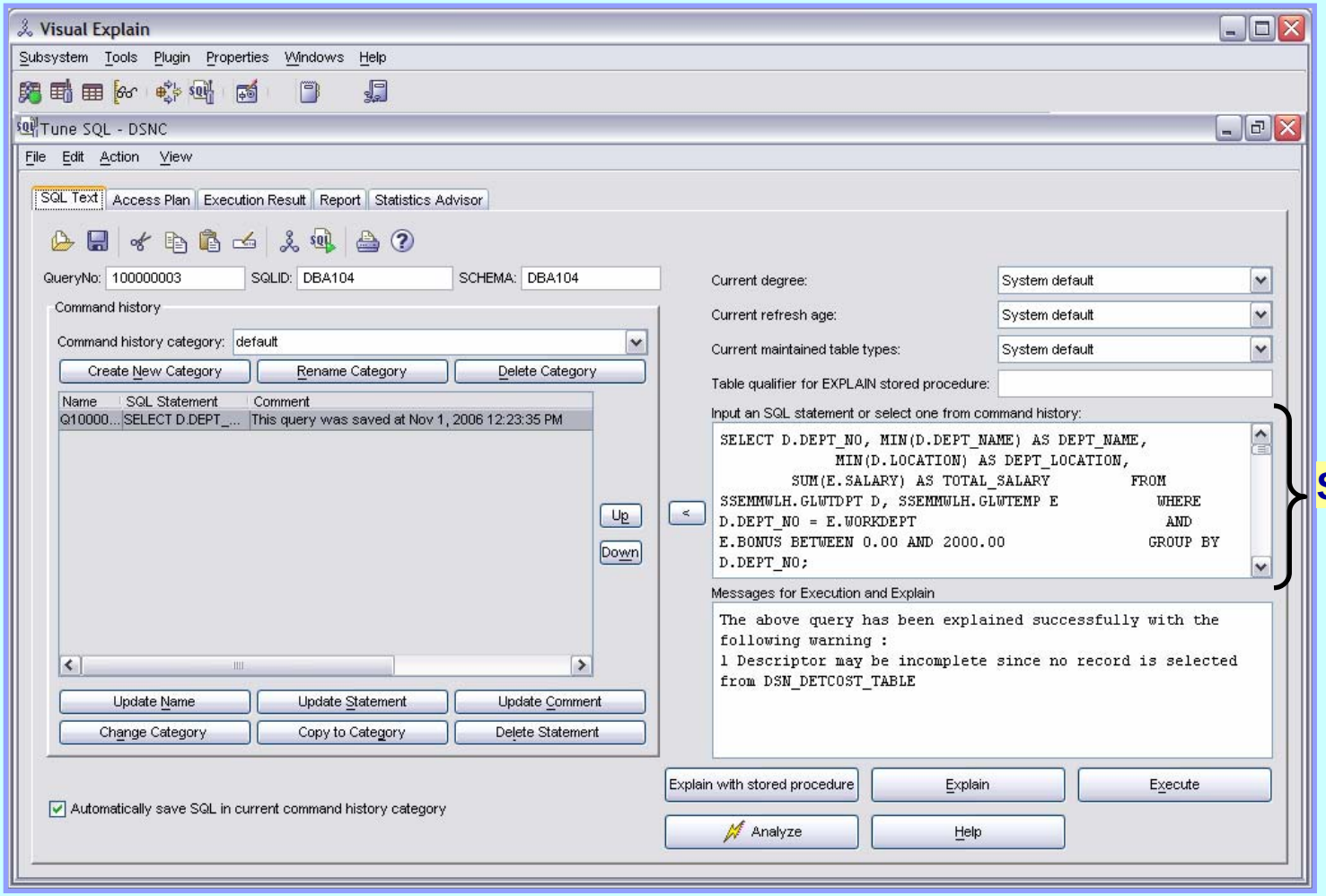


VISUAL EXPLAIN – Predicate Summary Report

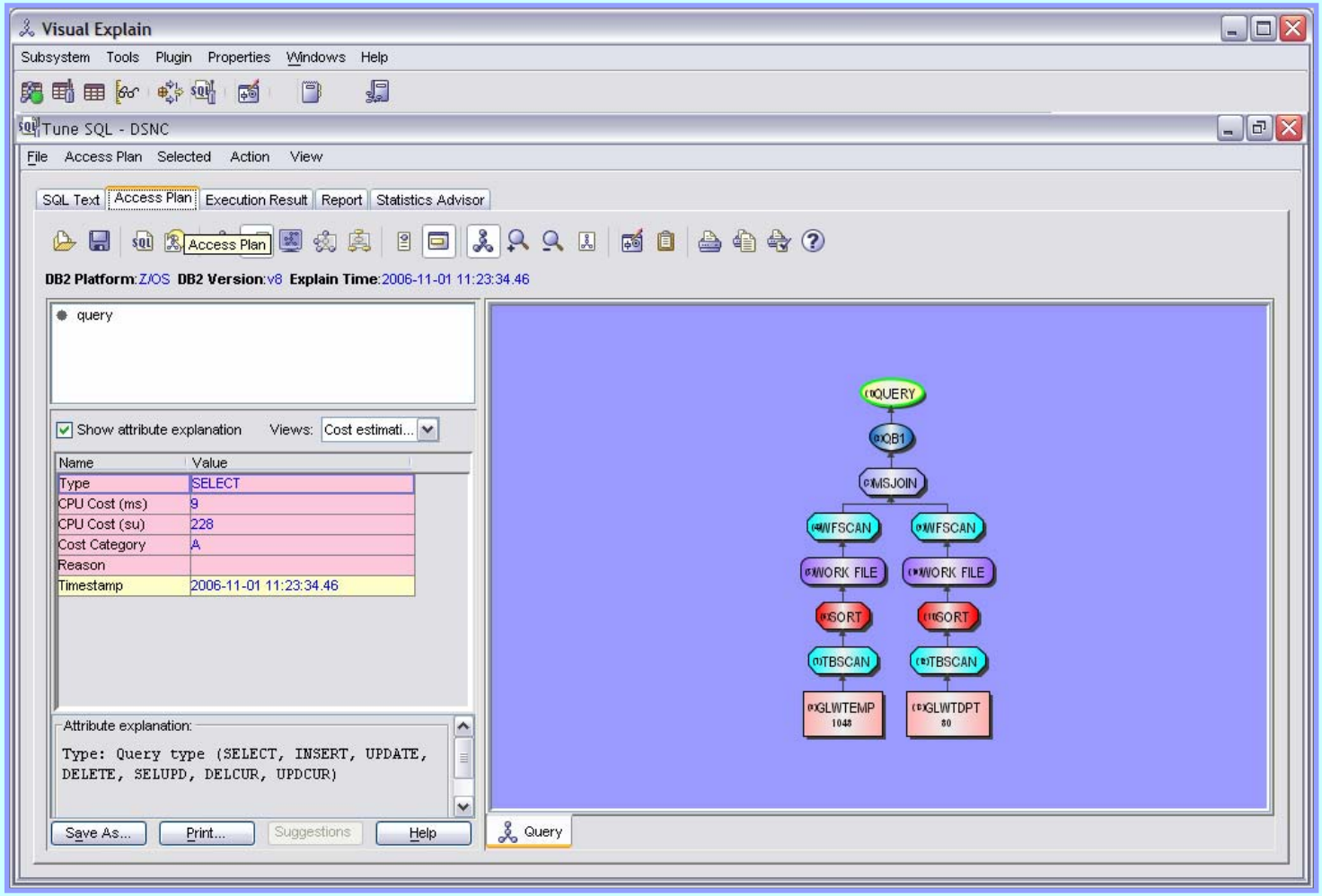


Predicate Number	Left-hand Side	Left-hand Side Column Cardinality	Predicate Type	Right-hand Side	Right-hand Side Column Cardinality	Filter Factor	Is Sargable	Is Join Predicate	Is After Join Predicate	Has parameter marker
1			AND			1	Y	N		N
2	DEPTNO	0	EQUAL	VALUE		0.04	Y	N		N
3	EMPNO	0	EQUAL	MGRNO	0	0.04	Y	Y		N

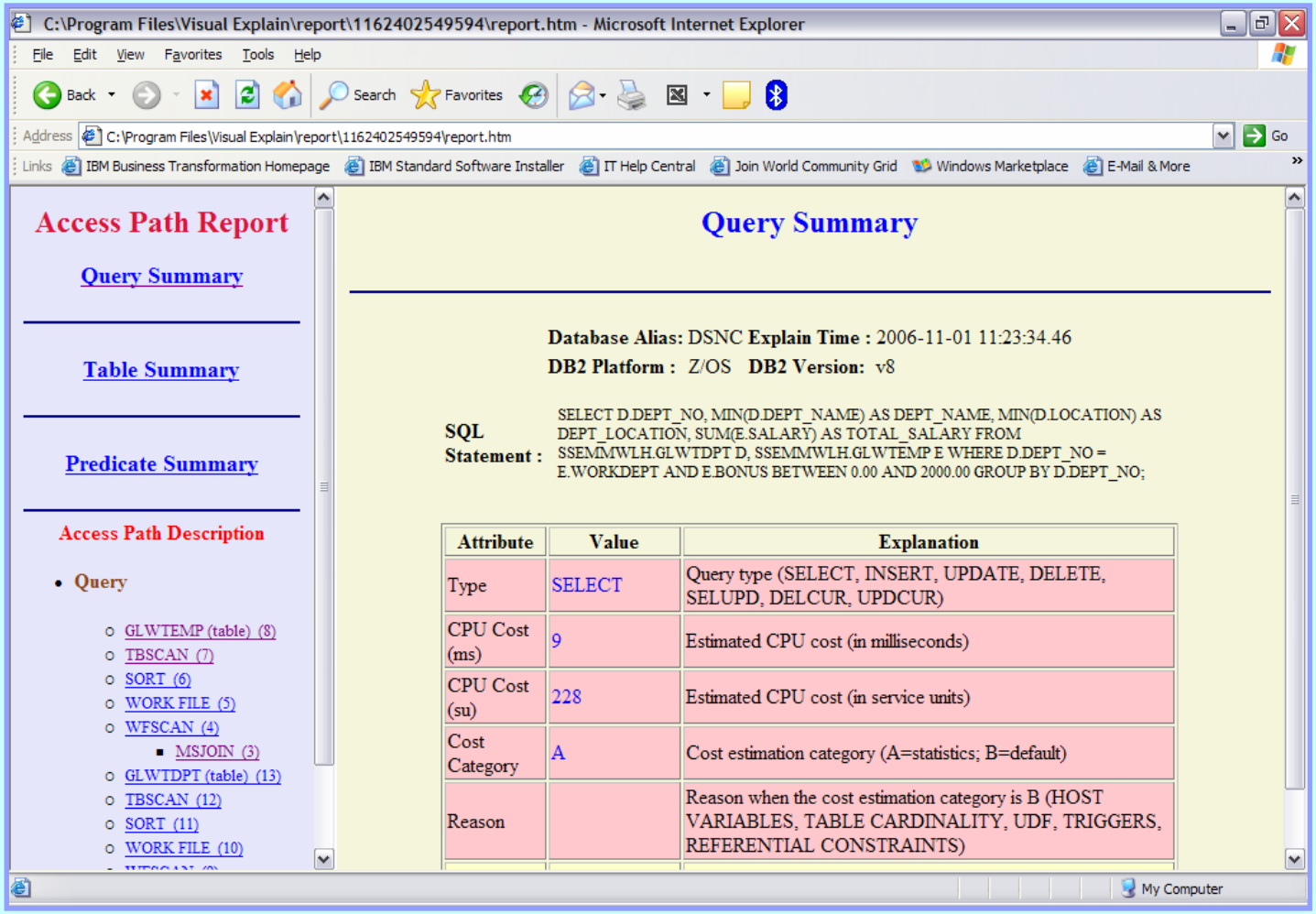
VISUAL EXPLAIN - Key in an SQL Statement



VISUAL EXPLAIN – Access Path Information



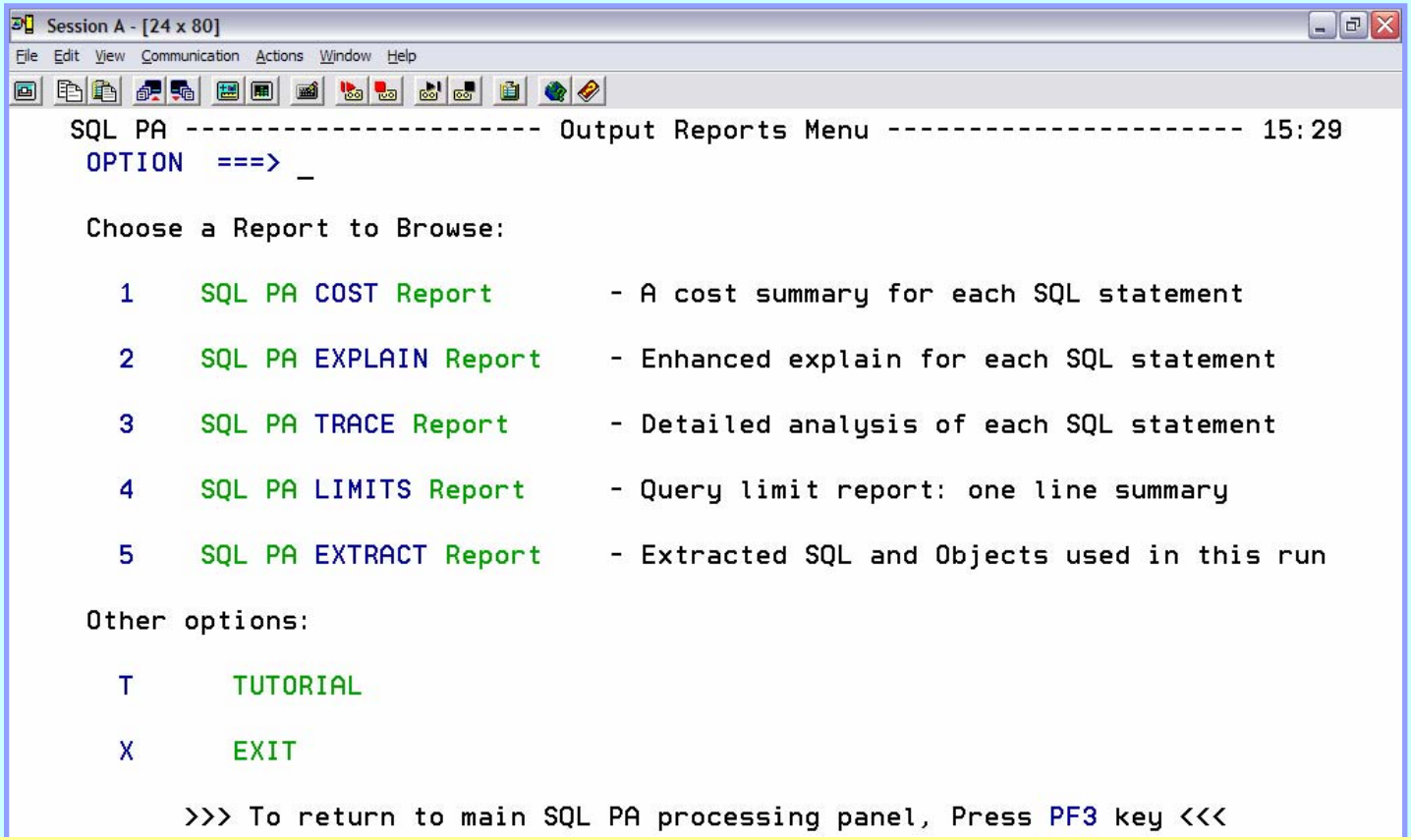
VISUAL EXPLAIN – Query Summary Report



DB2 SQL PERFORMANCE ANALYZER

- Enhanced Explain Tool
- Can run
  - In BATCH, as normal z/OS job
  - In TSO, under ISPF Panel interface
- Input
  - Sequential data sets (PS)
  - Partitioned data set members (PDS)
  - DBRM library members (PO)
    - ***Entire*** DBRM Libraries, with member selectivity
  - In QMF, as a Governor Intercept
  - In **any** DB2 application, via a Stored Procedure call
- Produces a variety of reports

