Preparing for the V5R2 SQL Query Engine

Robert C Driesch

BP10

Trademarks & Disclaimer

- Course content and examples are based upon Version 5 Release 2 of OS/400 and DB2 UDB for iSeries. System and application behavior may be different on other releases of OS/400 and DB2 UDB.
- Example queries may not compile or run.
- NOTICE: this publication may refer to products that are not currently available in your country.
- IBM makes no commitment to make available any products referred to herein.
- IBM, eServer, iSeries, AS/400, OS/400 and DB2 are trademarks of the IBM Corporation in the United States or other countries or both.
- Other company, product, and service names may be trademarks or service marks of others.
- Java and all java-based trademarks and logos are trademarks of Sun Microsystems, Inc. In the United States and/or other countries.
- IBM's VisualAge products and services are not associated with or sponsored by Visual Edge software, ltd.
- A list of trademarks may be found on the worldwide web: http://www.ibm.com/trademarks.html

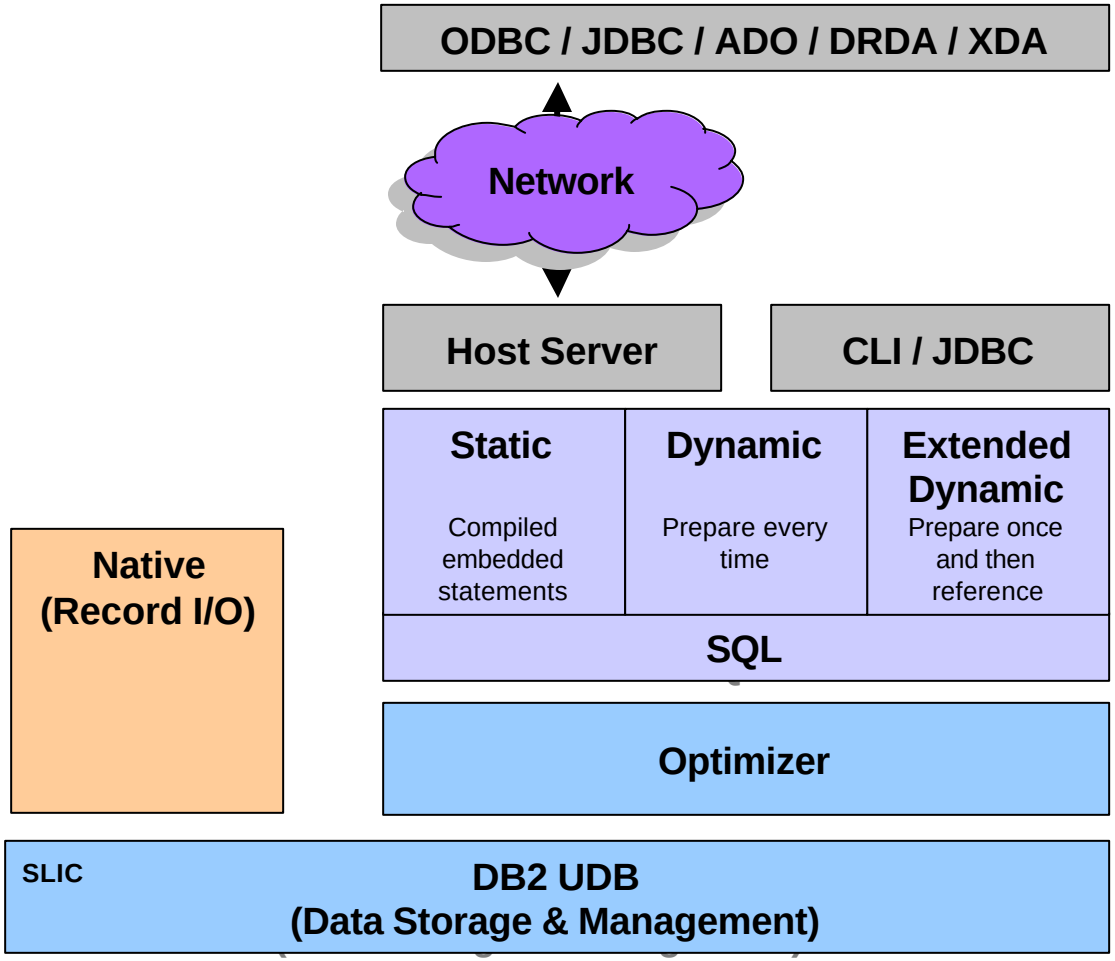
