



*Skill and Leadership*

# **System I Navigator as a Database Admin Tool**

Presented by

**Barbara Brown**

Director, Technical Services

Alliance Systems & Programming, Inc.



*Skill and Leadership*

# 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



*Skill and Leadership*

# 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

- 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)

The image shows two overlapping dialog boxes from a database management tool. The 'New Table' dialog is in the background, showing 'Table name: INVENTORY\_LIST', 'Schema: SAMPLELIB', and 'Text: table for getting started'. The 'New Column' dialog is in the foreground, showing 'Column name: ITEM\_NUMBER', 'Short name: System-generated', 'Data type: CHARACTER', 'Length: 6', and 'Encoding: Data type default'. It also has a 'Text' field and a 'Null' checkbox.

The screenshot shows a table view for 'SAMPLELIB.INVENTORY\_LIST'. The table has the following columns and data:

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

- Create view

New View SAMPLELIB.TEST

Column Name	Type	Descr
ITEM_NUMBER	CHAR	
ITEM_NAME	VARCHAR	
UNIT_COST	DECIMAL	
QUANTITY_ON...	SMALLINT	
LAST_ORDER_...	DATE	
ORDER_QUAN...	SMALLINT	

Select Tables

Select Rows

Columns

- SAMPLELIB.INVENTORY\_LIST.ITEM\_NUMBER
- SAMPLELIB.INVENTORY\_LIST.ITEM\_NAME
- SAMPLELIB.INVENTORY\_LIST.UNIT\_COST
- SAMPLELIB.INVENTORY\_LIST.QUANTITY\_ON\_HAND
- SAMPLELIB.INVENTORY\_LIST.LAST\_ORDER\_DATE
- SAMPLELIB.INVENTORY\_LIST.ORDER\_QUANTITY

Operators

- +
- 
- \*
- /
- <
- <=
- =
- >
- >=
- <>
- \*\*
- ||

Functions

- All
- ABS
- ABSVAL
- ACOS
- ANTILOG
- ASIN
- ATAN
- ATAN2
- ATANH
- AVG
- BIGINT

Clause

WHERE LAST\_ORDER\_DATE > CURRENT DATE - 14 DAYS

Table

- SAMPLELIB.INVENTORY\_LIST
- SAMPLELIB.INVENTORY\_LIST
- SAMPLELIB.INVENTORY\_LIST

OK Cancel Help

OK Cancel Help



# Basic Database Tasks

- Create view over multiple tables

New View LIBRARY1.LOWER\_COST

Column Name	Type	Description
ITEM_NUMBER	CHAR	
ITEM_NAME	VARCHAR	
UNIT_COST	DECIMAL	
QUANTITY_ON_HAND	SMALLINT	
LAST_ORDER_DATE	DATE	
ORDER_QUANTITY	SMALLINT	

Column Name	Type	Desc
ITEM_NUMBER	CHAR	
SUPPLIER_NUMBER	CHAR	
SUPPLIER_COST	DECIMAL	

Table	Column Name	Description	Column Headings	Group By
SAMPLELIB.INVENTORY_LIST	ITEM_NUMBER			
SAMPLELIB.INVENTORY_LIST	UNIT_COST			
LIBRARY1.SUPPLIERS	SUPPLIER_NUMBER			
LIBRARY1.SUPPLIERS	SUPPLIER_COST			

Buttons: Select Tables, Select Rows, Summary Rows, Show SQL, Edit SQL, Delete, Formula, OK, Cancel, Help



# 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

The screenshot shows a window titled "Current SQL - 10.50.0.11". It contains a table of "Available jobs" and a text area for the "SQL Statement".

