System I Navigator as a Database Admin Tool

Presented by

Barbara Brown

Director, Technical Services

Alliance Systems & Programming, Inc.

Agenda

- Basic Database Tasks in System i Navigator
- Programming & Debugging
- Performance & Query Optimization
- Visual Explain
- Database Mapping
- Database as an essential component in SOA
- Navigator as a tool in moving toward SOA

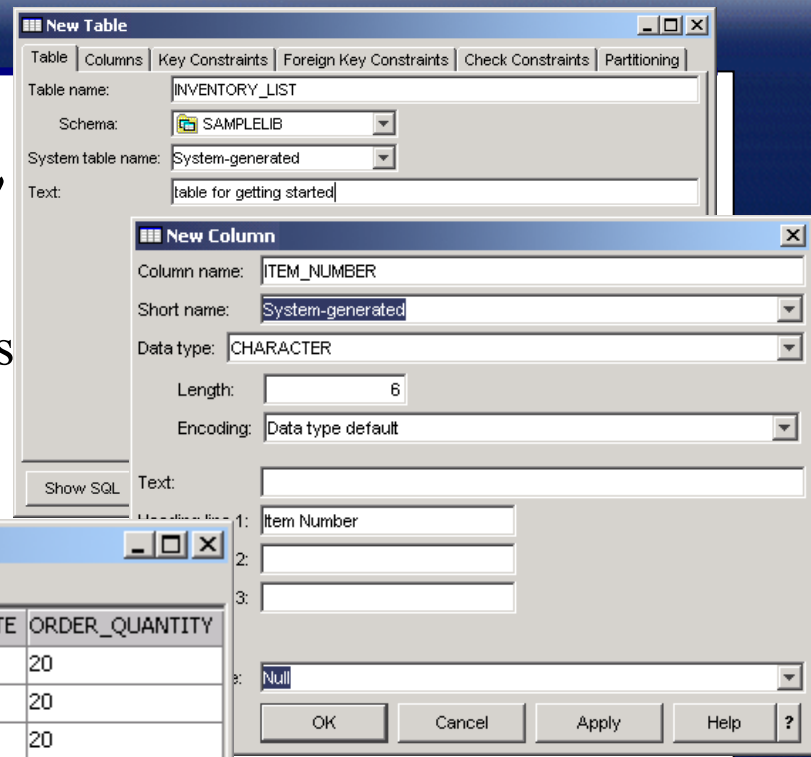
Database Terminology for Die-Hard 400 Folk

- Some Familiar Concepts w new names
  - Schema = Library
  - Table = Physical File
  - Row = Record
  - Column = Field
  - View = Logical File
  - Index = Access Path
- Some New Concepts
  - Relational Integrity – no orphan data, such as an order for a customer that has been deleted from the customer master
  - Constraints -
    - Unique Key
    - Check Constraint
    - Foreign Key – value for column must exist in another table, for example the customer number in the order header must be on the customer master

Basic Database Tasks

Skill and Leadership

- Edit list of schemas displayed
- Create, Delete – schema, table, view, index, constraint
- Define column / copy column definitions
- View, insert, delete or change table contents
- Copy or move a table
- Edit Authority (Permissions)