Problems Facing the Database

- Most new applications written or ported to the iSeries are SQL based
- Existing database engine getting too difficult to enhance and maintain with the many required features of SQL
- Need a database engine to be able to easily handle the complexities of SQL set-based queries
- Need to reduce the cost of adding new features to the database in order to keep up with an ever changing industry

Solution

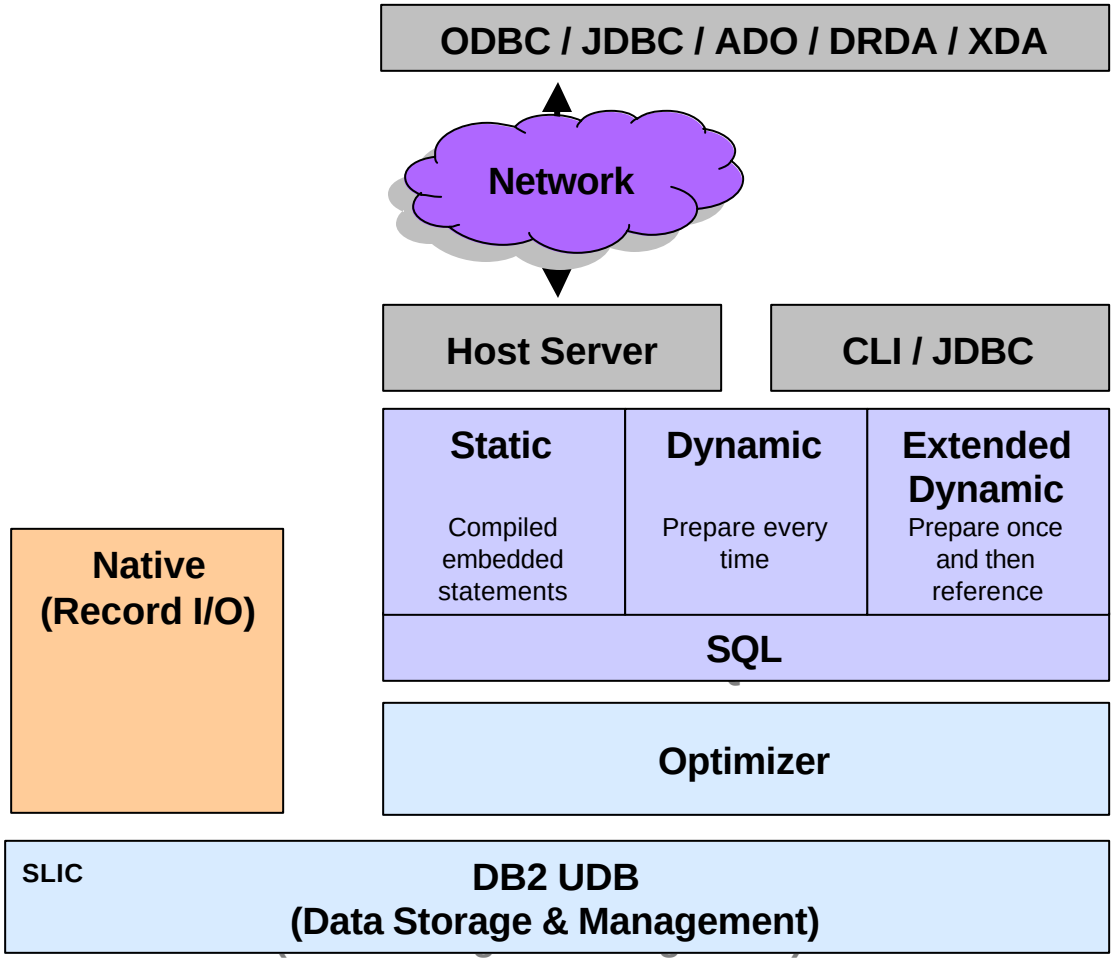
- Create a database engine with the flexibility to handle all of the SQL based queries we expect to see now and in the future
- Integrate all aspects of query processing to leverage overlapping technologies
  - Where able, leverage knowledge and skills from other DB2 UDB platforms
- Maintain flexibility in all aspects of the design
  - Allow different approaches to be used to solve similar queries
    - Example: Nonsense queries now optimize more efficiently by default
- Primarily going to improve longer running complex SQL Queries
  - ***Not*** designed to improve OLTP queries