Name	User	Number	Subsystem	Current User
QZDASOINIT	QUSER	706935	QSERVER	CMSVR
QZDASOINIT	QUSER	707916	QSERVER	BBROWN
QZDASOINIT	QUSER	707919	QSERVER	BBROWN
QZDASOINIT	QUSER	707929	QSERVER	BBROWN
QZDASOINIT	QUSER	707932	QSERVER	BBROWN
QZDASRVSD	QUSER	706904	QSERVER	QUSER
QZDSTART	QSNADS	706791	QSNADS	QSNADS
QZHQSRVD	QUSER	706900	QSYSWRK	QUSER

SQL Statement:  
-- Job: 707929/Quser/Qzdasoinit  
-- Last statement to finish as of 11:28:02 AM  
-- Relational database: S103483f  
-- Statement CCSID: 13488  
-- Statement length: 345  
select \* from prod1.hup1000 left outer join prod1.hup1501 a on plynum = a.henply left outer join prod1.hup1502 b on plynum = b.henply left outer join prod1.hup1503 c on plynum = c.henply left outer join prod1.hup1504 d on plynum = d.henply left outer join prod1.hup1505 e on plynum = e.henply left outer join prod1.hup1506 f on plynum = f.henply



# Performance and Query Optimization: Query Optimizer Debug Messages

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

Job Log - 10.50.0.11

File View Options Help

Job: 707612/QUSER/QZDASOINIT 0 minutes old

Message ID	Message	Date sent	Time sent
SQL7963	1 rows fetched from cursor CRSR0003.	10/09/08	09:53:31
SQL7962	Cursor CRSR0003 opened.	10/09/08	09:53:31
SQL7916	Blocking used for query.	10/09/08	09:53:31
SQL7912	ODP created.	10/09/08	09:53:31
CPI434B	**** Ending debug message for query .	10/09/08	09:53:31
CPI4326	File ORDERHDR processed in join position 2.	10/09/08	09:53:31
CPI4326	File ORDERDTL processed in join position 1.	10/09/08	09:53:31
CPI432C	All access paths were considered for file ORDERHDR.	10/09/08	09:53:31
CPI432C	All access paths were considered for file ORDERDTL.	10/09/08	09:53:31
CPI4339	Query options retrieved file QAQQINI in library QUSRSYS.	10/09/08	09:53:31
CPI434A	**** Starting optimizer debug message for query .	10/09/08	09:53:31
CPI4339	Query options retrieved file QAQQINI in library QUSRSYS.	10/09/08	09:53:31
SQL7913	ODP deleted.	10/09/08	09:53:31
SQL7918	Reusable ODP deleted. Reason code 11.	10/09/08	09:53:31
SQL7967	PREPARE of statement STMT0003 completed.	10/09/08	09:53:31
SQL7968	DESCRIBE of prepared statement STMT0003 completed.	10/09/08	09:53:31
SQL7959	Cursor CRSR0003 was closed.	10/09/08	09:53:31
SQL7914	ODP not deleted.	10/09/08	09:53:31
SQL7959	Cursor CRSR0007 was closed.	10/09/08	09:53:31
SQL7914	ODP not deleted.	10/09/08	09:53:31

Items 1 - 21 of 71

demo\orphanOrderDetail.sql - Run SQL Scripts - 10.50.0.11(S103483)

Run Visual Explain Monitor Options Connection Help

- ✓ Stop on Error
- ✓ Smart Statement Selection
- Display Results in Separate Window
- ✓ Include Debug Messages in Job Log
- Run Statement On Double-Click
- DEFER RUN HISTORY
- Change Query Attributes...