DB2 SQLPA Reports



All reports are a forecast: what will be if the SQL is executed

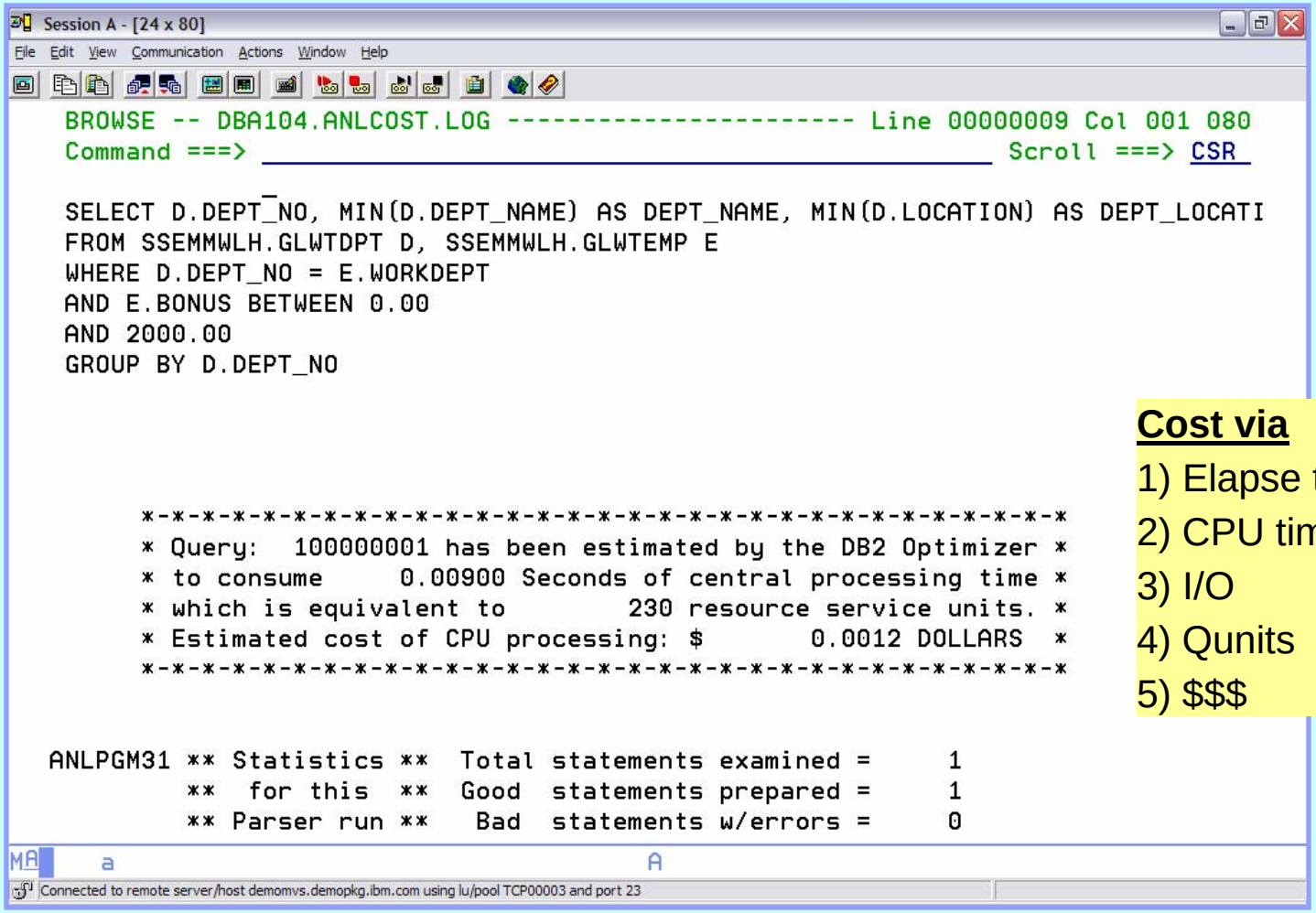
What does DB2 SQLPA actually do ??

- It **forecasts** SQL performance:
  - Response Times
  - CPU Times
  - I/O Counts

- It **forecasts** the **COST** of the query, in terms of:
  - Charge Back (monetary, in national currency)
  - QUNITS ™ (query service units)

SQL PA transforms Optimizer access paths into their real world costs

DB2 SQLPA - Cost Report



Cost via

1) Elapse time
2) CPU time
3) I/O
4) Qunits
5) $$$

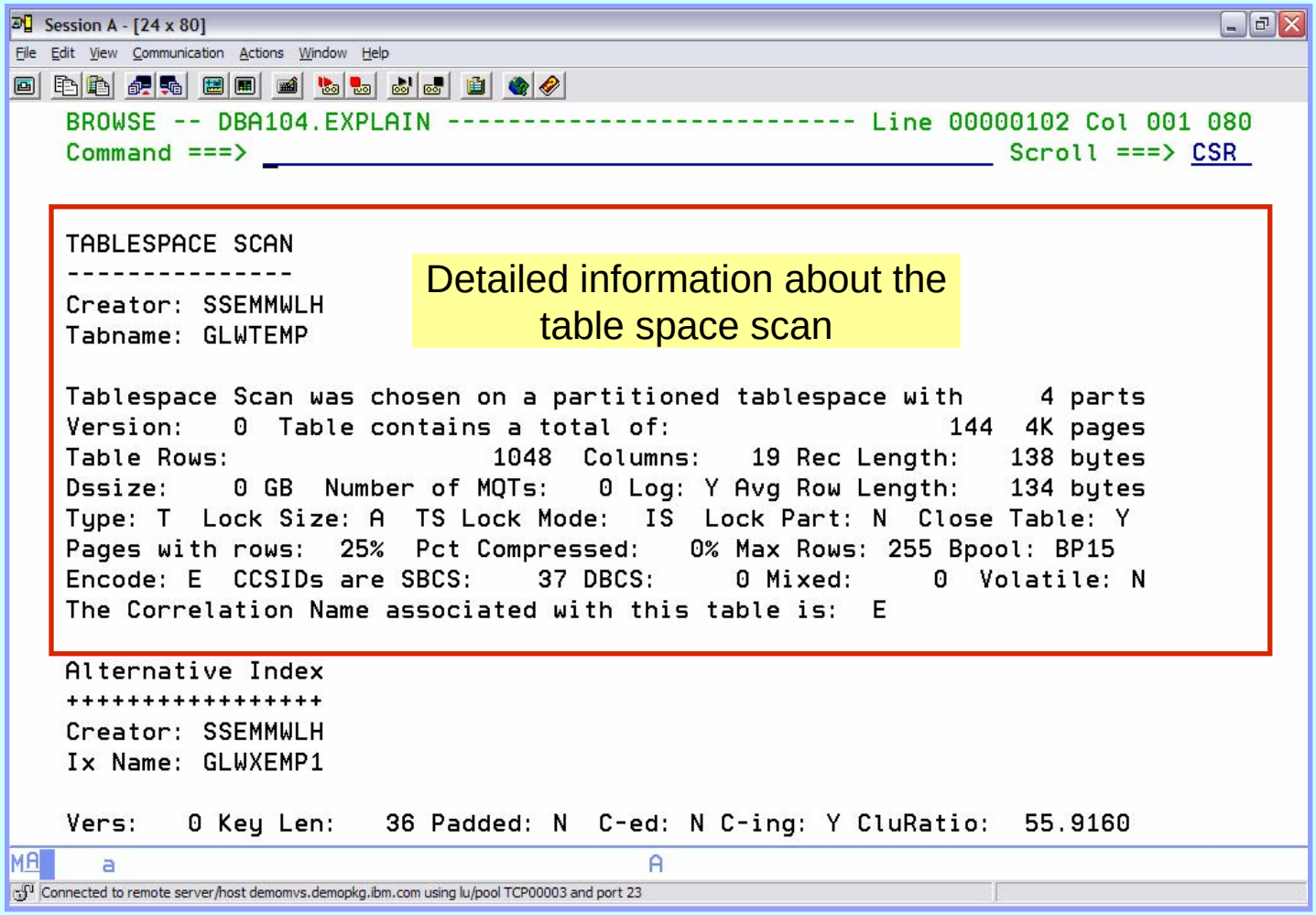
What else does DB2 SQLPA do ??

- It provides an *Enhanced* EXPLAIN report:
  - Catalog Statistics
  - Access Path Information
  - RI Relationships

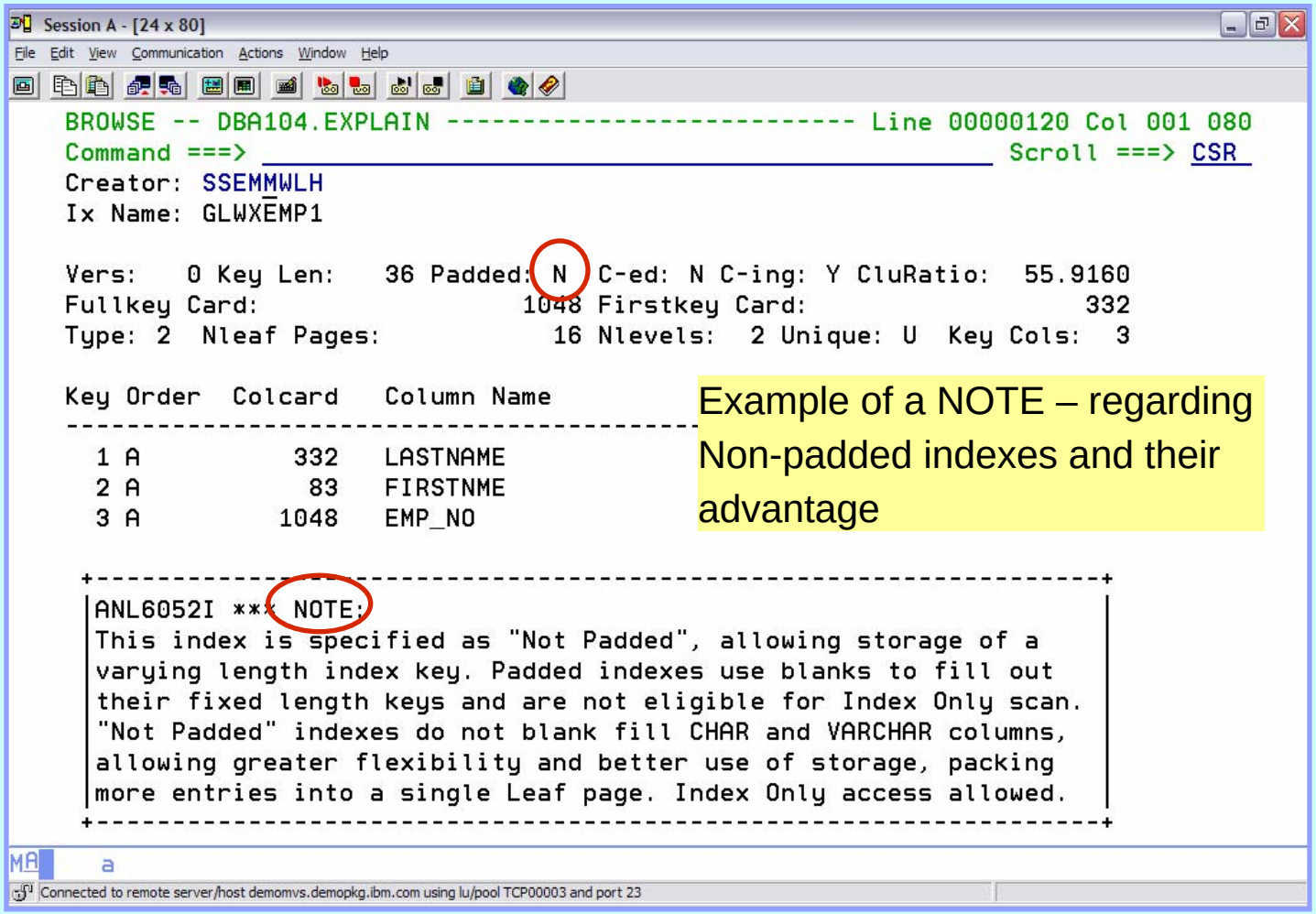
- It provides key **ADVICE** on each SQL statement:
  - Warnings and Alerts
  - Guidelines and Recommendations
  - Performance Notes and Good News

SQL PA teaches users how to write better SQL

DB2 SQLPA – Enhanced Explain Report



DB2 SQLPA Enhanced Explain Report (2)

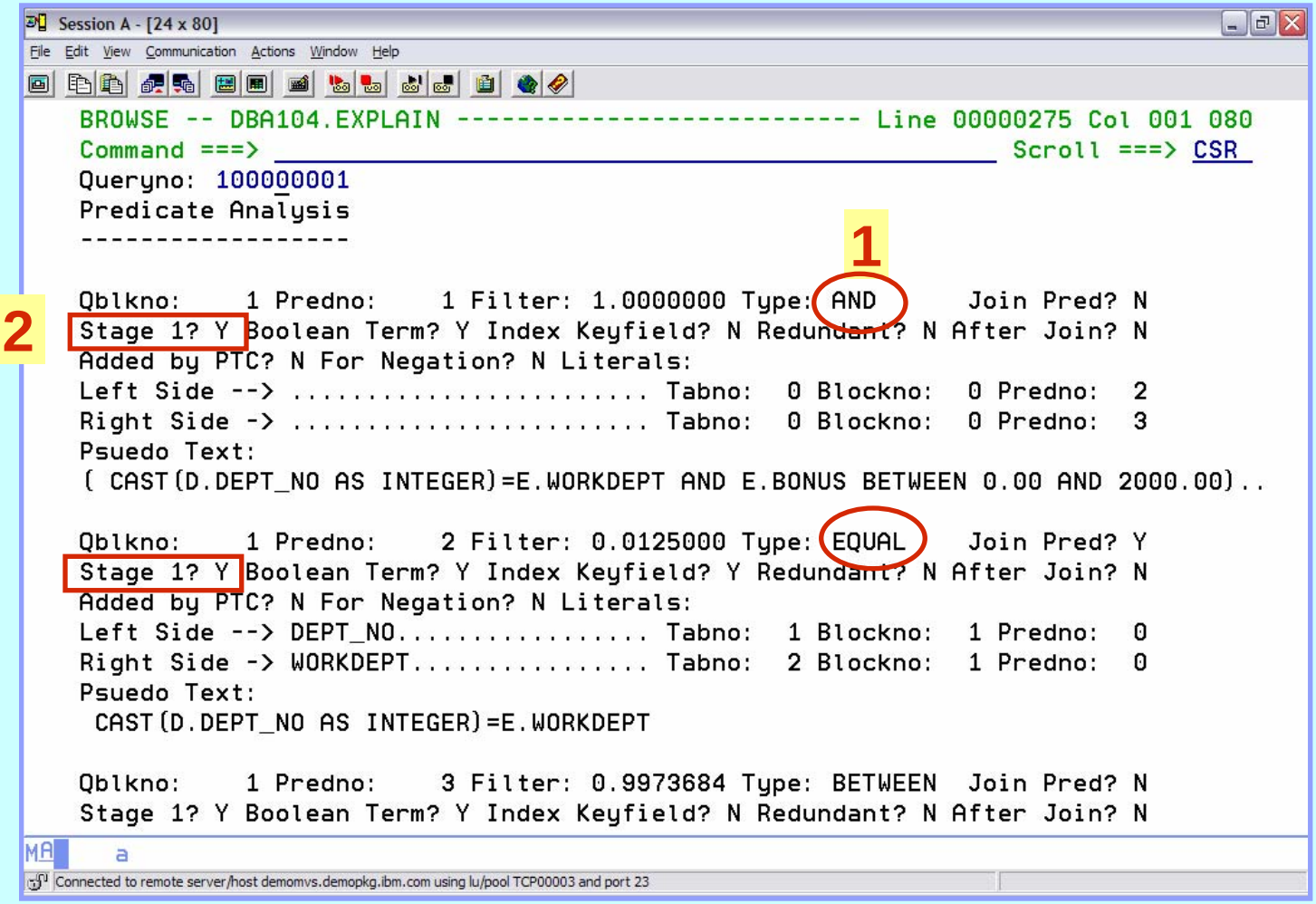


Example of a NOTE – regarding Non-padded indexes and their advantage

DB2 SQLPA Predicate Analysis Report

- Embedded in the Enhanced Explain Report
- Provides optimizer insights and **filter factors** for each predicate, even those generated by predicate transitive closure
- All PRED characteristics are provided: predicate type, stage 1 or 2, indexable, boolean term, used for join, generated by DB2, redundant, etc.