SAMPLELIB.INVENTORY_LIST

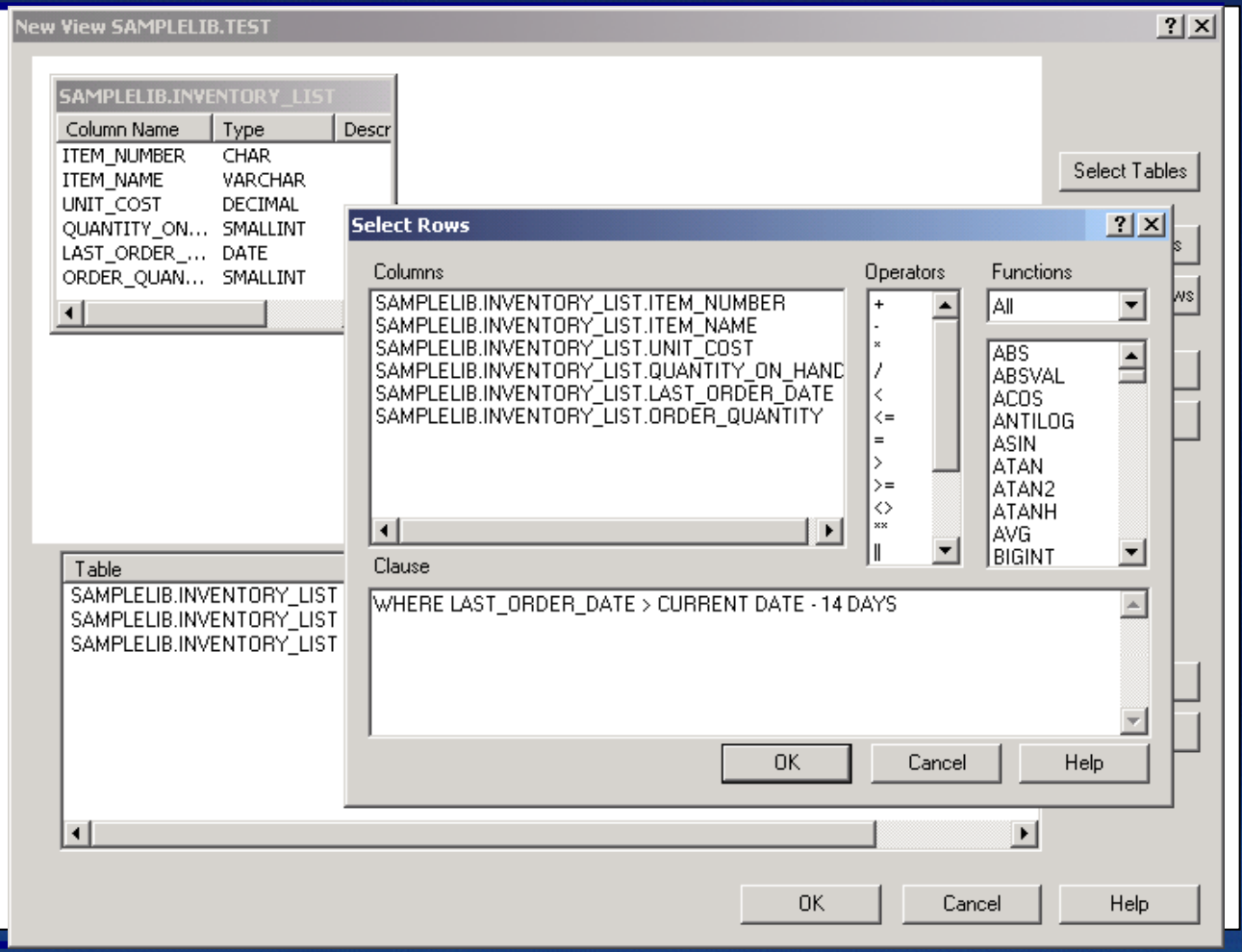
File Edit View Rows Help

ITEM_NUMBER	ITEM_NAME	UNIT_COST	QUANTITY_ON_HAND	LAST_ORDER_DATE	ORDER_QUANTITY
153047	PENCILS, RED	10.00	25		20
229740	LINED TABLETS	1.50	120		20
303476	PAPER CLIPS	2.00	100		20
559343	ENVELOPES, LEGAL	3.00	500		20
544931	UNKNOWN	5.00			20
775298	CHAIRS, SECRETARY	225.00	6		20
073956	PENS, BLACK	20.00	25		20
291124	ENVELOPES, STANDARD	0			20