*Skill and Leadership*

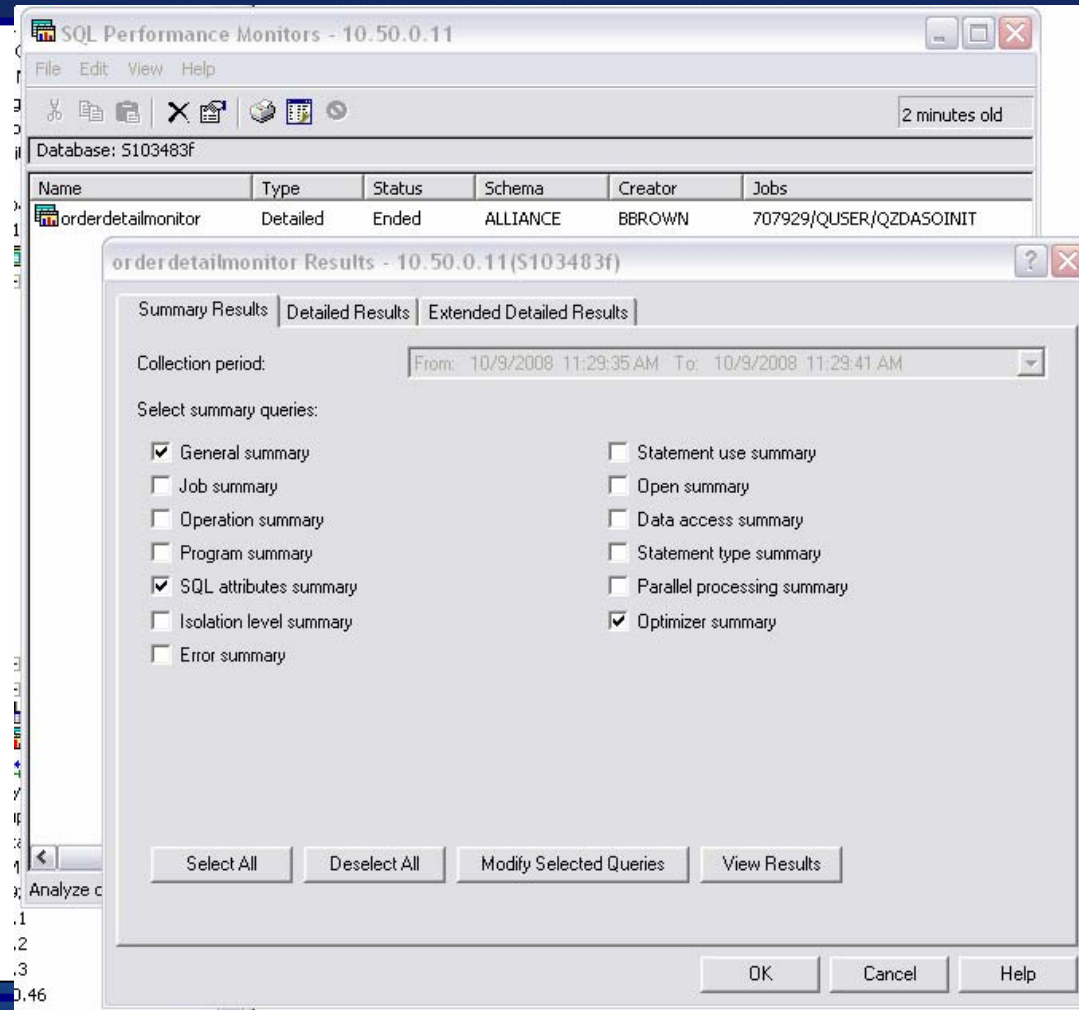
# Performance and Query Optimization: Explain SQL

- 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)

- QAAQINI can be unique for a job
- IGNORE\_\_DERIVED\_INDEX
- QUERY\_TIME\_LIMIT
- OPEN\_CURSOR\_THRESHOLD
- PARALLEL\_DEGREE

QPARM	QVAL	QQTEXT
ALLOW_TEMPORARY_INDEXES	*DEFAULT	This option all...
APPLY_REMOTE	*DEFAULT	Specifies for ...
ASYNC_JOB_USAGE	*DEFAULT	Specifies the ...
CACHE_RESULTS	*DEFAULT	For SQE que...
COMMITMENT_CONTROL_LO...	*DEFAULT	Specifies the ...
FORCE_JOIN_ORDER	*DEFAULT	Specifies that...
IGNORE_DERIVED_INDEX	*YES	Allows SQE to...
IGNORE_LIKE_REDUNDANT_S...	*DEFAULT	Specifies whe...
LIMIT_PREDICATE_OPTIMIZA...	*DEFAULT	Indicates tha...
LOB_LOCATOR_THRESHOLD	*DEFAULT	Specifies eith...
MATERIALIZED_QUERY_TABL...	*DEFAULT	This paramet...
MATERIALIZED_QUERY_TABL...	*DEFAULT	This paramet...
MESSAGES_DEBUG	*DEFAULT	Specifies whe...
NORMALIZE_DATA	*DEFAULT	Specifies whe...
OPEN_CURSOR_CLOSE_COUNT	*DEFAULT	Specifies the ...
OPEN_CURSOR_THRESHOLD	*DEFAULT	Specifies the ...