DB2 SQLPA - Predict Analysis Sample Output



Does DB2 SQLPA do anything else ??

- It provides a detailed execution *"forecast"* report:

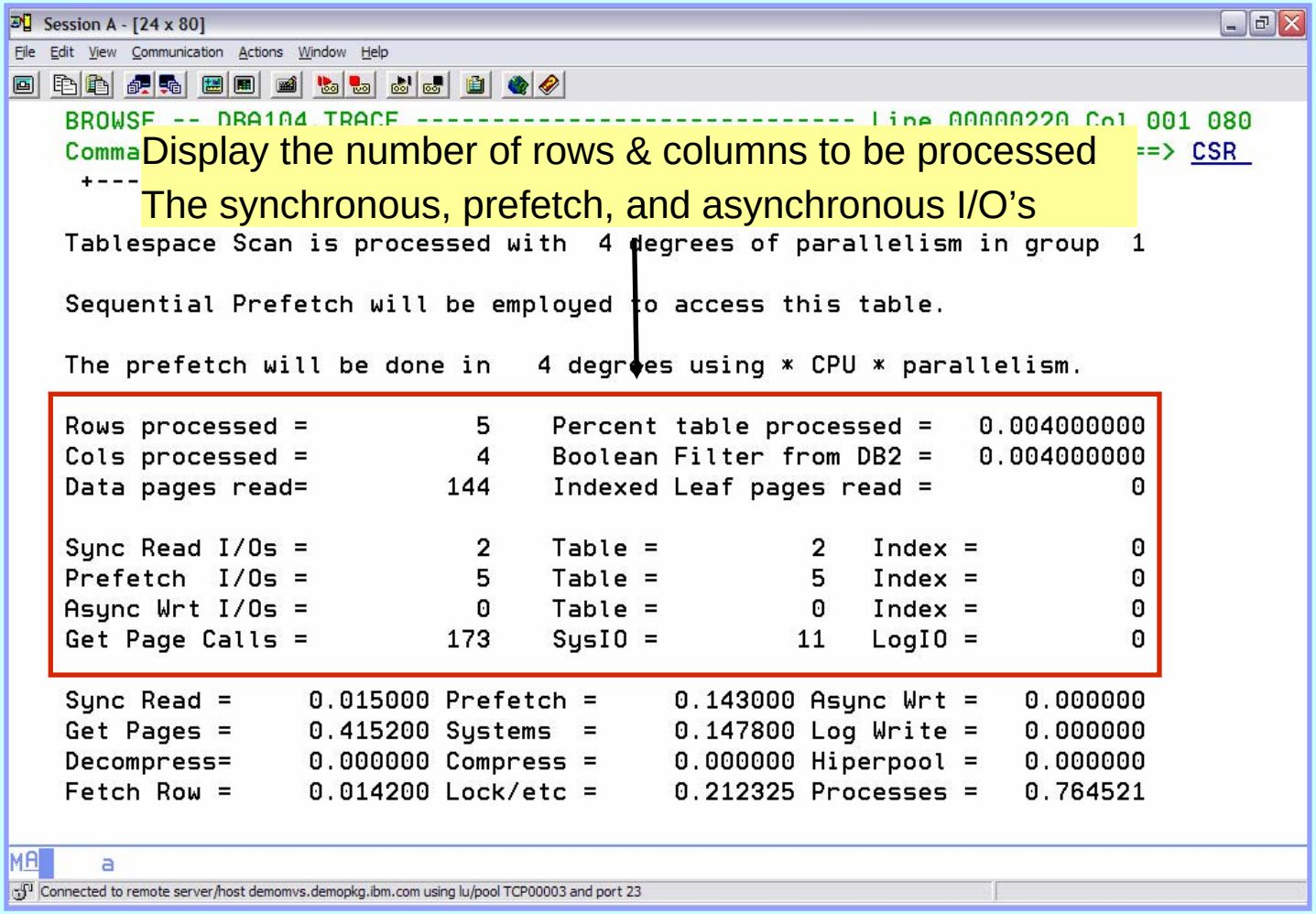
- Breaks down SQL by time spent in DB2 components

  - Wait Times and Bottlenecks, Path Lengths and I/O Types

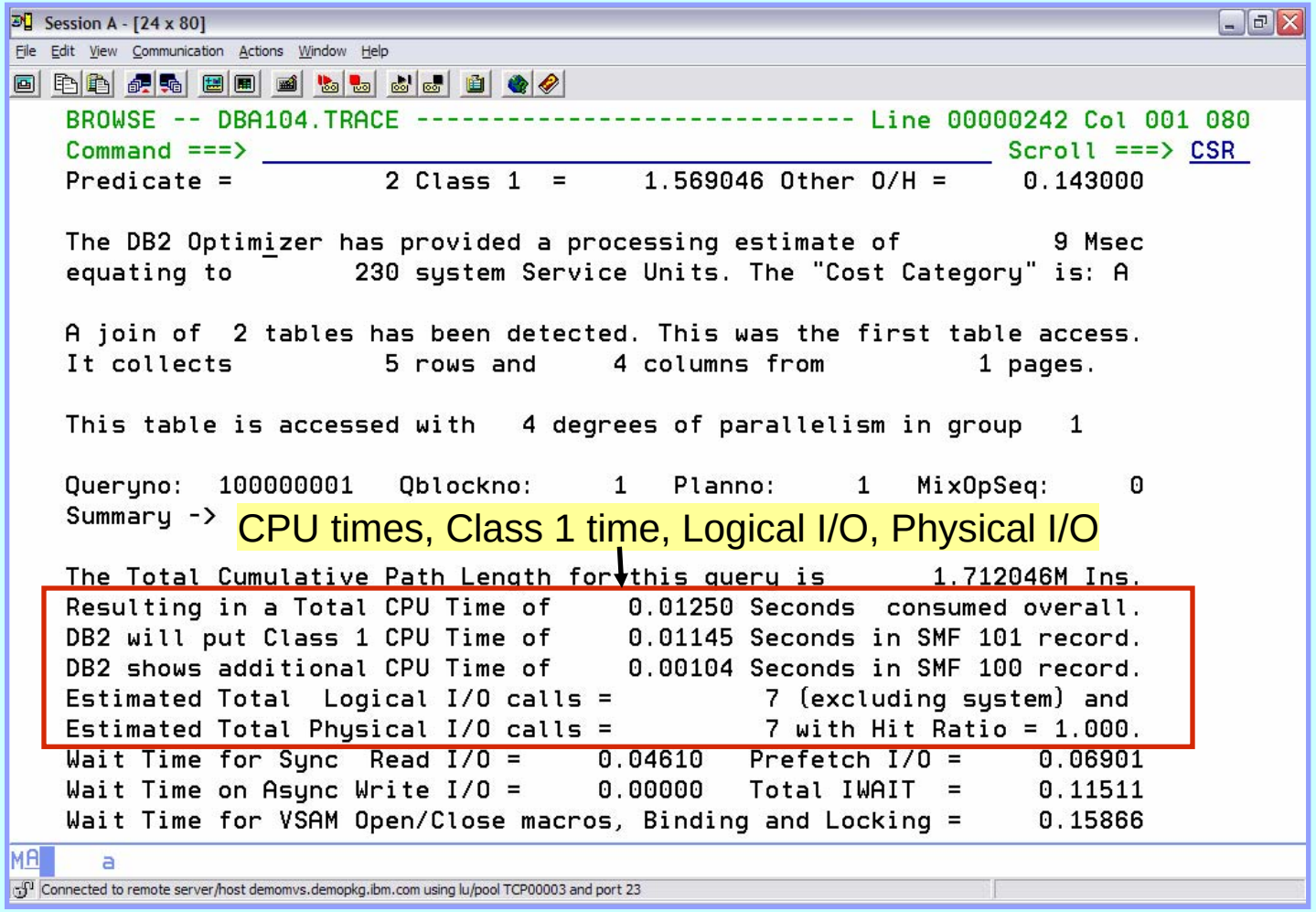
  ...*for the serious 'bits and bytes' DBA*

Suitable for SQL Development, Tuning and Control

DB2 SQLPA – Trace Report



DB2 SQLPA – Trace Report (2)



DB2 SQLPA - QLIMITS Report

- One line summary for each SQL statement evaluated
- Recap of costs
- Quick eye catcher for problem queries
- Can be sorted by any column on the report or combination thereof

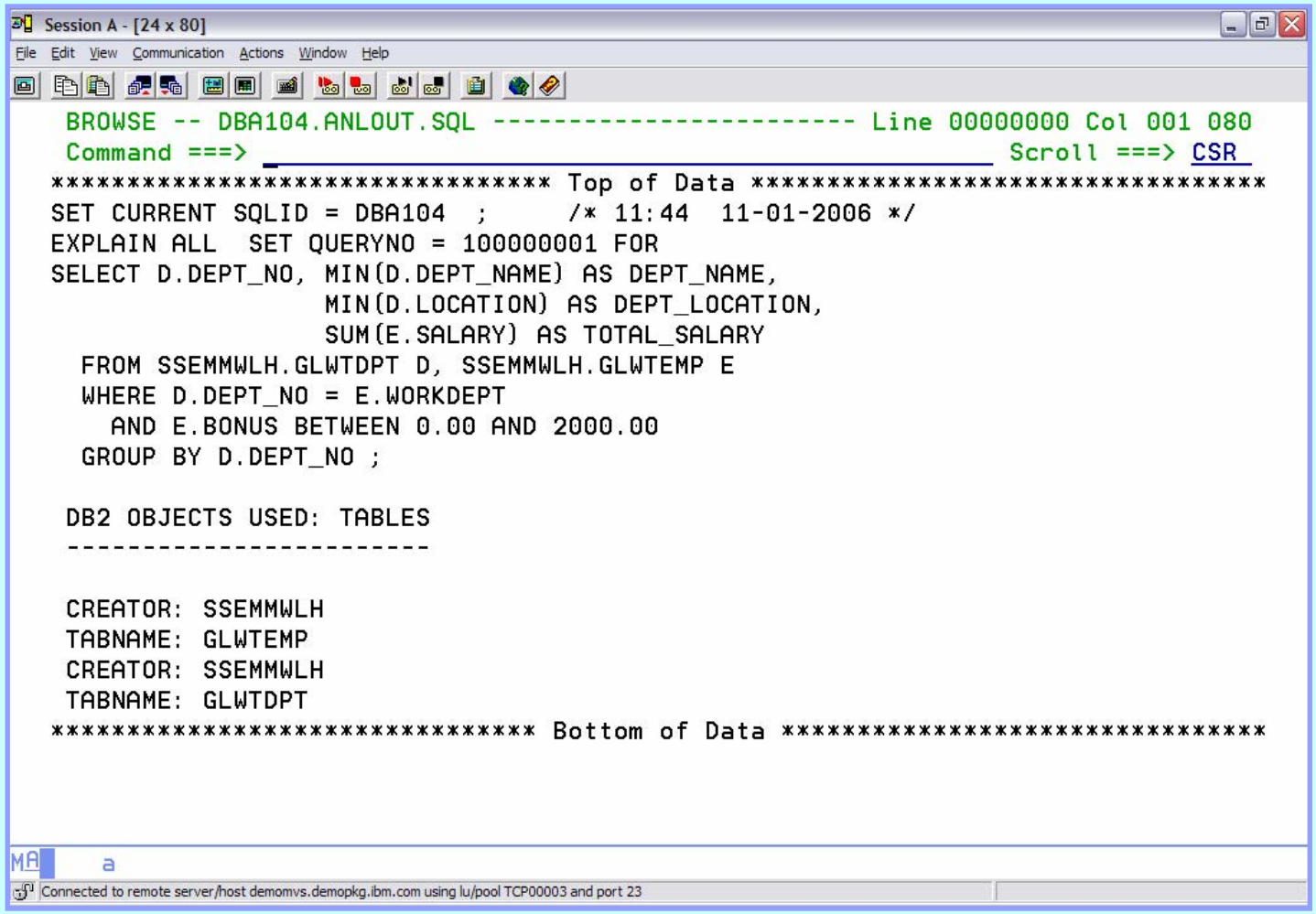
DB2 SQLPA – QLIMITS Report

Exceeded threshold

```
Session A - [24 x 80]
File  Edit  View  Communication  Actions  Window  Help
BROWSE -- DBA104.QLIMIT ------------------------------------ Columns 001 072
Command ===> _____________________________________________ Scroll ===> CSR
****** ******************************** Top of Data ********************************
==MSG> -Warning- The UNDO command is not available until you change
==MSG>           your edit profile using the command RECOVERY ON.
000001 15:57:27.844              SQL Performance Analyzer                Versio
000002  10-30-2006                 Query Limits Report                  Level
000003
000004 CEIQ$ Error  Queryno  Type    CPU Time     Elapsed   Phys I/O    Qunits
000005 ---Y-     0  100000004 ST       0.61100       1.544         18     15654
000006 -----     0  100000003 ST       0.00500       0.313         15       115
000007 -----     0  100000001 ST       0.00100       1.268       6299         5
000008 -----     0  100000002 ST       0.00100       1.584      18830         4
****** ***************************** Bottom of Data ******************************
. . . . . .
  Menu  Functions  Confirm  Utilities  Help
MA    a                                A
Connected to remote server/host demomvs.demopkg.ibm.com using lu/pool TCP00079 and port 23
```

Type of statements: ST = SELECT, IT = INSERT, UE = UPDATE, DE = DELETE …

DB2 SQLPA – Extract Report



What can DB2 SQLPA do for you?

- Provide expert advice on how to improve your SQL
- Warn users of long running queries
- Illustrate the incremental components of cost
- Help fine tune design of DB2 queries and databases
- Help tune production SQL via DBRM scans
- Evaluate future production volume performance
- Preempt costly QMF Governor cancels
- Implement governing for ***any*** DB2 application

All of this is done without ever executing the SQL

How does DB2 SQLPA work?