Basic Database Tasks

Skill and Leadership

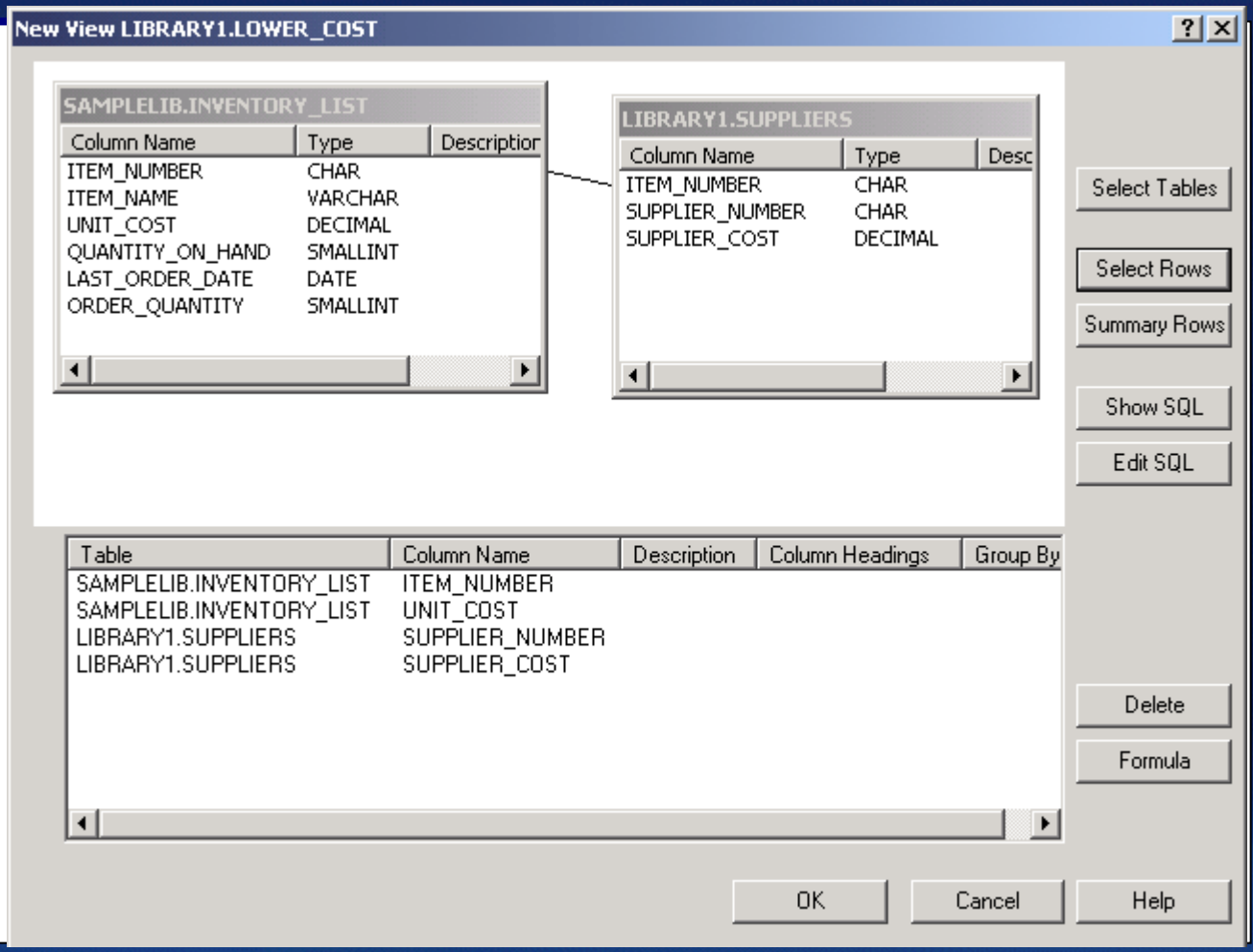
- Create view



Basic Database Tasks

Skill and Leadership

- Create view over multiple tables



Add Constraint

- Unique Key
- Check Constraint
- Foreign Key

Key Constraint Definition - 10.50.0.11(S103483f)

Constraint name: Q_ALLIANCE_ORDERHDR_ORDNUM_00001

Constraint type: Unique key

Column Name	Short Na...	Data Type	Length	Null...	Default Value	Text	CCSID
ORDERNUMBER	ORDNUM	CHARAC...	15	Yes	Null		37

Check Constraint Definition - 10.50.0.11(S103483f)

Constraint name: Q_ALLIANCE_ITEMMASTER_PRDGRP_00001

Check condition:

PRODUCTGROUP IN ('DAIRY' , 'SPICES' , 'CONDIMENT' , 'GROCERY' , 'PRODUCE' , 'CEREAL' , 'PASTA' , 'SUNDRY' , 'BREADS' , 'SANDWICH' , 'PREPAID SERVICE' , 'ICE CREAM' , 'MEATS' , 'HOT DRINKS' , 'COLD DRINKS' , 'COOKIES' , 'CANDY' , 'SNACKS' , 'SODA' , 'PREPARED FOODS' , 'CHEESE' , 'FROZEN' , 'PASTRY')

Close Help ?

Foreign Key Constraint Definition - 10.50.0.11(S103483f)

Constraint name: Q_ALLIANCE_ORDERHDR_CSTNUM_00001

Parent table

Table schema: ALLIANCE

Table name: CUSTOMERMASTER

Key constraint: Q_ALLIANCE_CSTNUM_CSTNUM_0...

Key columns:

Column Name	Short Na...	Data Type	Length	Null...	Defa
CUSTOMERNUMB...	CSTNUM	CHARAC...	15	No	"

Column Name	Short Na...	Data Type	Length	Null...	Default Value	Text	CCSID	Is
CUSTOMERNUMB...	CSTNUM	CHARAC...	15	Yes	Null		37	

Action upon delete: Restrict

Action upon update: Restrict

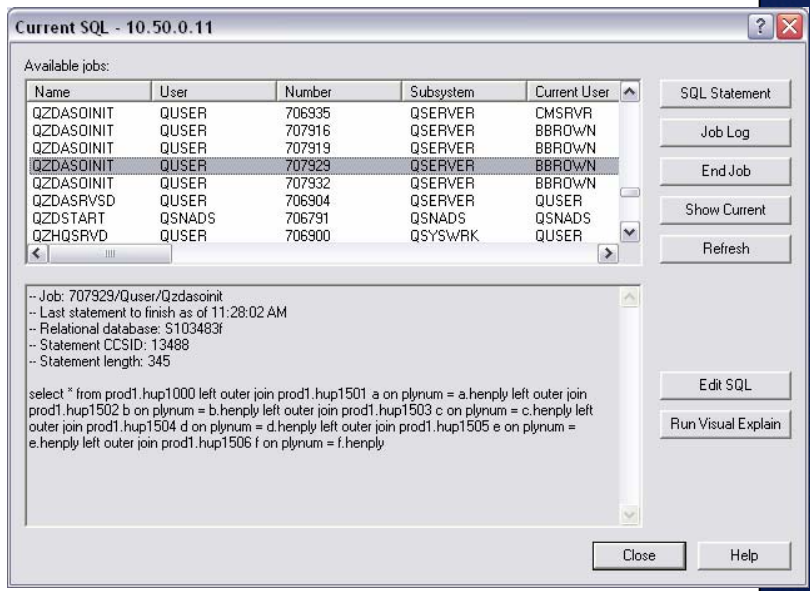
Close Help ?

Triggers

- Add Trigger – system (ADDPFTRG) or SQL (CREATE TRIGGER…)
- Uses:
  - Enforce business rules
  - Validate input data
  - Generate a unique value for a newly inserted row on a different file
  - Write to other files for audit trail purposes
  - Query from other files for cross-referencing purposes
  - Access system functions
  - Replicate data to different files to achieve data consistency
- Benefits:
  - Faster application development.
  - Global enforcement of business rules.
  - Easier maintenance.
  - Improve performance in client/server environment.

