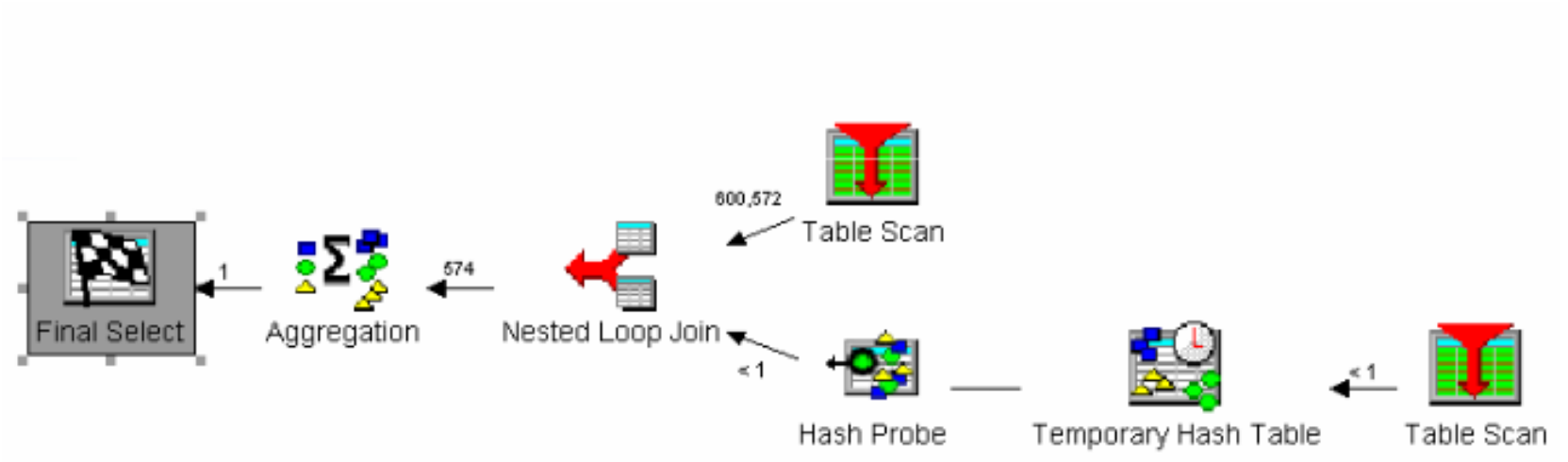
Optimize SQL Performance on DB2 for i

Strategies to make your query and application as fast as possible.



Today's Speaker

Andrew Clark

CTO

RT Analytics

aclark@rtanalytics.com

Today's Agenda

How Optimization Works

What is an Index

How is the SQL Query Engine (SQE) Faster?

Improving Performance

Summary

Questions & Answers

References

Preparing for and Tuning the SQL Query Engine on DB2 for i5/OS *Redbook 2006*
http://www.redbooks.ibm.com/redbooks/pdfs/sg246598.pdf

Optimizing query performance using query optimization tools *Knowledge Center*
https://www-01.ibm.com/support/knowledgecenter/ssw_ibm_i_72/rzajq/queryopt.htm

DB2 for i SQL OLAP Functions *Mike Cain*
http://www.scandevconf.se/db/SQL%20Programming%20DB2%20for%20i%20SQL%20OLAP%20Functions%20%20Mike%20Cain.pdf