- *Emulates* and *Invokes* the DB2 Optimizer
- *Analyzes* the Access Plan chosen by DB2
- *Estimates* the Execution Cost in the **Real World**

All of this is done without actually executing the SQL statements!

DB2 SQLPA - Parameter Driven

- User defined parameters
- System defined parameters
- Advisor Thresholds

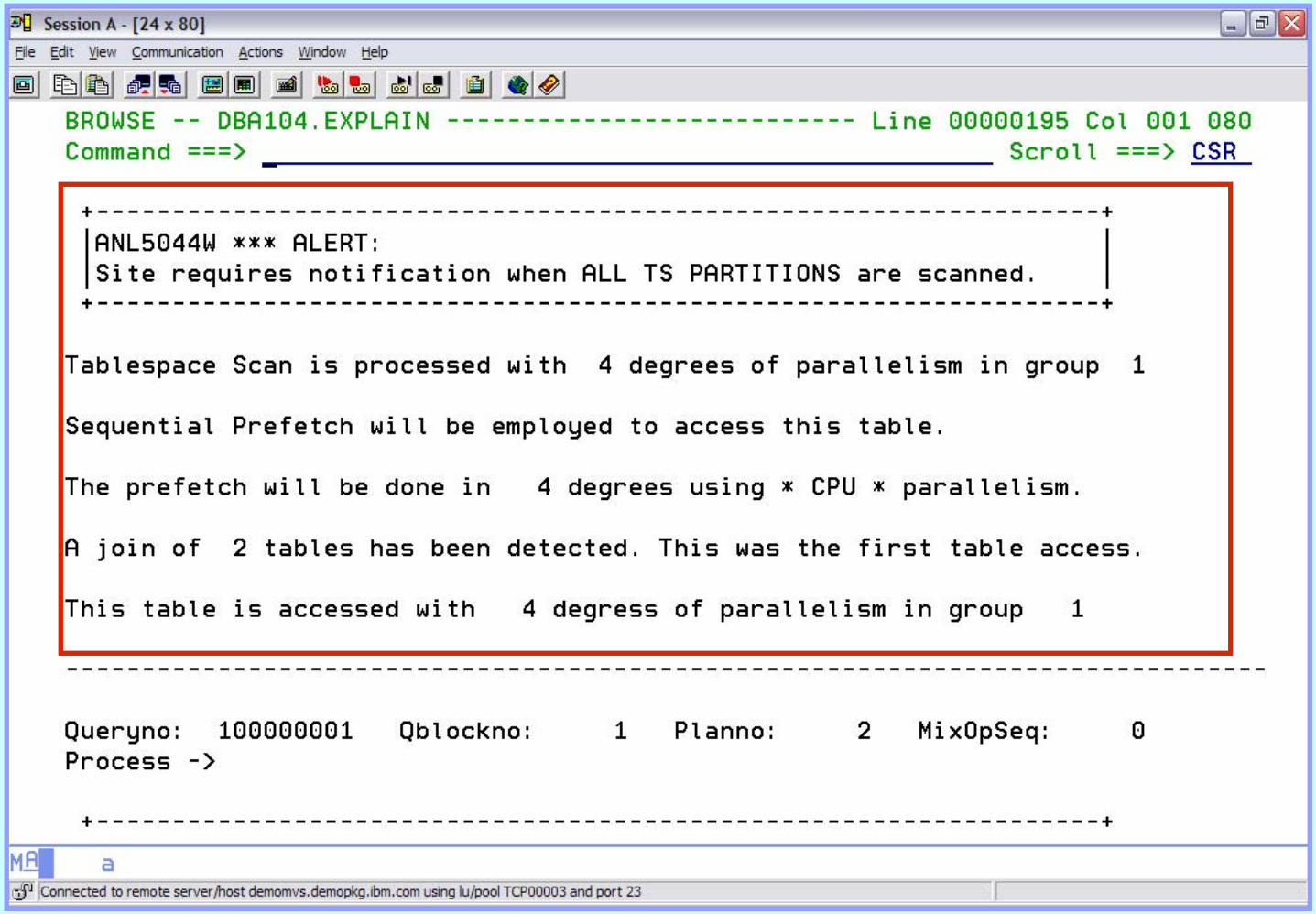
DB2 SQLPA – User Defined Parameters

- Select which reports to produce (Cost and QLIMITS are mandatory)
- Control amount of detail displayed in reports
- Show or not show alternate indexes
- Consider MQT's when determining access path
- Display the objects being accessed
- Define the qualifier to be used for unqualified SQL
- Specify sort sequence for columns on QLIMITS report

DB2 SQLPA - Advisor Thresholds

- Specify whether or not to display the full message text
- Notified if an object has not had RUNSTATS run against it
- Flag
  - Table space scans
  - Non-matching index scans
  - When all partitions are scanned
- Specify a high water mark for
  - Number of matching index scans
  - Number of non-matching index scans
  - Number of table space scans
  - Acceptable number of tables to be included in a join
  - Acceptable number of items to be included in a LIST
  - Acceptable number of indexes to be updated

DB2 SQLPA - Example of an Alert



What factors are considered inside DB2 SQLPA?

- The influence of DB2 catalog statistics
- The types of predicates being processed
- The hardware and software configuration
- User and installation parameters
- Predicate filter factors
- Optimizer cost estimates
- DB2 current release level
- CPU Processor and DASD speeds

SQL PA has minimal catalog impact: access once for most objects, then store them internally

DB2 SQLPA - Catalog Statistics

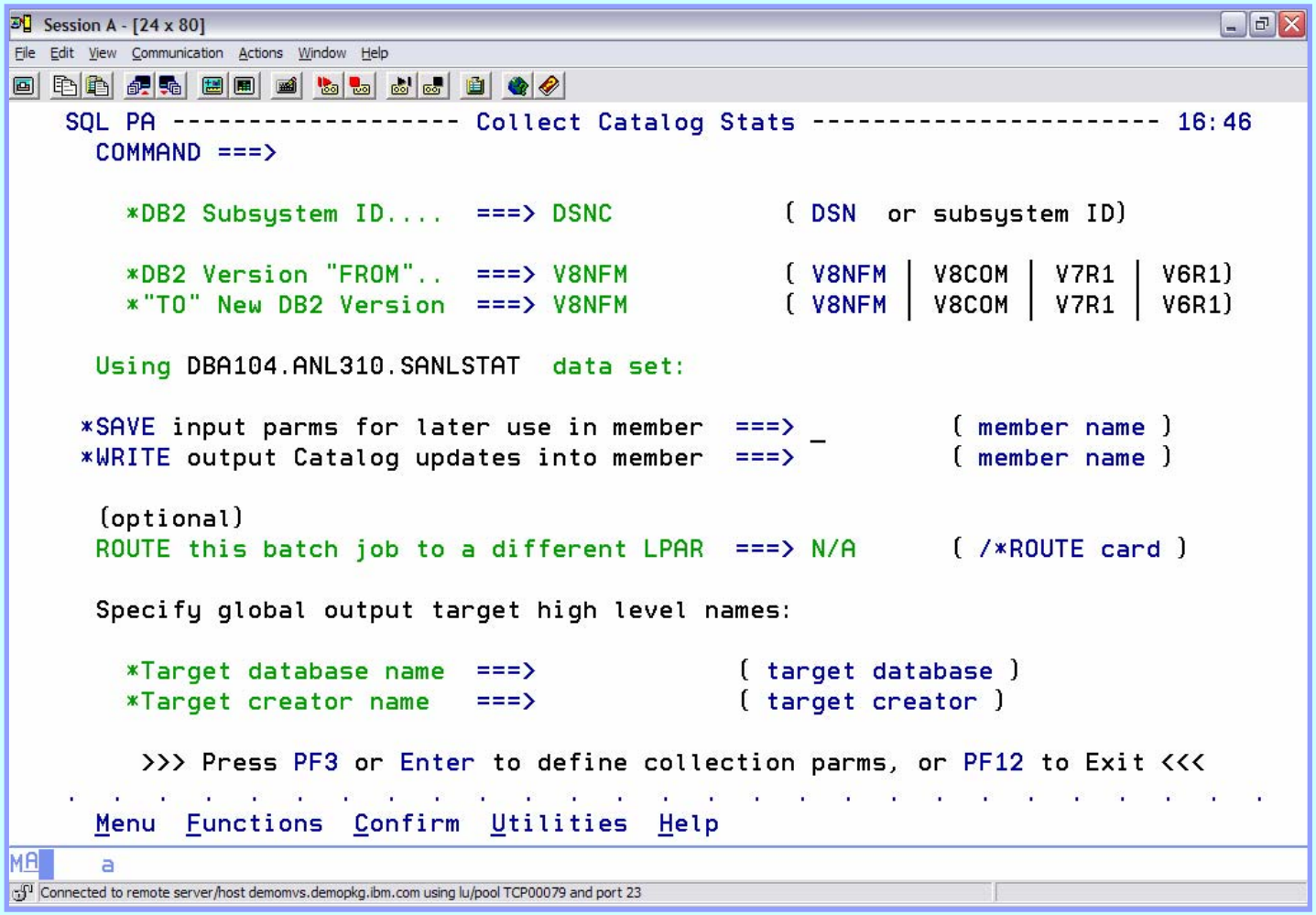
- Statistics are in important ingredient in the optimizers determination of the access path
- The gathering of statistics via RUNSTATS
  - Has no impact on static SQL until the next BIND
  - Impacts the performance of dynamic SQL immediately

DB2 SQL/PA Supports

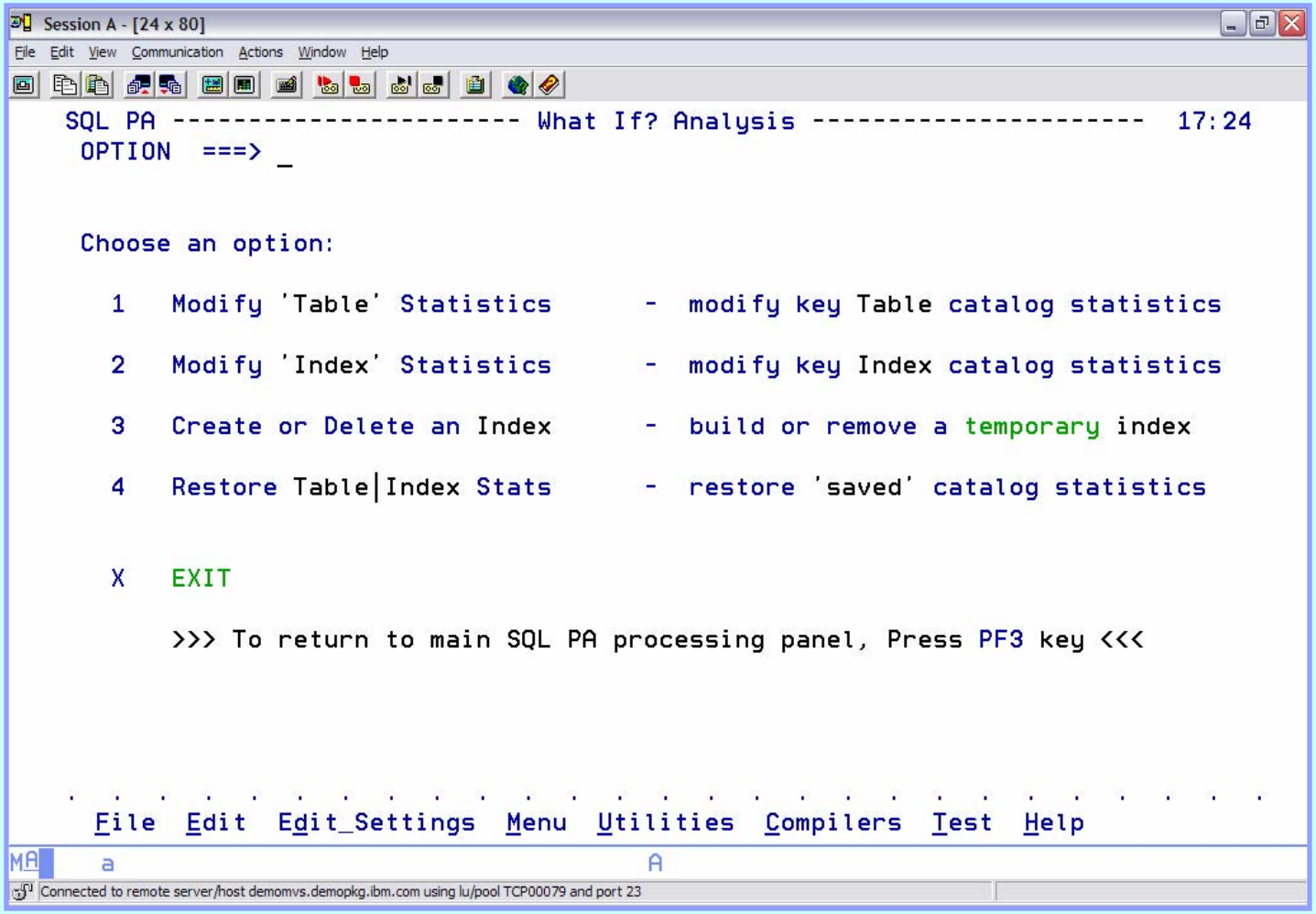
Import of DB2 Catalog Statistics from one subsystem to another (supports wildcarding)

Modification of table and index statistics for WHAT IF scenarios (original statistics may be restored)