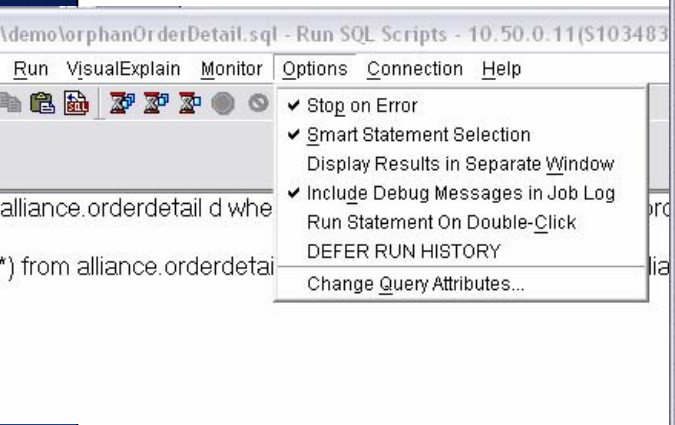
Programming & Debugging

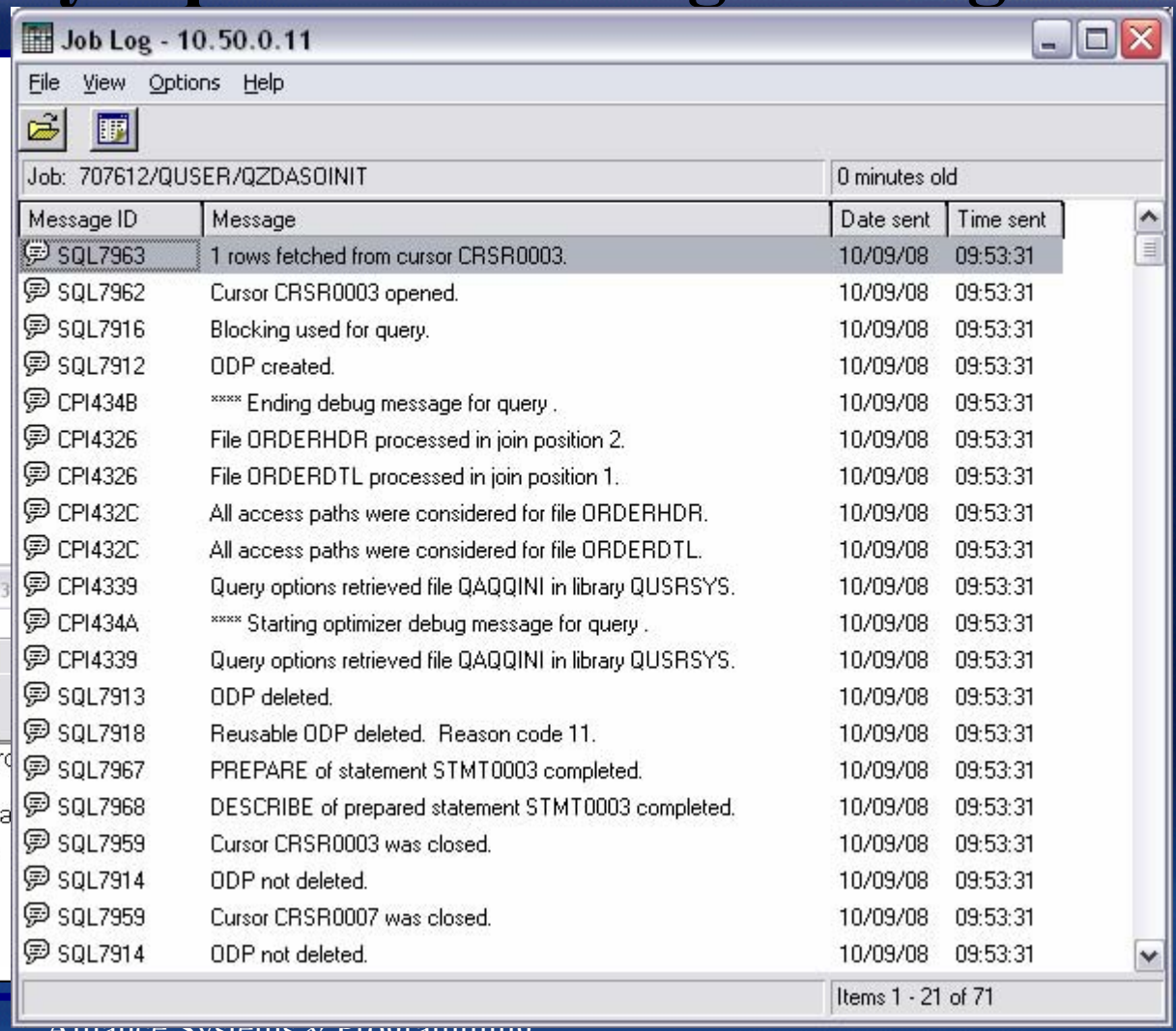
- Display Locked Rows of a Table
- Reorganize a Table
- Display current SQL for a job
- Work with Journals
  - Creating a journal
  - Creating a journal receiver
  - Adding a remote journal
  - Removing a remote journal
  - Activating a remote journal
  - Deactivating a remote journal
  - Displaying journal information
  - Swapping journal receivers
  - Starting and stopping a journal



Performance and Query Optimization: Query Optimizer Debug Messages

- Query optimizer debug messages
  - RUN SQL scripts
  - Options – include Debug Messages in Job Log
  - View - JobLog