How Optimization Works

Query Optimizer uses “cost” to calculate fastest method to implement plan

Optimizer Logging

`STRDBG`

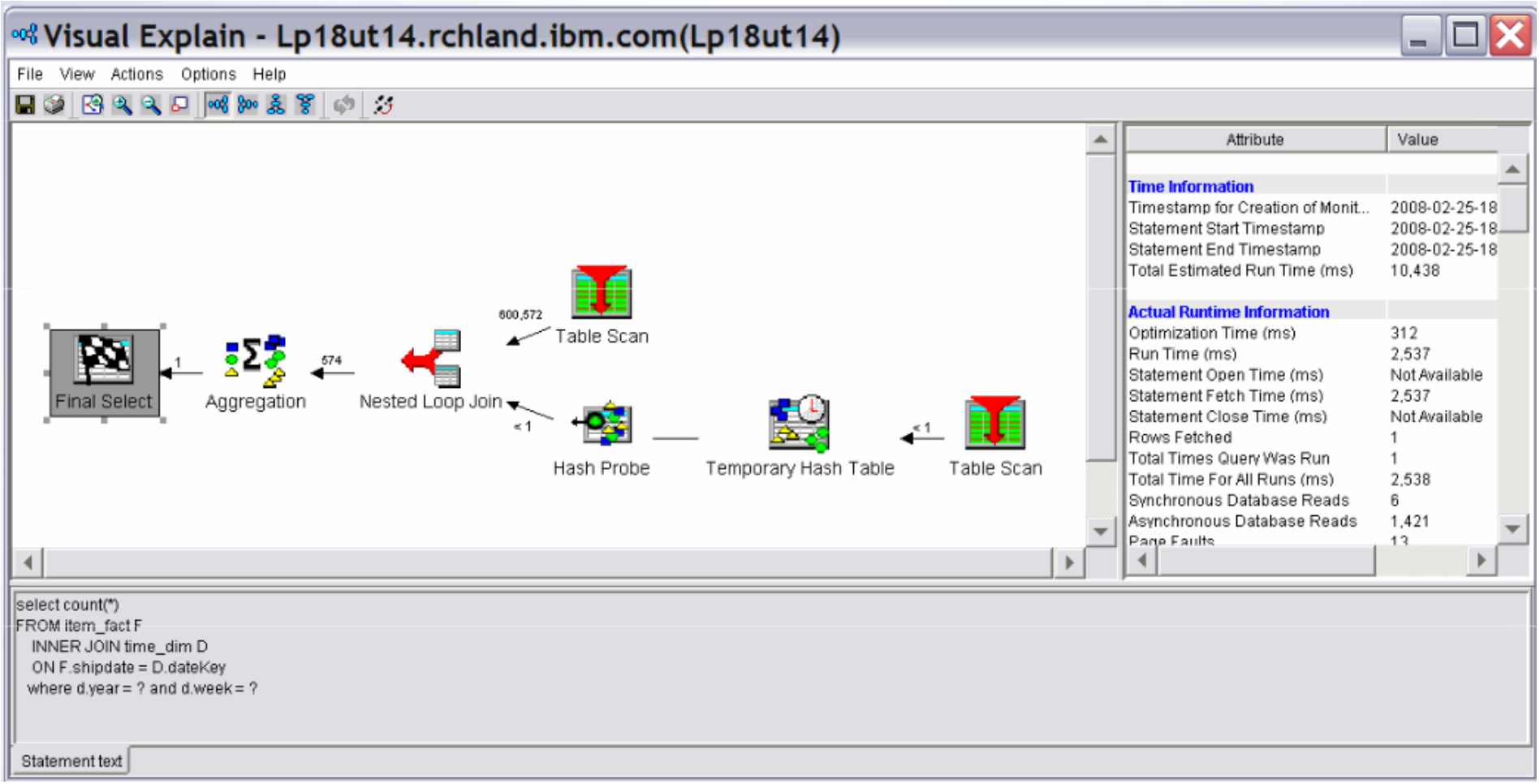
`PRTSQLINF` (*PGM/*SRVPGM only)

QAQQINI: MESSAGES_DEBUG=*YES

Visual Explain

SQL Performance Monitor

Visual Explain



Scan vs Probe vs Create

Strategy	How	When	Example	Debug Message
Scan	Reads all records, Buffered I/O	>~70% match	SELECT * FROM Employee WHERE WorkDept BETWEEN 'A01'AND 'E01'	CPI4329 — Arrival sequence was used for file EMPLOYEE
Probe	Randomly position to needed records	<~30% match	CREATE INDEX X1 ON Employee (LastName)	CPI4328 — Access path of file X1 was used by query
Create		No suitable object exists		CPI4325 -- Temporary result file built for query