V5R1 Database Architecture



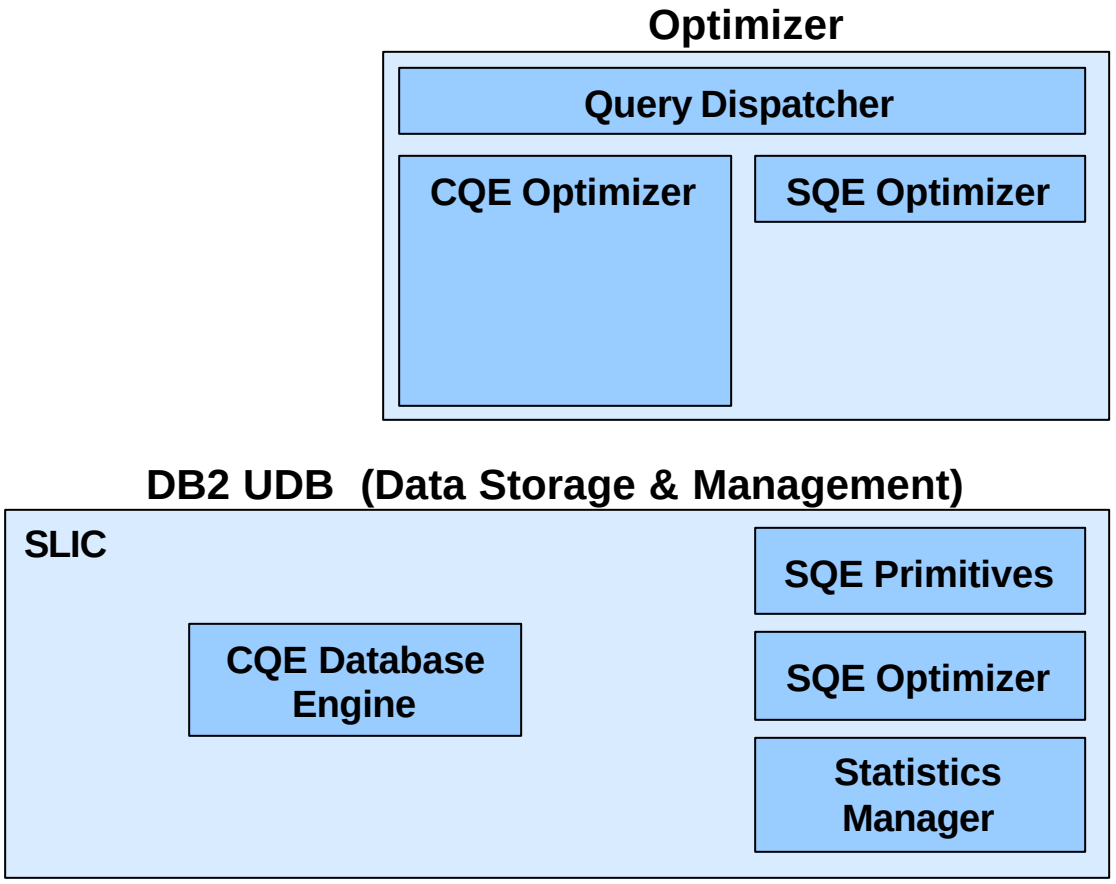
The optimizer and database engine are separated at different layers of the operating system