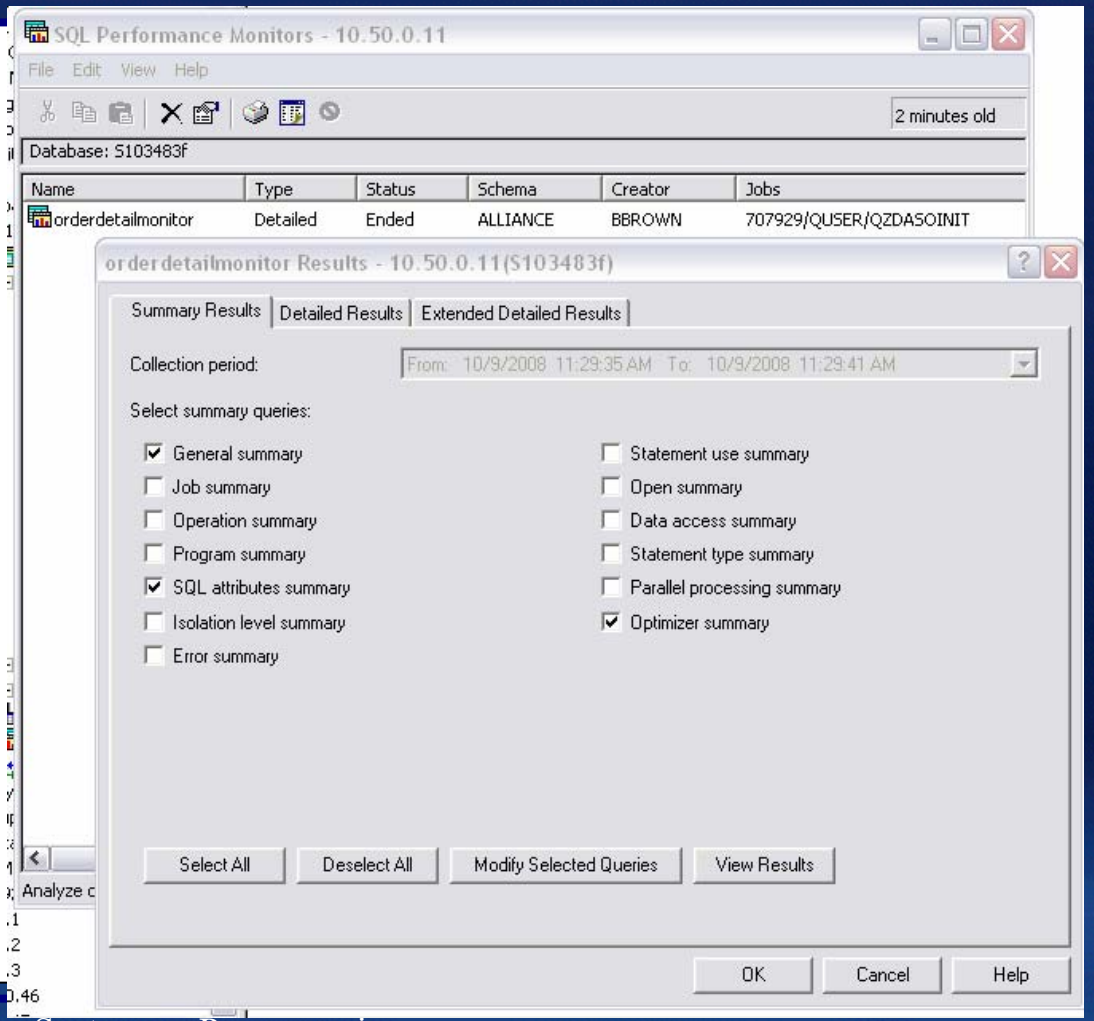


Performance and Query Optimization: Explain SQL

- Explain SQL for functions, stored procedures, triggers, SQL packages, and programs (PRTSQLINF)
  - Sql statements
  - Access paths used
  - Command parameters