DB2 SQLPA - Collect Catalog Statistics



DB2 SQLPA - WHAT IF



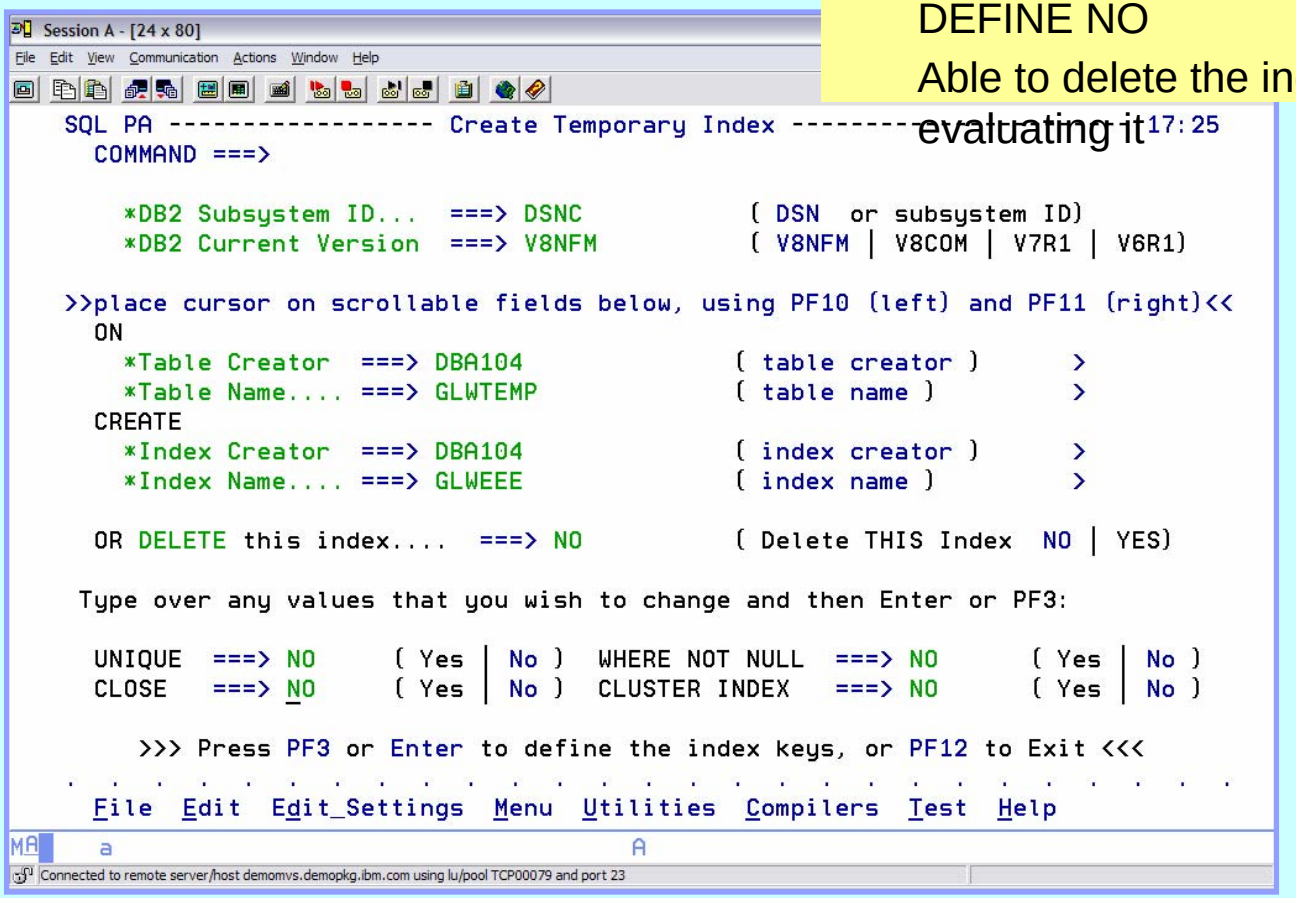
DB2 SQL/PA - WHAT IF

- **Modify catalog statistics**
  - Specify new values for **tables**
    - Number of rows (CARD)
    - Number of pages (NPAGES / NACTIVE)
    - % of the table space used by the table (PCTPAGES)
    - % of row compression (PCTROWCOMP)
  - Specify new values for **indexes**
    - Number of rows indexed by 1st column (FIRSTKEYCARD)
    - Number of rows indexed by all columns (FULLKEYCARD)
    - Size of the index (NLEAF / NLEVELS)
    - Cluster ratio (CLUSTERRATIO)

DB2 SQL/PA - WHAT IF

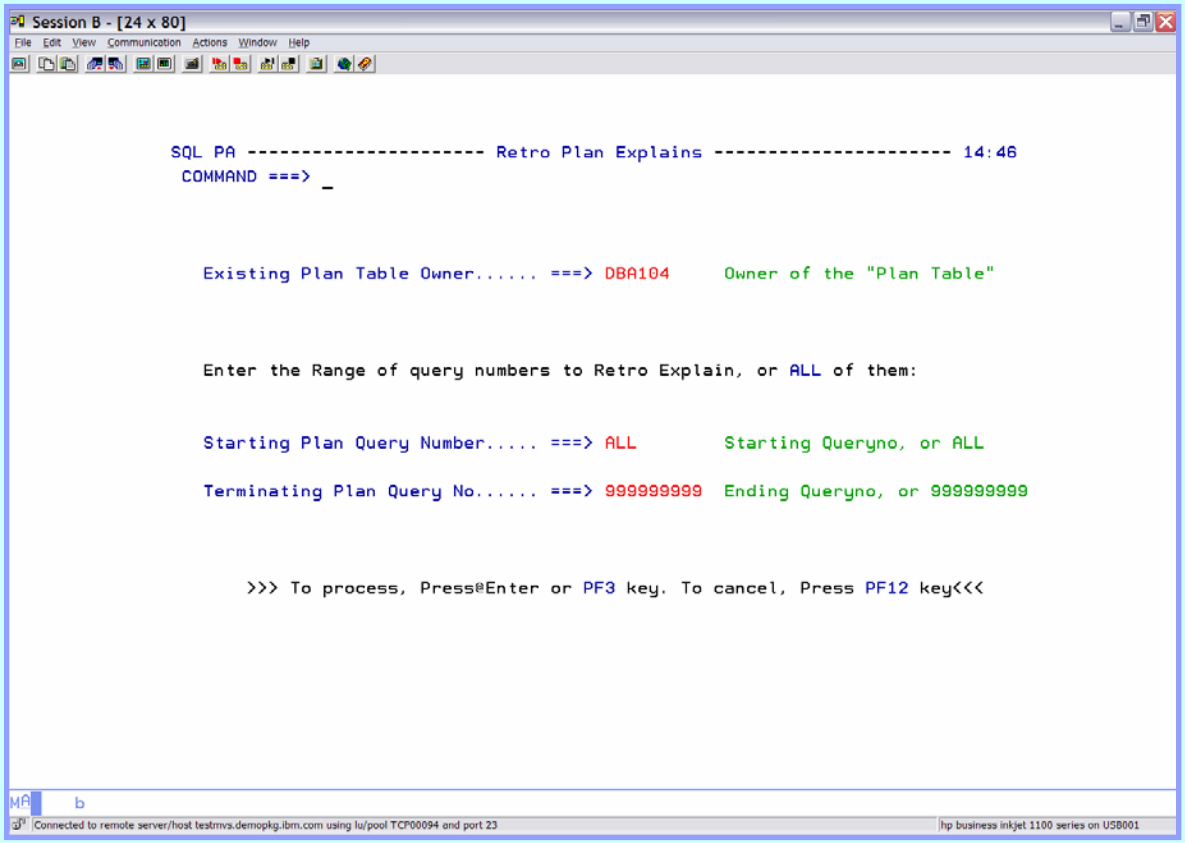
Create a new index

- Maximum of 5 columns
- DEFER YES
- DEFINE NO
- Able to delete the index after evaluating it



DB2 SQLPA - Retro Explain

- Ability to reexamine the old access path stored in a plan table

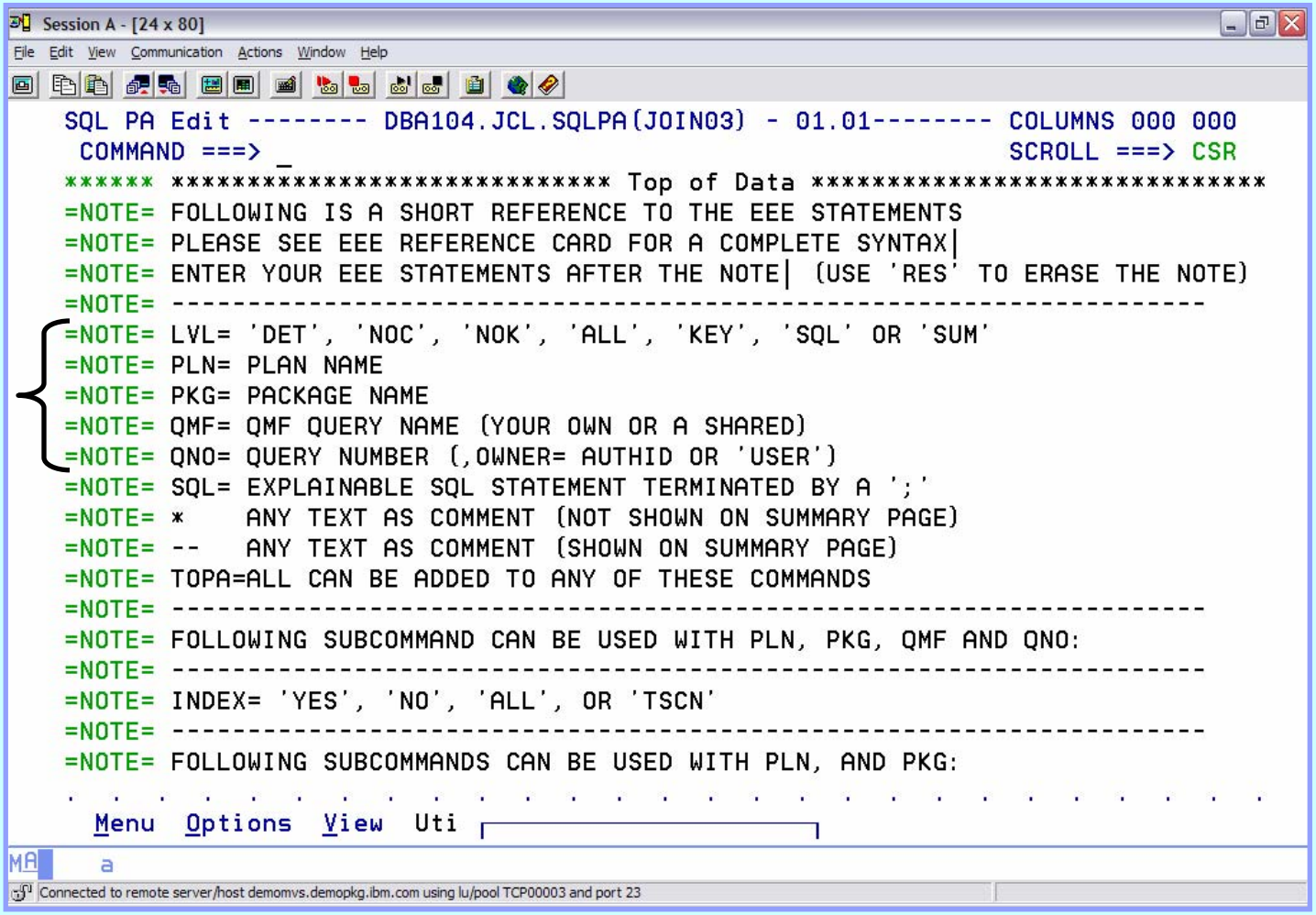


DB2 SQLPA - Easy Explain Feature

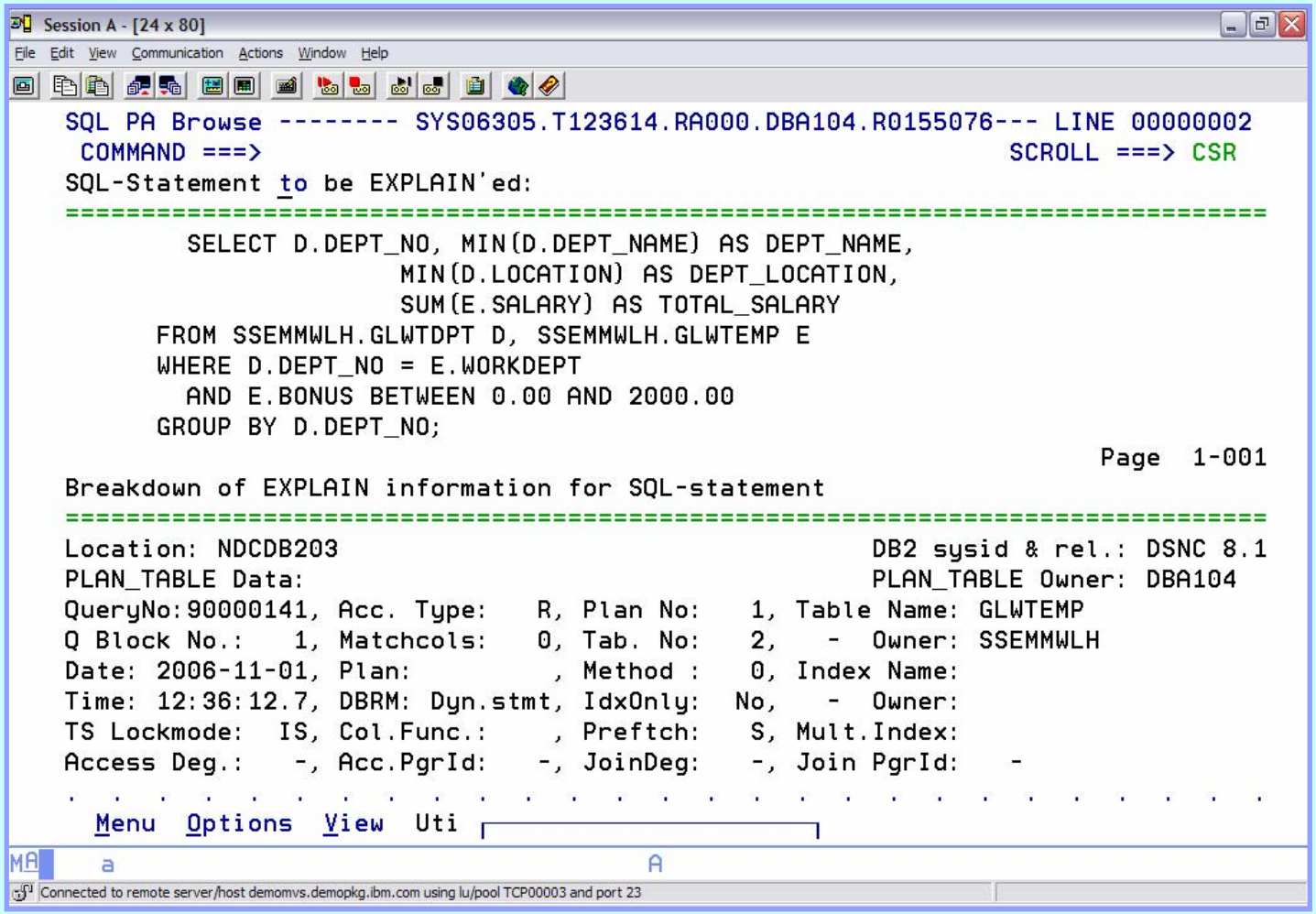
- Provides information about how DB2 accesses data for a given SQL statement – additional DB2 Catalog information
- Input
  - Plans / packages (DB2 Catalog)
  - Query number of previous explain operation (Plan table entry)
  - QMF
  - PDS
  - Sequential file
- Provides the ability to store results and compare old to new
- ISPF or batch
- Reports