V5R2 Database Architecture



The optimizer and database engine merged to form the SQL Query Engine and much of the work was moved to the System Licensed Internal Code (SLIC)

V5R2 Database Architecture



The optimizer and database engine merged to form the SQL Query Engine and much of the work was moved to the System Licensed Internal Code (SLIC)

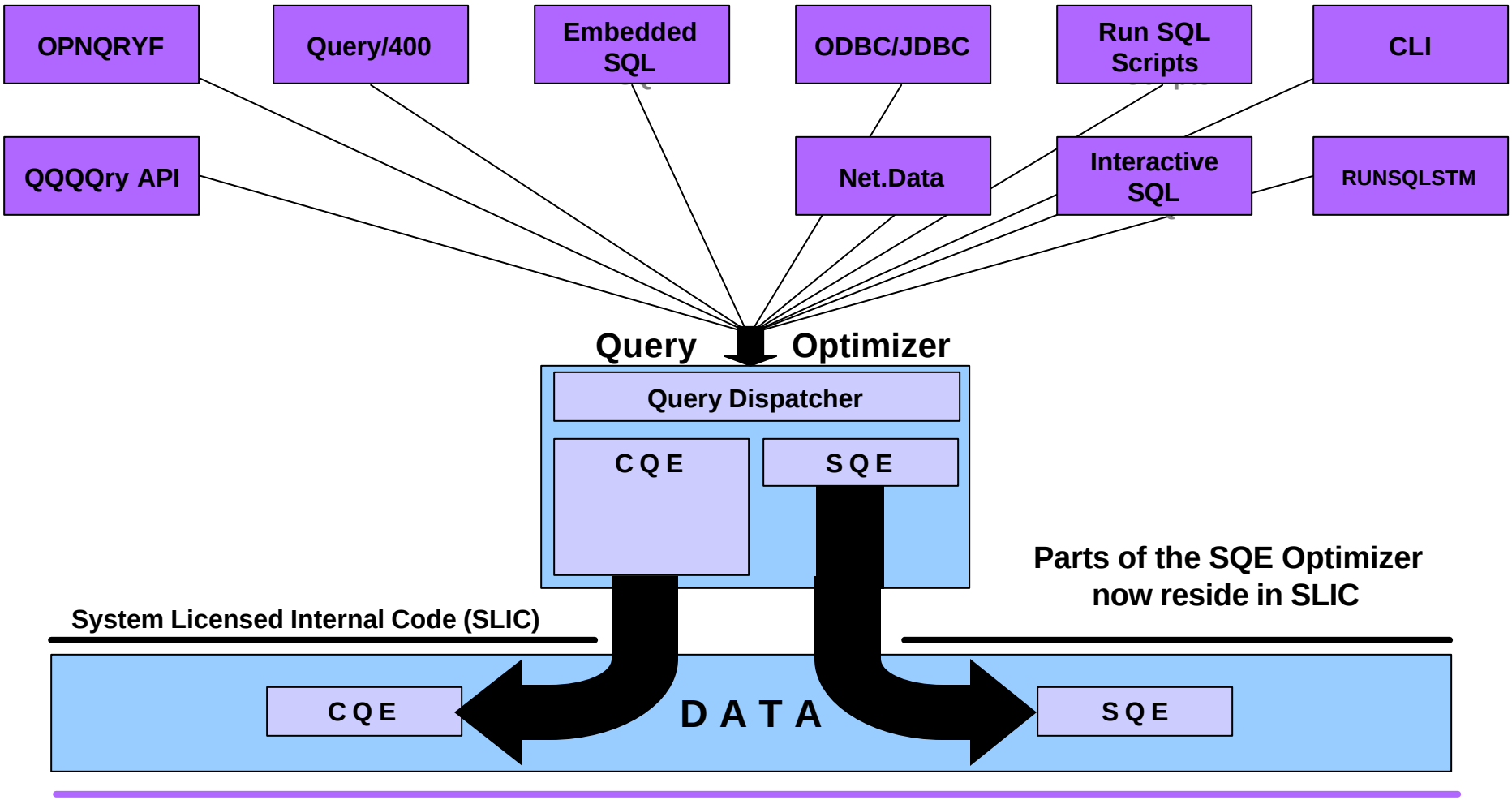
What's in a Name

- DB2 UDB for the iSeries represents the entire integrated database for the iSeries
- Still only one optimizer shipped with the system
- Optimizer now determines what engine will handle the query request (i.e. the Query Dispatcher)
  - Classic Query Engine (CQE)
  - SQL Query Engine (SQE)

The Query Dispatcher

- Determines which engine will optimize and process each query request
  - **Only SQL queries are considered for the SQL Query Engine**
- Initial step for all query optimization that occurs on the system
- Initially few queries will take advantage of SQE in V5R2:
  - In general, single table, read-only SQL queries will be considered
  - More enhancements (i.e. more SQL queries) will be added in the future
  - Ability to “back up” and use the Classic Query Engine when non-standard indexes are encountered during optimization

Optimizer Interfaces



The Query Dispatcher

- **Dispatched to CQE initially if:**
  - ***>1 Table (i.e. no joins)*** ***New features PTF'd into V5R2***
  - ***OR & IN predicates***
  - ***SMP requested***
  - Non-Read (INSERT with subselect can use new path)
  - LIKE predicates
  - UNIONS
  - View or Logical File references
  - Subquery
  - Derived Tables & Common Table expressions
  - LOB columns
  - NLSS/CCSID translation between columns
  - DB2 Multisystem
  - Non-SQL queries (QQQQry API, Query/400, OPNQRYF)

The Query Dispatcher

- **Back up to CQE to complete optimization if any of the following are encountered:**
  - Select/omit logical file
  - Logical file over multiple members
  - Derived key (s)
    - Native logical files that perform some intermediate mapping of the fields referenced in the key. Common ones are renaming fields, adding a translate or only selecting a subset of the columns.