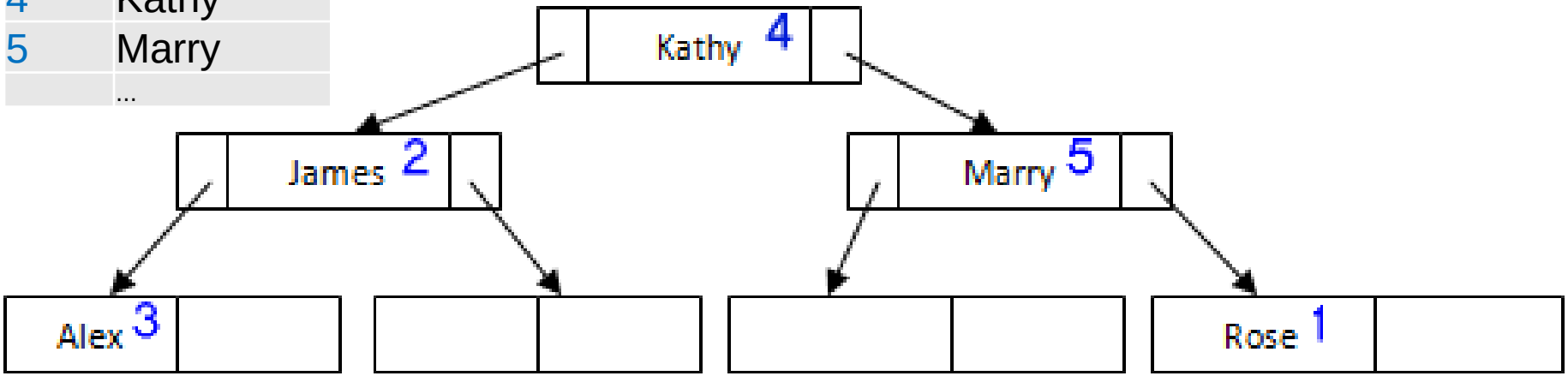
What is an Index?

An index is really just a binary tree where each “node” is the key value and a pointer to the record it is in.

$\log_2(N)$

100,000 records ~ 17 reads
1,000,000 (million) records ~ 20 reads
1,000,000,000 (billion) records ~ 30 reads
1,000,000,000,000 (trillion) ~ 40 reads