DB2 SQLPA – EEE is Parameter Driven

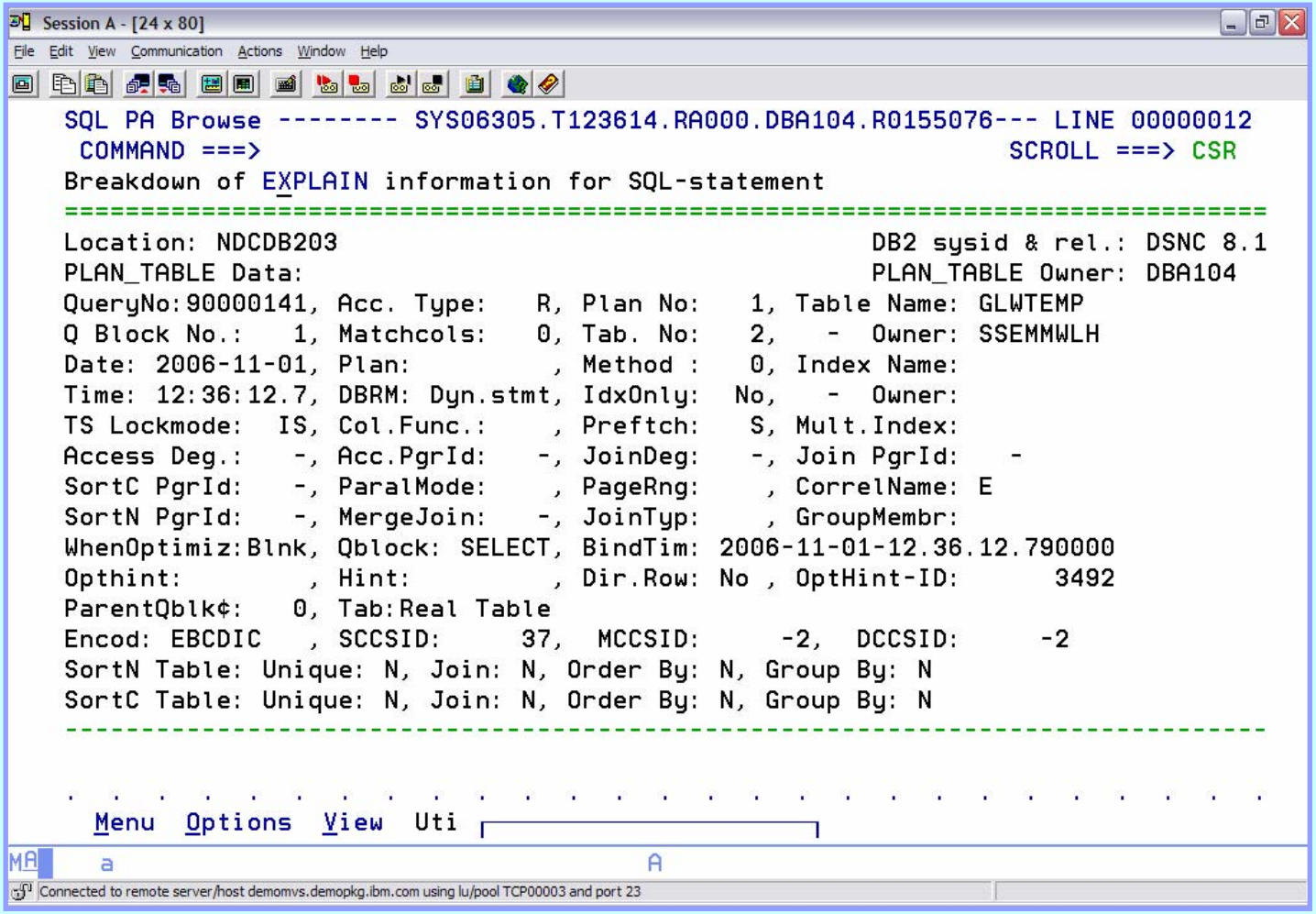
Level of report
Source of input
SQL



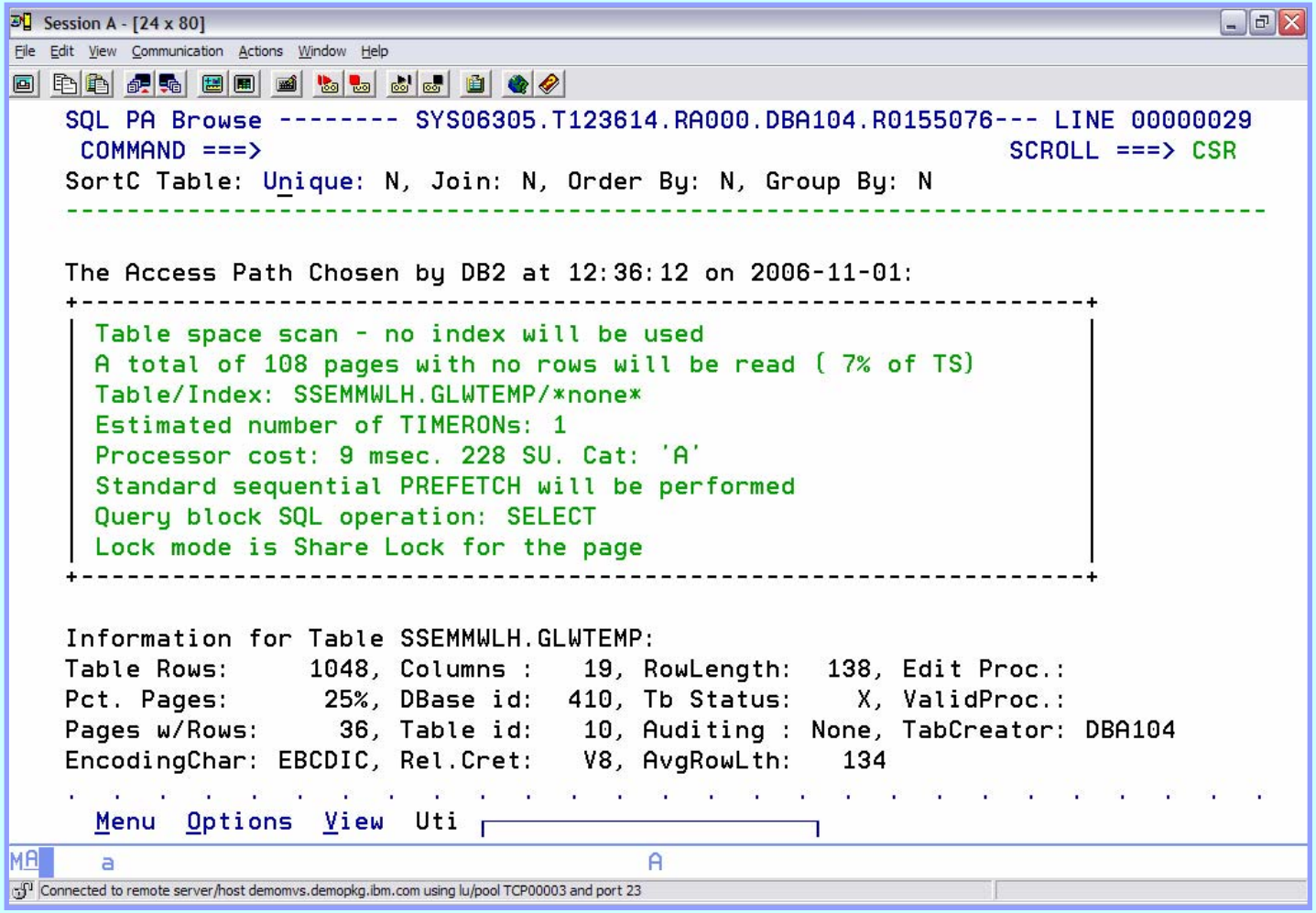
DB2 SQLPA – EEE Report



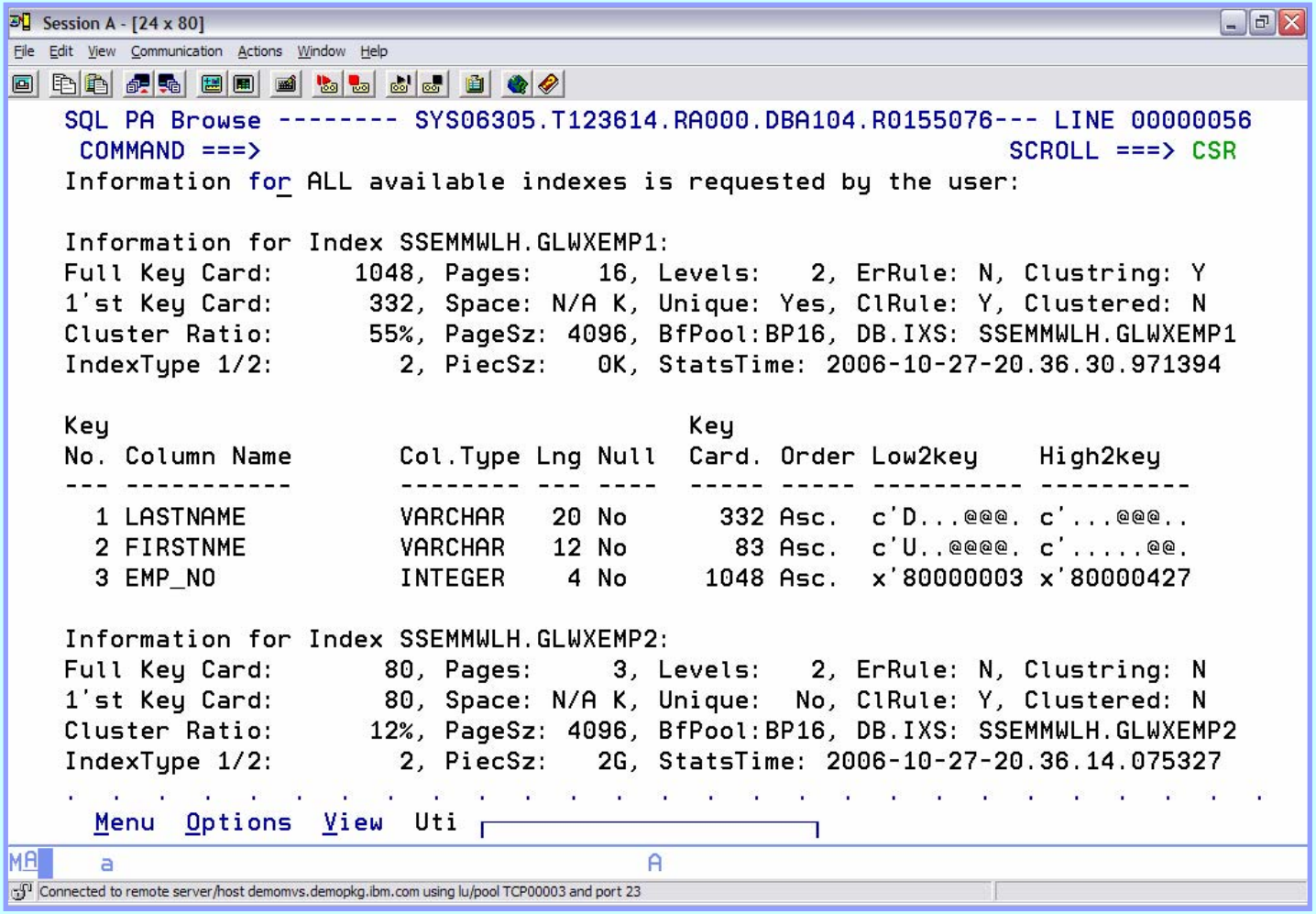
DB2 SQLPA – EEE Report



DB2 SQLPA – EEE Report – Access Path

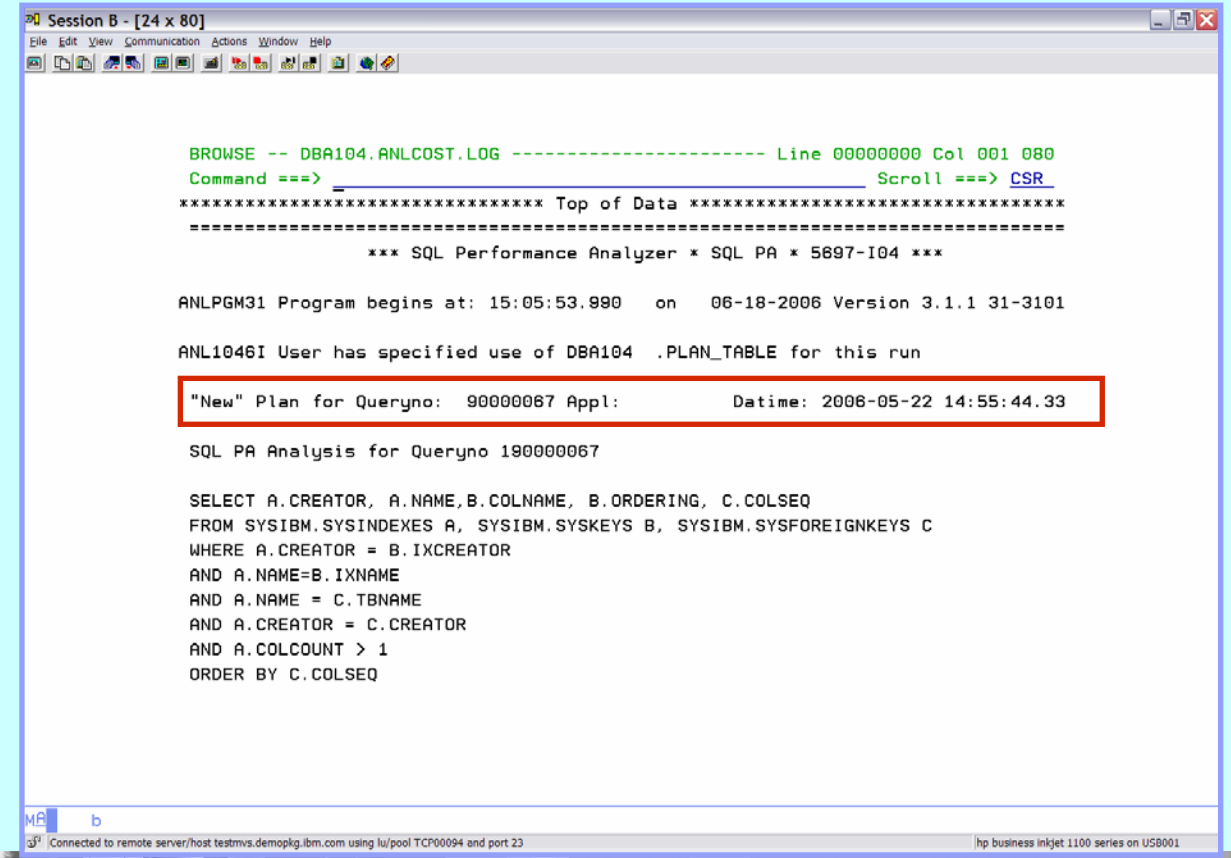


DB2SQLPA – EEE Report – Available Index Info

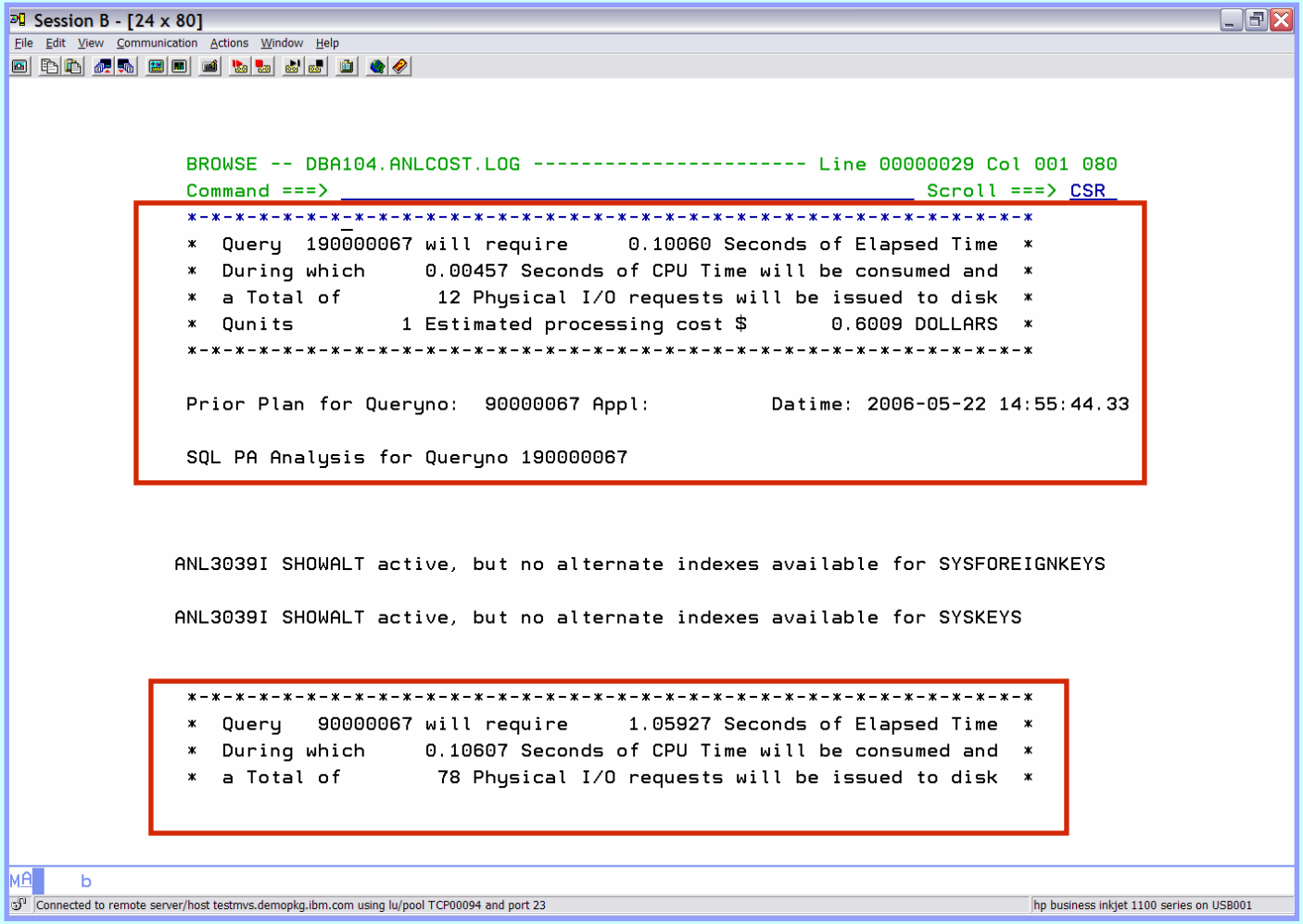


DB2 SQLPA - Compare Old and New Plans

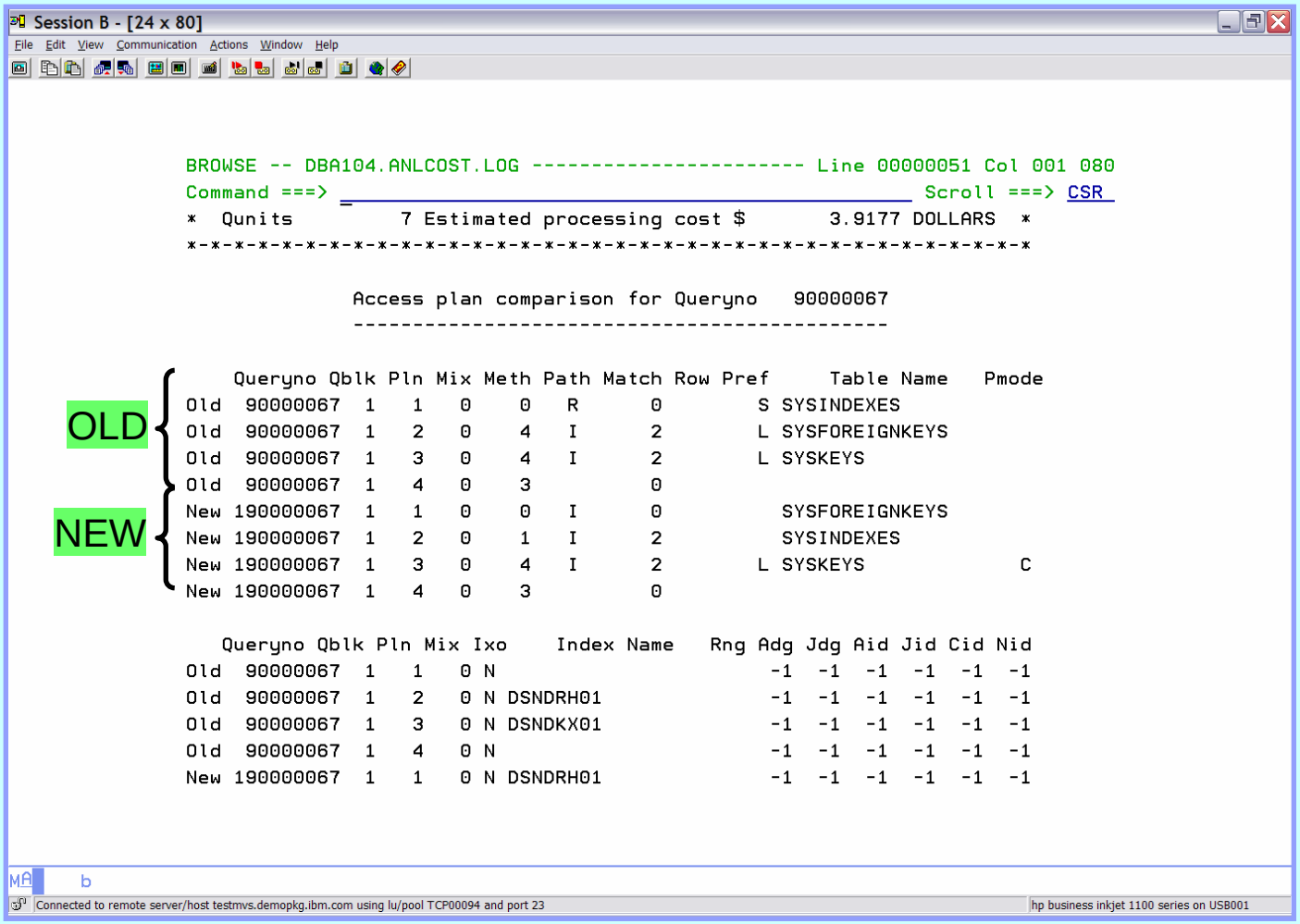
- Compare access paths and costs of SQL plans previously stored by Easy Explain in the EEE path tables with those of the current plan



DB2 SQLPA - Compare Old and New Plans



DB2 SQLPA - Compare Old and New Plans



RECAP DB2 SQL/PA

- DB2 SQL PA provides enhanced explain information for SQL statements
  - DB2 SQL PA
    - SQL input TSO, sequential files, DBRM, QMF
    - Output – multiple reports providing detailed explanation of access paths, provides warnings, alerts, guidelines, recommendations, ..
  - EEE Easy Explain
    - SQL input Plan, Package, DBRM, sequential file, QMF
    - Output – reports providing formatted explanation of access paths

DB2 PATH CHECKER

- Identifies potential access path changes