Performance and Query Optimization:
SQL Performance Monitor

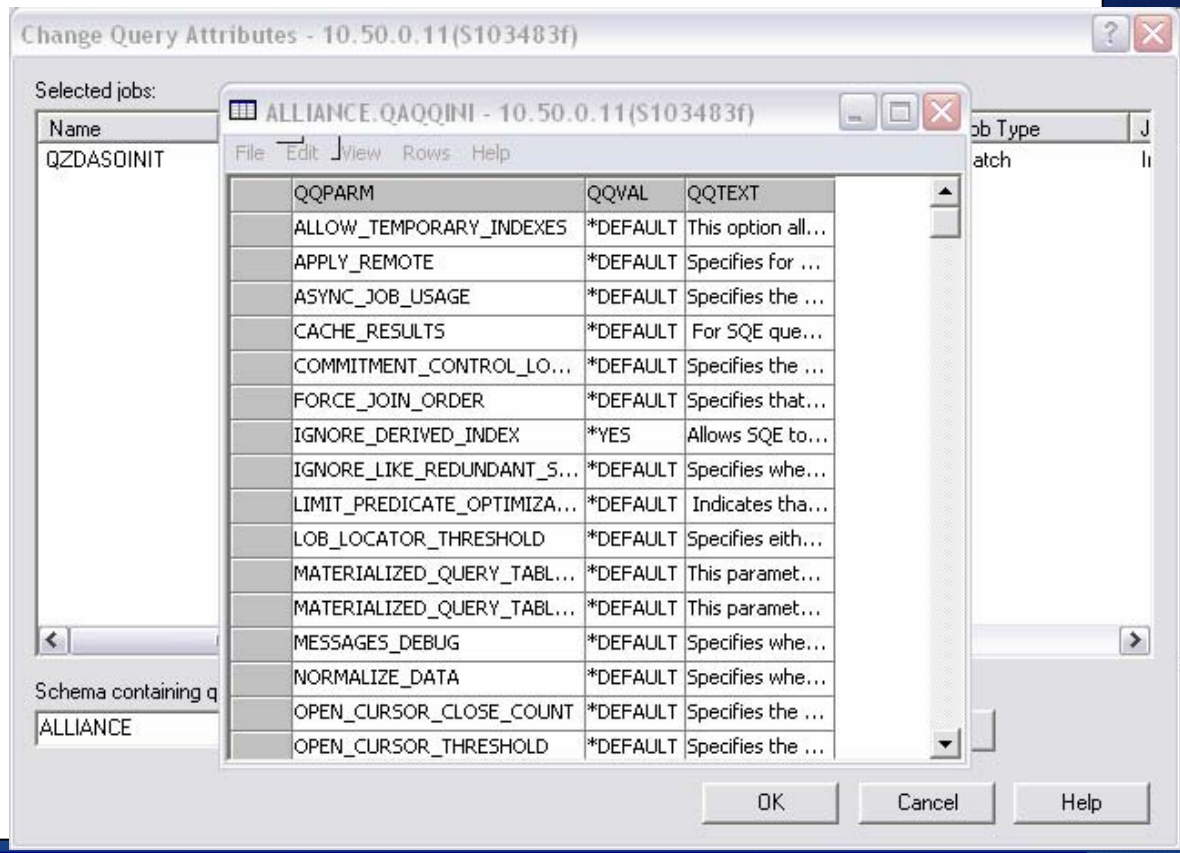
- SQL Performance Monitor (STRDBMON)
  - Collect performance data on a job
  - Request analysis later



Performance and Query Optimization:
Query Attributes

- Change attributes of queries (CHGQRYA

  - QAQQINI can be unique for a job
  - IGNORE__DERIVED_INDEX
  - QUERY_TIME_LIMIT
  - OPEN_CURSOR_THRESHOLD
  - PARALLEL_DEGREE



Performance and Query Optimization: Statistics

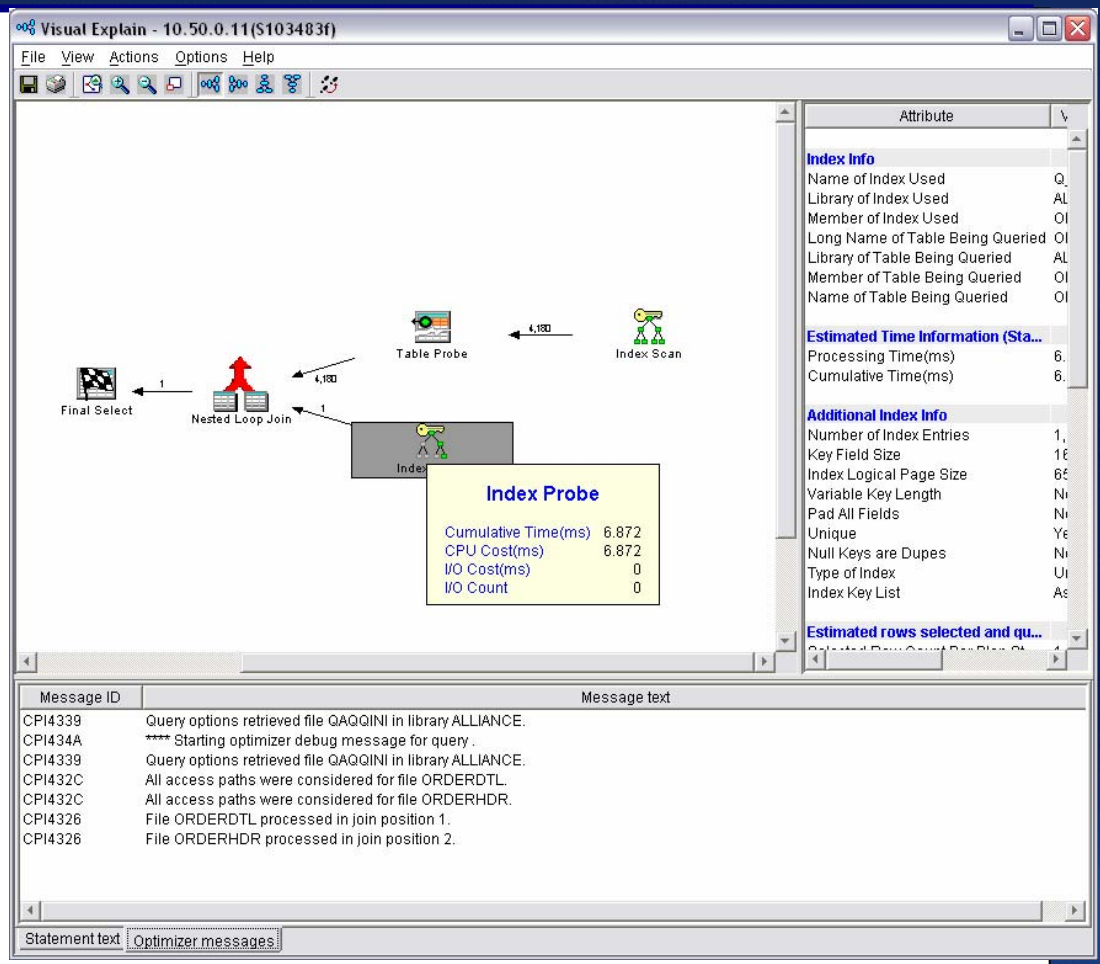
- Statistics automatically collected as system resources available
  - **Cardinality of values**
    number of unique or distinct occurrences of a specific value in a single column or multiple columns of a table
  - **Selectivity**
    how many rows will be selected by any given selection predicate or combination of predicates.
  - **Frequent values**
    Top 100 most frequent values of a column (excluding default and NULL values) and how frequently each value occurs.
  - **Metadata information**
    total number of rows in the table, indexes that exist over the table, and which indexes would be useful for implementing the particular query.
  - **Estimate of IO operation**
    estimate of the amount of IO operations required to process the table or the identified index.
- Statistics manager also returns confidence level to query optimizer