    - Specifying an Alternate Collating Sequence (ACS) on a field used for a key will also make a “derived key” (an implied map occurs within the index).
  - Sort Sequence (NLSS) specified for index or logical file
    - Probably the trickiest one to detect for users. The index is built while an NLSS table is specified in the query environment.
    - Can effect SQL create indexes.
  - Cost to “back up” and revert to CQE adds about 15% to the total optimization time

SQE Optimizer

- Controls the strategies and algorithms used to determine what data access methods should be employed
- No knowledge of the meta-data or the systems capabilities
  - Asks questions about the system and the tables and uses the answers in its algorithms
  - Separation of responsibilities:
    - The Statistics Manager is asked for the number of expected I/O's so an I/O Cost can be calculated
    - The SQE Primitives are asked to model the data access method so a CPU Cost can be calculated
- Builds access plans of the data access methods organized into a tree-based structure
- The strategies allow the optimizer to consider plan possibilities never before considered on the iSeries
  - Even strategies similar to CQE have been approached differently to make them more robust and powerful
- Still responsible for collecting and processing the feedback information

SQE Primitives

- Controls the actual implementation of the query using data access methods derived from an OO tree-based architecture
- More aggressive on utilizing I/O subsystems and main memory
  - Different memory footprint because the mechanisms and structures used by SQE have changed from CQE
- Redesigned and implemented many of the same existing data access methods
  - On average less CPU consumption
  - Temporary indexes will no longer be created
- Some new data access methods have been added
- Temporary results have been retooled to eliminate the need for a “true” database object to be created
  - Leverages the proximity of the code in relation to the data to minimize data movement and take advantage of efficient structures to satisfy the request