  - Determine if the access path will change and identify what the change is **before** doing the rebind
  - **Tune the application** to achieve a better access path before the rebind
- Used to avoid performance issues related to these changes
- Chooses course of action before changes affect DB2 production
- Helps DBAs to verify access path changes

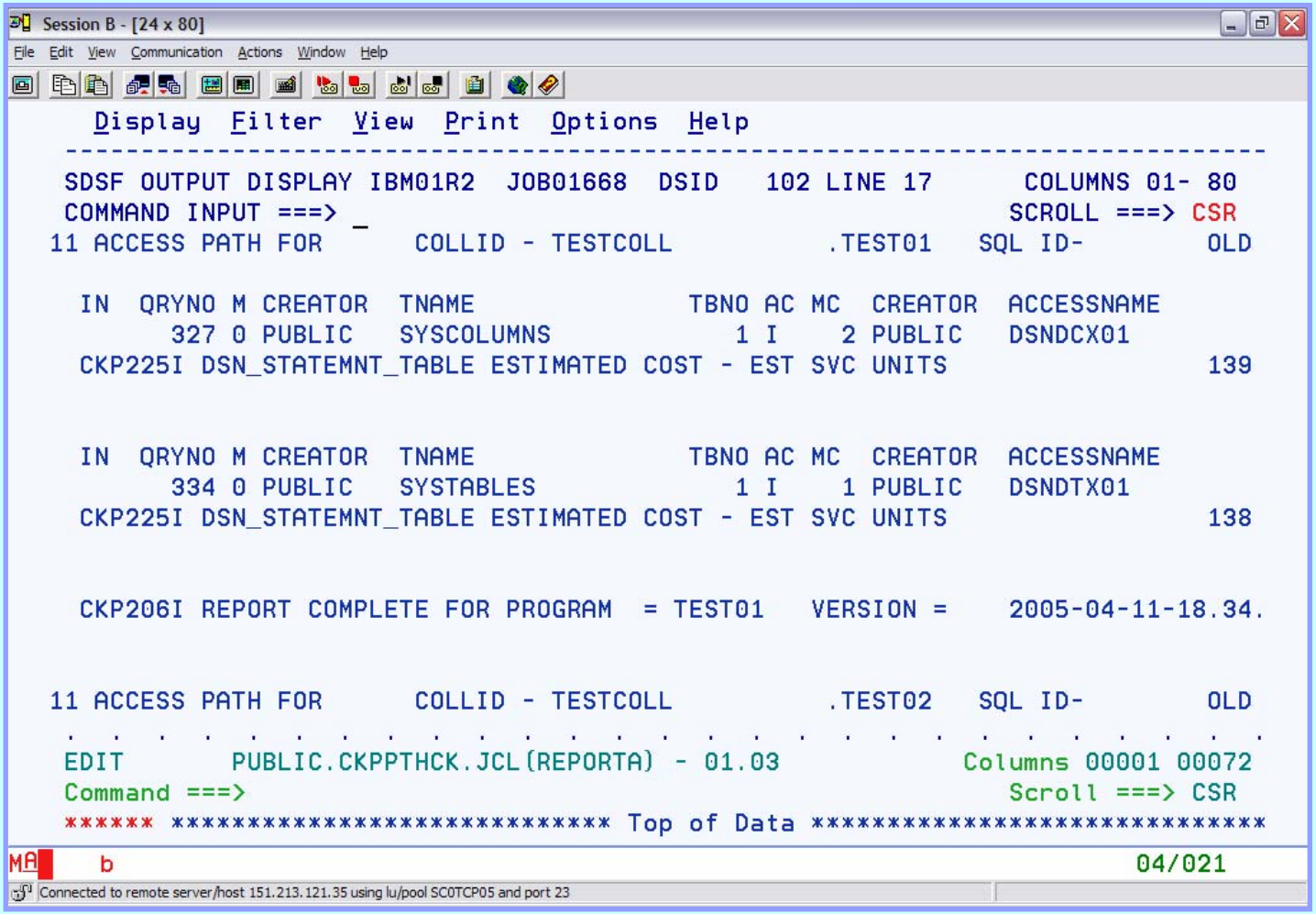
DB2 PATH CHECKER – 3 Major Functions

- **REPORT** – generate an Access Report
  - Can view just the changed access paths
  - Or view all of the access paths
- **TEST** – explain access paths of a DBRM and compare to a previously explained access path
- **COMPARE** – compare the access paths after an explain has been done (ea. in their own plan table)

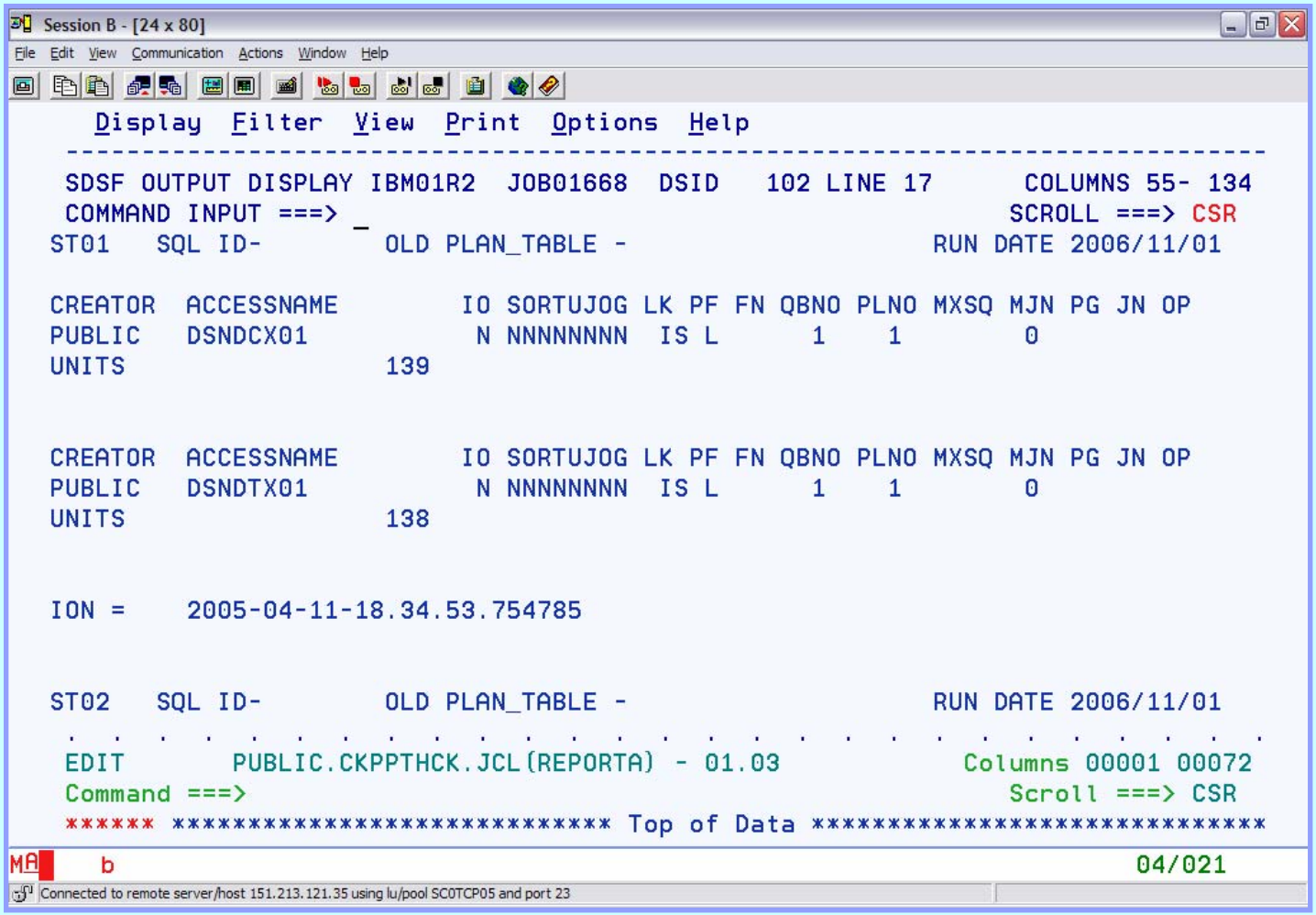
DB2 PATH CHECKER – REPORT Feature

- **Input**: Plan Table
  - Uses Creator-ID to determine the plan_table to read
  - If the corresponding plan table is not populated with the selected plan or package explain data, the program will return no information
- **3 Types of reports**
  - Summary Report (default) → SYSPRINT
  - Detail Report – EXPLAIN → SYSEXPLN
  - Directory Report - Summary of BIND and EXPLAIN Activity