Performance and Query Optimization:
Indexes vs. column statistics

- Column statistics can only be used by SQL Query Engine (SQE). For Classic Query Engine (CQE), all statistics are retrieved from indexes.
- Indexes are permanent objects updated when changes to underlying table occur.
- Indexes more current than column statistics for constantly changing data, but require more overhead.
- When new indexes become available that are candidate for implementing query, Optimizer will re-optimize the query
- When new or refreshed column statistics are available, the Statistics Manager will interrogate immediately. Re-optimization will occur only if the answers are significantly different from before.
- Accessing column statistics to answer questions is faster than trying to obtain these answers from indexes.
- Finally, column statistics can be used only for query optimization. They cannot be used for the actual implementation of a query, whereas indexes can be used for both.

Performance and Query Optimization:
Visual Explain

- Main window contains a query graph that displays the implementation of an SQL statement.
  - both static and dynamic SQL statements.
  - supports SELECT, INSERT, UPDATE, and DELETE.
  - icons represent different operations that occur during implementation.
- Lower portion shows the SQL statement being graphed.
- Click Optimizer Messages tab to view debug messages (only available if started from Run SQL Scripts)
- Query attributes displayed on the right pane.



Performance and Query Optimization:
Visual Explain

- **Information about each operation (icon) in the query graph**:
  - order of operations shown by connecting arrows
  - Double arrows indicates parallelism used to process operation
  - Crossed lines indicate hash tables were shared
- Select icon to view **Attributes** table in the right pane. Or right-click and select **Help**.
- To view information about the environment, click an icon and then select **Display query environment** from the **Action** menu.
- **Highlight expensive icons (View** menu) – highlights problem areas by processing time or number of rows.
- **Statistics and index advisor (Action** menu): optimizer can determine if statistics need to be created or refreshed, or if an index could make the query run faster.
- **Predicate implementation of the query**: Visual explain allows you to view the implementation of query predicates (represented by a blue plus sign next to an icon).
- **Highlight LPG (View** menu) - Look Ahead Predicate Generation can minimize the random I/O costs of a join.
- **Basic and full information in the graph**: **(Graph Detail** in **Options** menu)