Statistics Manager

- Controls access to all meta-data used for query optimization
  - Does not actually run or optimize a query
- Answers questions posed by the SQE Optimizer
  - Accuracy of the answers will dictate the optimizers ability to chose the best plan
  - Must *always* provide an answer to a question
- Answers are derived from different stats sources

Why Are Stats Important?

"The cornerstone of SQL optimization is accurately determining the number of rows which will qualify for an SQL statement. If the correct statistics are collected, the optimizer will more accurately estimate the number of rows each table will join. Collecting these statistics should result in better join order, join method, more appropriate sort decisions, as well as improved table access."

Patrick Bossman, IDUG August 1999

Types of Stats Questions

Type of Question	Description
Selectivity	How many records will be selected by a given selection predicate or combination of predicates?
Cardinality	How many distinct occurrences of a value exist for a single column or multiple columns in a query?
Meta-data	How many records exist within a table?
Meta-data	What indexes exist over a given table and what keys are interesting?
I/O Estimation	How many I/O's will be required to processed this table or index?

Sources for Answers

- Stats
  - Column Cardinality, Histograms & Frequent Values List
  - Constructed over a single column in a table
    - Stored internally as a part of the table object after created
  - Collected automatically by default for the system
  - Stats not immediately maintained as the table changes. Instead stats are refreshed as they become “stale” over time
- Existing indexes (Radix or Encoded Vector)
  - More accurately describes multi-key values
  - Adds overhead to database for inserts and updates
- Default information
  - No representation of actual values in columns

Types of Stats

Type of Question	Description
Column Cardinality	The number of distinct values in a column.
Histograms	Distribution statistic that describes the selectivity and the distribution of values for a given column.
Frequent Values List	A table of values that most frequently occur within a column and a count of their frequency.

NOTE: Stats will add 8 to 12K per column to the total size of a database object.

Reactive Statistics Collection

- Allows Statistics Manager to self-manage the collection of stats needed to answer the questions more accurately
- By default, the Statistics Manager will automatically perform statistics collections in a reactive mode
- The system will reach a steady state where frequently used tables and columns will have their stats collected
- Feedback from the Database Monitor can be used to identify stats for reactive collection

Proactive Statistics Collection

- Implement a good indexing strategy
  - http://www.iseries.ibm.com/developer/bi/documents/strategy/strategy.html
- Can use iSeries Navigator interface to create stats
- Within a “Well Tuned” query environment
  - Existing need for indexes for implementation choices
  - Less need for stats
- Within an Ad Hoc query environment
  - More stats for frequently referenced columns
  - Indexes over multi-key combinations with large data skew
  - More equal distribution of indexes and stats

Statistics Collection Interface

- System value to control automatic stats collection
  - QDBFSTCCOL Database file statistics collection
    - *ALL Allow both user & system requested stats collections
    - *NONE No stats collection is allowed
    - *USER Only user requested stats collection allowed
    - *SYSTEM Only system requested stats collection allowed
- QDBFSTCCOL – Stats collection job
  - Automatic collection runs in the background at very low priority
  - Statistics Manager analyzes entries in the Plan Cache and queues up requests for the collection job
- iSeries Navigator graphical interface to request stats collection
  - Immediate stats generation on a specified table and column (s)

Refresh Statistics Collection

- All collected stats (system or user) are managed by the Statistics Manager
- Stats are classified as “stale” when the base table has changed by 15% or more
- Stale stats are queued to be reprocessed through the Stats Collection job
  - System value must allow automatic collection
- Refreshed stats may not always cause a plan to be reoptimized