*Skill and Leadership*

# 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



*Skill and Leadership*

# 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.

The screenshot shows the Visual Explain tool interface. The main window displays a query graph with nodes for 'Final Select', 'Nested Loop Join', 'Table Probe', and 'Index Scan'. A tooltip for 'Index Probe' is visible, showing performance metrics:

Index Probe	
Cumulative Time(ms)	6.872
CPU Cost(ms)	6.872
I/O Cost(ms)	0
I/O Count	0

The right pane displays query attributes:

Index Info	
Name of Index Used	Q...
Library of Index Used	AL
Member of Index Used	OI
Long Name of Table Being Queried	OI
Library of Table Being Queried	AL
Member of Table Being Queried	OI
Name of Table Being Queried	OI

The bottom pane shows optimizer messages:

Message ID	Message text
CP14339	Query options retrieved file QAQQINI in library ALLIANCE.
CP1434A	**** Starting optimizer debug message for query .
CP14339	Query options retrieved file QAQQINI in library ALLIANCE.
CP1432C	All access paths were considered for file ORDERDTL.
CP1432C	All access paths were considered for file ORDERHDR.
CP14326	File ORDERDTL processed in join position 1.
CP14326	File ORDERHDR processed in join position 2.



# 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)**



*Skill and Leadership*

# Query Optimization Tools

**Query optimization tools: Comparison table**

<b>PRTSQLINF</b>	<b>STRDBG or CHGQRYA</b>	<b>File-based monitor</b>	<b>Memory -Based Monitor</b>	<b>Visual Explain</b>
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.

The screenshot shows the Database Navigator interface for a database named 'MAP\_DSITXREF\_ONLY\_DSFILE501'. The 'Search for Objects' panel is set to search for 'Table' objects in the 'DSFILE501' schema. The 'Schema Tree' panel lists various objects, including tables like DSITXREF, DSINMST, and DSVNDPT, and views like DSINMST1 through DSINMST9. The main map area displays a complex network of relationships between these objects, with arrows indicating dependencies or relationships between tables and views.

Schema Tree			
Schema Table	Objects In Map		
...	Name		
...	Schema		
...	Type		
<input checked="" type="checkbox"/>	DSITXREF	CORPDATA	Table
<input checked="" type="checkbox"/>	DSINMST	DSFILE501	Table
<input checked="" type="checkbox"/>	DSINMST1	DSFILE501	View
<input checked="" type="checkbox"/>	DSINMST2	DSFILE501	View
<input checked="" type="checkbox"/>	DSINMST3	DSFILE501	View
<input checked="" type="checkbox"/>	DSINMST4	DSFILE501	View
<input checked="" type="checkbox"/>	DSINMST5	DSFILE501	View
<input checked="" type="checkbox"/>	DSINMST6	DSFILE501	View
<input checked="" type="checkbox"/>	DSINMST7	DSFILE501	View
<input checked="" type="checkbox"/>	DSINMST8	DSFILE501	View
<input checked="" type="checkbox"/>	DSINMST9	DSFILE501	View
<input checked="" type="checkbox"/>	DSINMSTA	DSFILE501	View
<input checked="" type="checkbox"/>	DSVNDPT	DSFILE501	Table
<input checked="" type="checkbox"/>	DSVNDPT1	DSFILE501	View
<input checked="" type="checkbox"/>	DSVNDPT2	DSFILE501	View
<input checked="" type="checkbox"/>	DSVNDPT3	DSFILE501	View
<input checked="" type="checkbox"/>	DSVNDPT4	DSFILE501	View
<input checked="" type="checkbox"/>	NEWVNDP1	DSFILE501	View
<input checked="" type="checkbox"/>	NEWVNDP2	DSFILE501	View
<input checked="" type="checkbox"/>	IBMDSIT_IN...	IBMPFRDTA1	Index



*Skill and Leadership*

# 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)



*Skill and Leadership*

# 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.



*Skill and Leadership*

# 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



*Skill and Leadership*

# 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