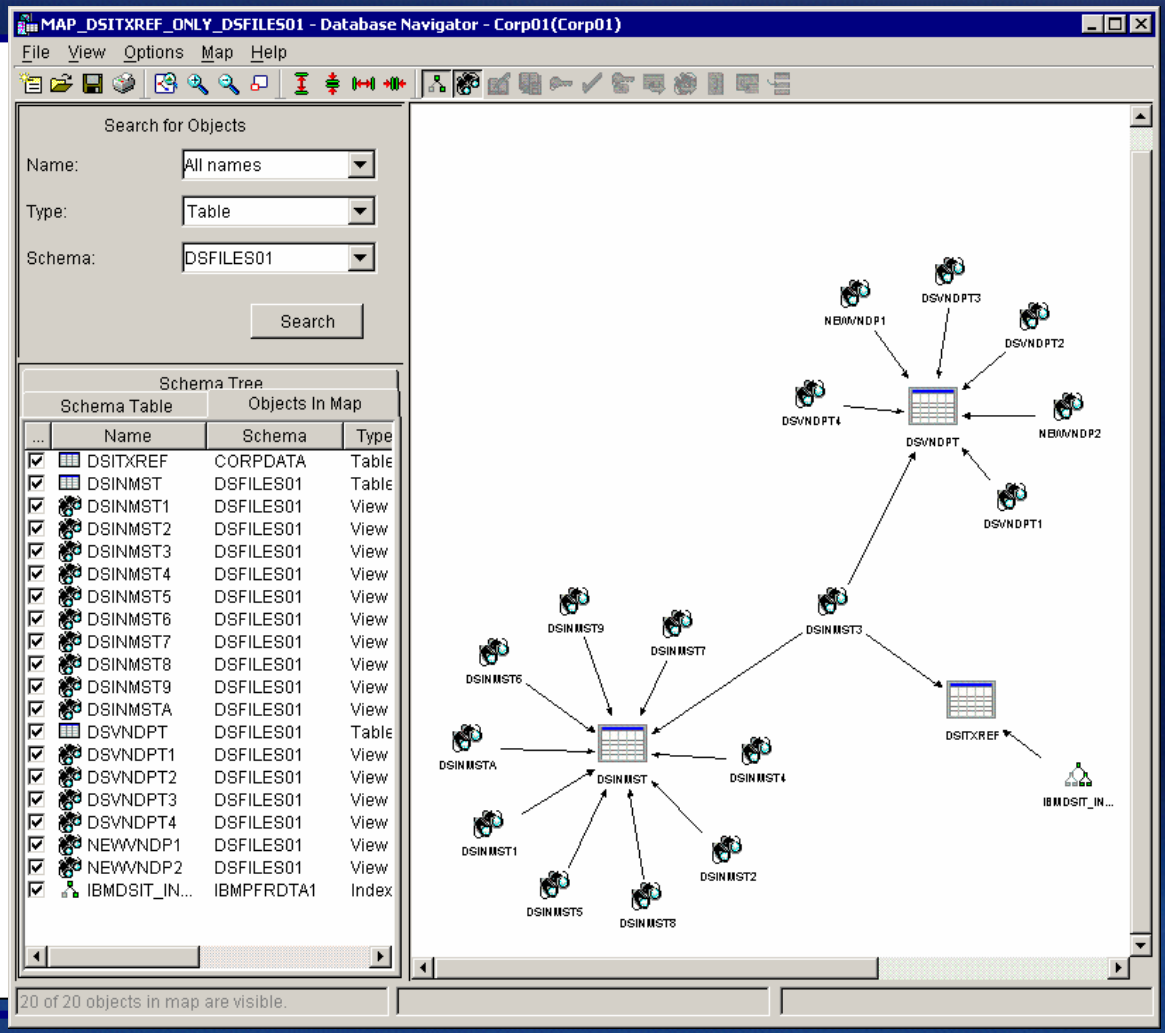
Query Optimization Tools

Query optimization tools: Comparison table

PRTSQLINF	STRDBG or CHGQRYA	File-based monitor	Memory -Based Monitor	Visual Explain
Available without running query (after access plan has been created)	Only available when the query is run	Only available when the query is run	Only available when the query is run	Only available when the query is explained
Displayed for all queries in SQL program, whether executed or not	Displayed only for those queries which are executed	Displayed only for those queries which are executed	Displayed only for those queries which are executed	Displayed only for those queries that are explained
Information about host variable implementation	Limited information about the implementation of host variables	All information about host variables, implementation, and values	All information about host variables, implementation, and values	All information about host variables, implementation, and values
Available only to SQL users with programs, packages, or service programs	Available to all query users (OPNQRYF, SQL, QUERY/400)	Available to all query users (OPNQRYF, SQL, QUERY/400)	Available only to SQL interfaces	Available through iSeries™ Navigator Database and API interface
Messages are printed to spool file	Messages is displayed in job log	Performance rows are written to database table	Performance information is collected in memory and then written to database table	Information is displayed visually through iSeries Navigator
Easier to tie messages to query with subqueries or unions	Difficult to tie messages to query with subqueries or unions	Uniquely identifies every query, subquery and materialized view	Repeated query requests are summarized	Easy to view implementation of the query and associated information

Database Mapping

- Visually depict relationships of database objects on your system.
- right-click **Database Navigator Maps** and select **New > Map**.
- Drag or double click objects to include in the map
- click **Database Navigator Maps** to display a list of existing maps in the right pane.



Service Oriented Architecture

- *Service Oriented Architecture* – a plan or structure for a system in which the components perform discrete well-defined units of work for requestors through well-defined interfaces
  - *Modular* – break down complex processes into components, only one service for each function
  - *Encapsulated* – hide complexity of implementation within modules with well defined interfaces – requester is not concerned about details of how a service is implemented
  - *Loosely coupled* – modules connected by simple interface – no technology dependence
  - *Composable* – modules may be assembled to create other services
  - *Coarse grained* —service provides complete business function – coarseness determined by reusability.
  - *Stateless* — each request is treated independent from what came before or what will come next – (does not prohibit passing of state-related information in parameters)

Toward SOA

- Many '400' shops still treat their database like a set of flat files
  - No referential integrity
  - Inconsistent data formats
  - Fields 're-used' to avoid having to recompile many RPG programs
  - Redundant physical and logical files
  - Code around bad data
- Use database mapping and SQL performance monitor to identify worst offenders in terms of space, performance, redundancy, etc.

Toward SOA

- Use Visual Explain to tune worst offenders
- Turning on referential integrity without necessary advance work could break too much
- Triggers could be used to
  - move toward referential integrity in a stepwise fashion
  - provide data services layer for master data

Master Data Management (MDM)

- Processes and tools to ensure data quality across non-transactional entities of an organization
  - Complete – all necessary information included
  - Standard formats – ex) phone numbers, dates
  - Consistent – no contradictory information
  - Avoid Duplication (multiple versions leads to contradictions)
  - Accurate – correctly reflects real world
  - Data Integrity – no broken relationships
- Partner to SOA
  - Data Services Layer avoids needs to duplicate code for managing data