Plan Cache

- Incorporates Self-Learning Technology
  - Cache is self-managed to keep most active queries available for reuse
  - Plans are optimized on-demand as new stats or indexes become available
  - Will allow for future interrogation of the plans to make wiser costing decisions
- Caches *all* plans optimized by the SQE Optimizer
  - Allows more reuse of existing plans for dynamic and ad-hoc queries across jobs
  - Works in conjunction with the System Wide Statement Cache and the SQL programs, packages and service programs
- Cache is cleared during an IPL

Plan Cache

- Better centralized management of plan information:
  - Plans are stored in a compressed mode
  - Plans stored independent of job information for better sharing of plans
  - Access is optimized to minimize contention on plan entries across the system
  - Out of date plans are cycled from the cache as more space is needed
- Repository of information that is used to determine feedback and automatic stats generation
- Enabling auto stats collection causes the Statistics Manager to interrogate the Plan Cache looking for plans where stats would have been helpful

Feedback Differences

- No new messages or records generated for SQE
  - Column QQC16 in the 3014 record will contain an indicator identifying a query implemented through SQE
- Data access methods mapped to existing messages & icons:
  - Debug messages in joblog
  - Database Monitor data
  - Visual Explain diagrams
  - PRTSQLINF messages
- Layout and pattern of the feedback may be different then what was experienced before

How Do I Get It?

- SQE is enabled on all V5R2 systems
- Code will be refreshed via PTF's to support new features through the Query Dispatcher
- Refer to the following web site to check for availability of the PTF's necessary to modify the Query Dispatcher
  - http://www.iseries.ibm.com/db2/sqe.html

Migration Tips

- ***Common sense rules the day!***
- Collection of feedback information prior to any changes can dramatically help problem determination later
- Any change to the environment in which queries run can effect the plans chosen
  - Optimizer strategy or algorithm changes
  - Hardware or system changes
  - Changes to the underlying tables, indexes or stats
- Implementing a good indexing strategy will minimize any effects these may have
  - http://www.iseries.ibm.com/developer/bi/documents/strategy/strategy.html

Additional Information

- **DB2 UDB for iSeries home page** – http://www.iseries.ibm.com/db2
- **Newsgroups**
  - USENET: comp.sys.ibm.as400.misc, comp.database.ibm-db2
  - iSeries Network (NEWS/400 Magazine) SQL & DB2 Forum – http://www.iseriesnetwork.com/Forums/main.cfm?CFApp=59
- **Education Resources – Classroom & Online**
  - http://www.iseries.ibm.com/db2/db2educ_m.htm
  - http://www.iseries.ibm.com/developer/education/ibo/index.html
- **DB2 UDB for iSeries Publications**
  - Online Manuals: http://www.iseries.ibm.com/db2/books.htm
  - Porting Help: http://www.iseries.ibm.com/developer/db2/porting.html
  - DB2 UDB for iSeries Redbooks (http://ibm.com/redbooks)
    - ***Stored Procedures & Triggers on DB2 UDB for iSeries*** (SG24-6503)
    - ***DB2 UDB for AS/400 Object Relational Suppor***t (SG24-5409)
    - ***SQL Query Engine***
      - (http://publib-b.boulder.ibm.com/Redbooks.nsf/RedpieceAbstracts/sg2456598.html)
  - ***SQL/400 Developer's Guide*** by Paul Conte & Mike Cravitz
    - http://iseriesnetwork.com/str/books/Uniquebook2.cfm?NextBook=183
  - ***iSeries and AS/400 SQL at Work*** by Howard Arner
    - http://www.sqlthing.com/books.htm
- **Please send questions or comments to** rchudb@us.ibm.com

Conclusion

- Initially limited queries will take advantage of SQE in V5R2
  - Minimizes risk to existing applications
  - Pervasive change
    - All SQL queries have potential to be effected
  - More enhancements (i.e. more SQL queries) in the future
- In general, some queries perform better, some perform the same
- More improvements expected in the future as the technology is leveraged