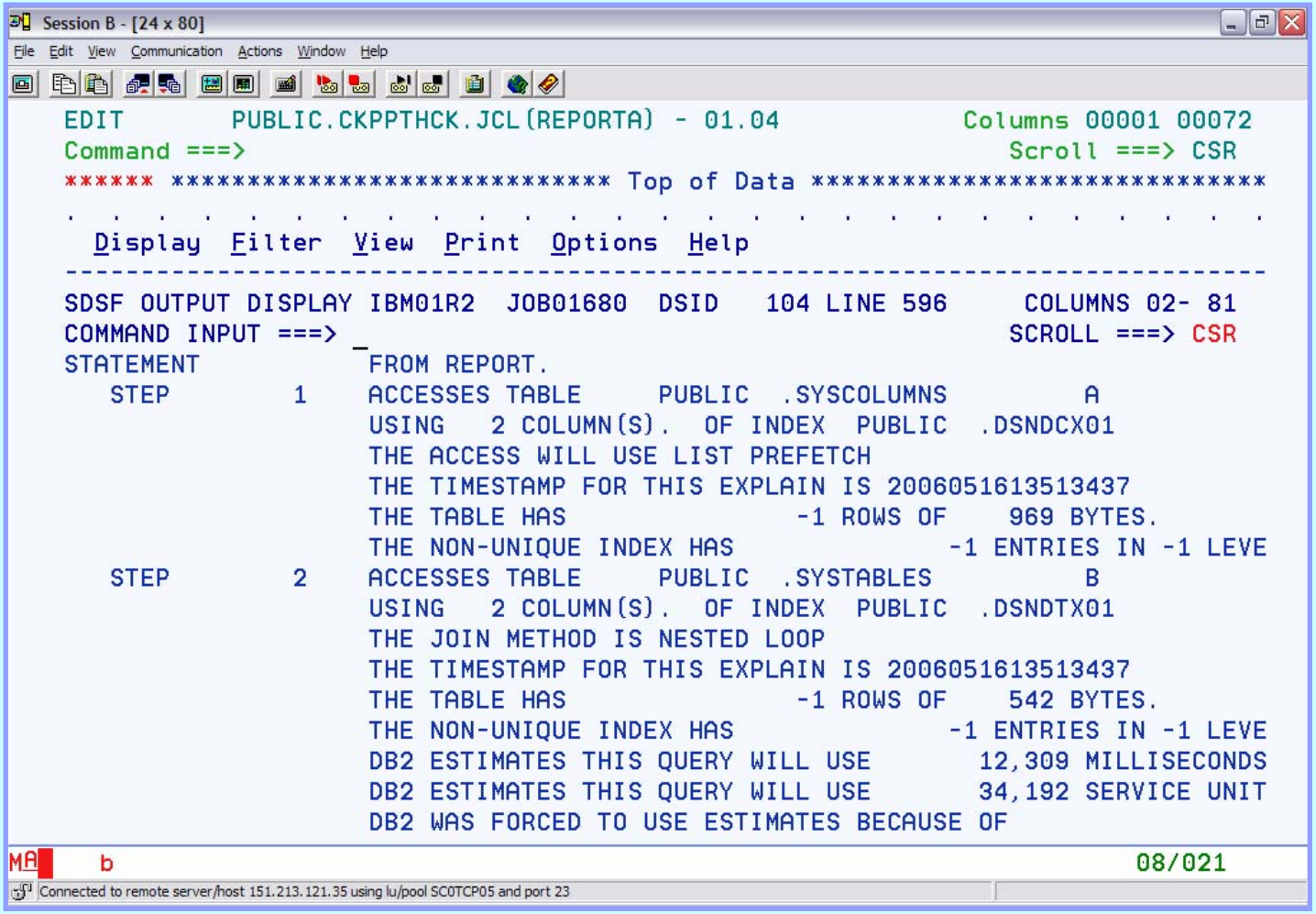
DB2 PATH CHECKER – REPORT Output



DB2 PATH CHECKER – REPORT Output (2)



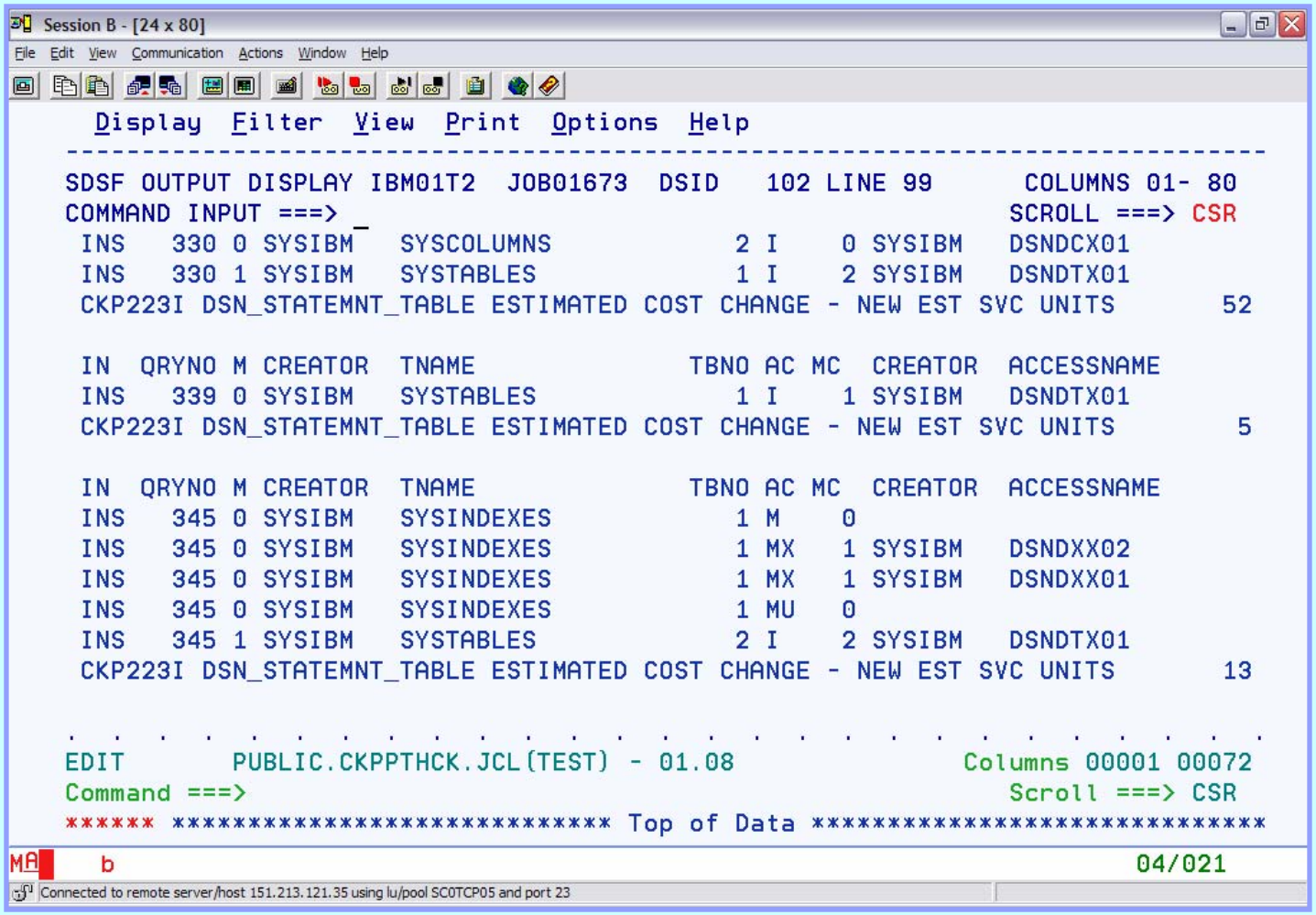
DB2 PATH CHECKER - REPORT Using EXPLAIN Option



DB2 PATH CHECKER – TEST Feature

- **Input:** DBRM
- TEST can be done WITHOUT binding the associate DBRM or package
- Examines each SQL statement (only those that can be explained)
- Performs an explain to determine the access path
- Compares the new access path to the existing access path (in the plan table)
- Generates a report to show the old and new access paths
  - Configure to show all statements
  - Or show only those that have changed

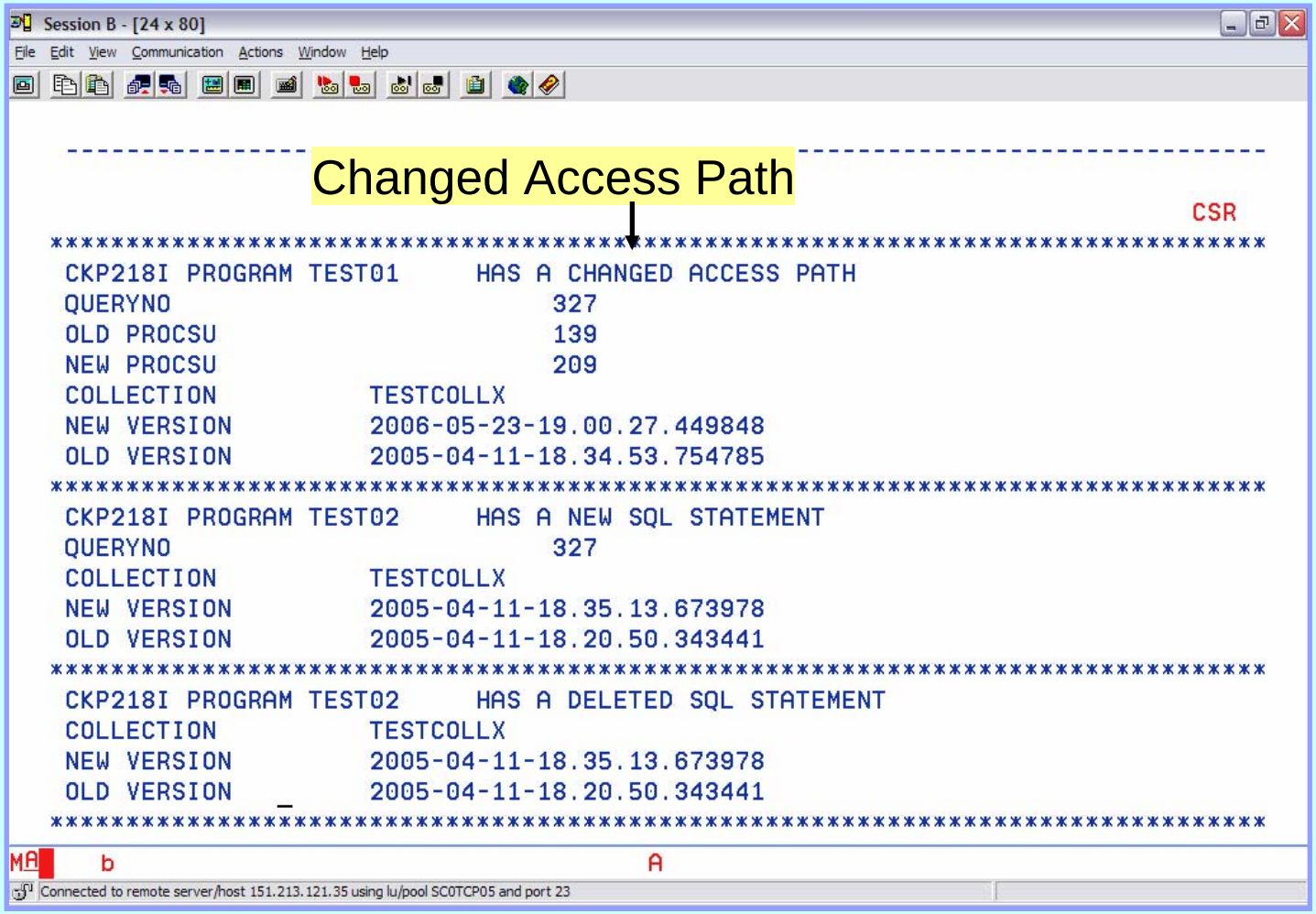
DB2 PATH CHECKER – TEST Output



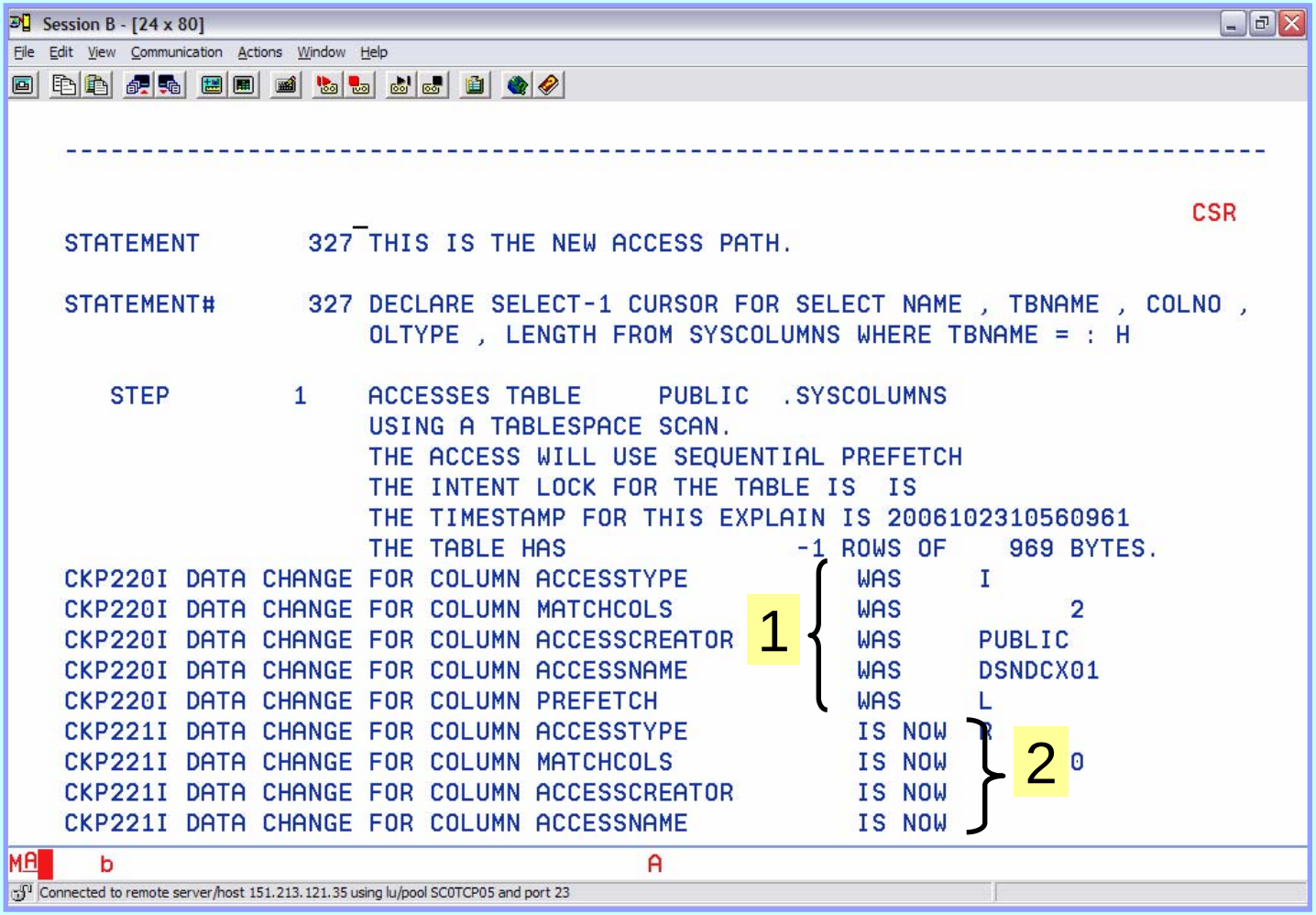
DB2 PATH CHECKER – COMPARE Feature

- The **COMPARE** Command contrast entries for Plans or Packages that exist in PLAN_TABLES
- Comparisons can be done within or between PLAN_TABLES
- The compared PLAN_TABLES can be within an SSID or on different SSIDs where DDF is available
- The default is to compare the most recent explain to the most recent previous entry
- Options are available to set the comparison to specific entries in the PLAN_TABLE

DB2 PATH CHECKER – TEST Output



DB2 PATH CHECKER - COMPARE Report



IBM Information Management software

- www.ibm.com/software/data/tools
- www.redbooks.ibm.com