RRN	First Name
1	Rose
2	James
3	Alex
4	Kathy
5	Marry
	…



CQE vs SQE

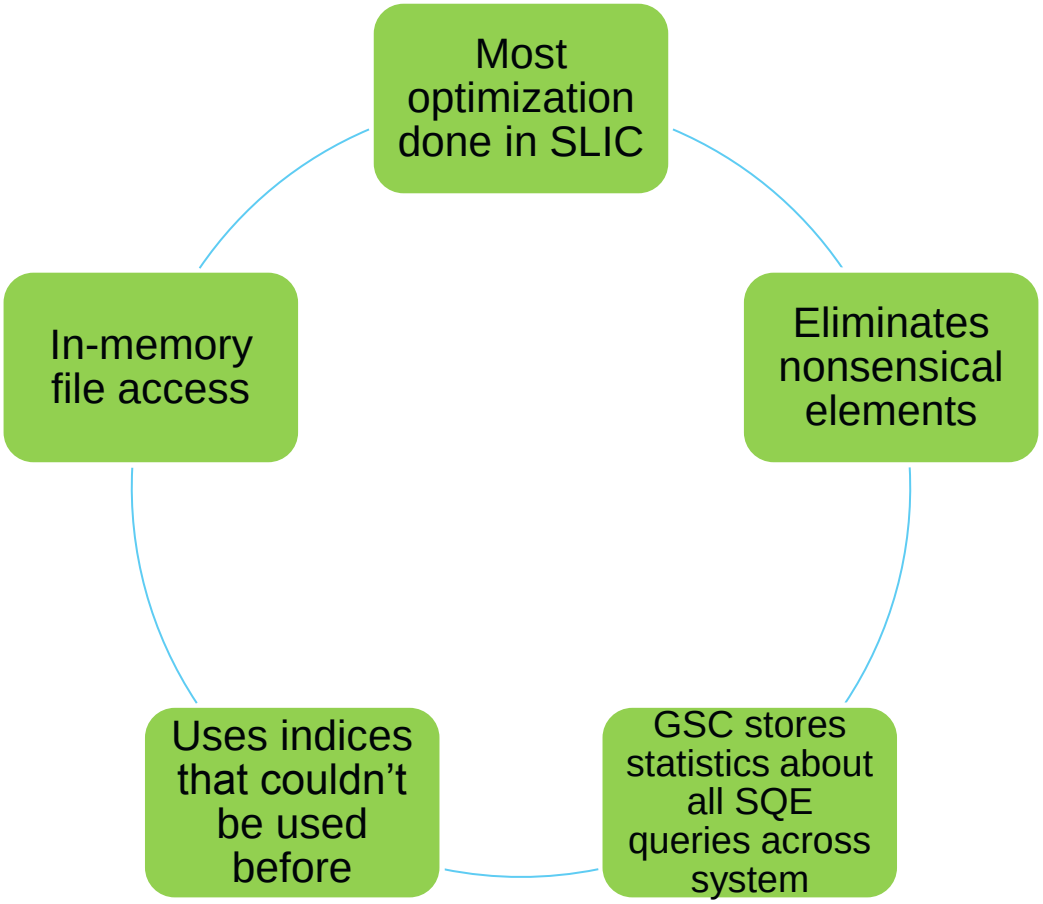
Classic Query Engine (CQE)

- Input is Query Definition Template (QDT)
- “Knows” about IBM i specific objects
  - Edit codes, edit words, multiple members/formats
- SQL converted to QDT

SQL Query Engine (SQE)

- Input is SQL
- Shared code base with DB2, not IBM i specific

How SQE is Faster



Encoded Vector Index (EVI)

Contains statistics on actual data stored:

Cardinality
Selectivity
Frequency

- 3 unique CSTTE values
- 1,000,000 records for CSTTE='IL' (Cardinality)
- 500,000 records for CSTTE='WI'
- 20,000 records stored for CSTTE='MN'
- The most common record is CSTTE='IL' (Frequency)
- CSTTE IN ('IL', 'WI') makes up 99% of records in index (Selectivity)

Materialized Query Tables (MQT)

Materialized Query Tables

Create work or summary files that are easily maintained, refreshed, and used for optimization.

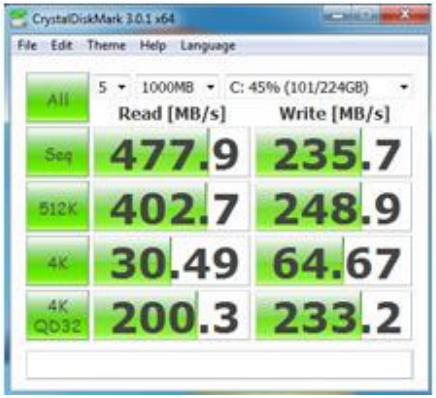
```
CREATE TABLE MQT2 AS
        (SELECT D.deptname, D.location,
        sum(E.salary) as sum_sal
        FROM DEPARTMENT D, EMPLOYEE E
        WHERE D.deptno=E.workdept
        GROUP BY D.Deptname, D.location)
    DATA INITIALLY IMMEDIATE
    REFRESH DEFERRED (someday, maybe REFRESH IMMEDIATE)
    ENABLE QUERY OPTIMIZATION
    MAINTAINED BY USER

REFRESH TABLE MQT2
```

HDD vs SSD vs RAM

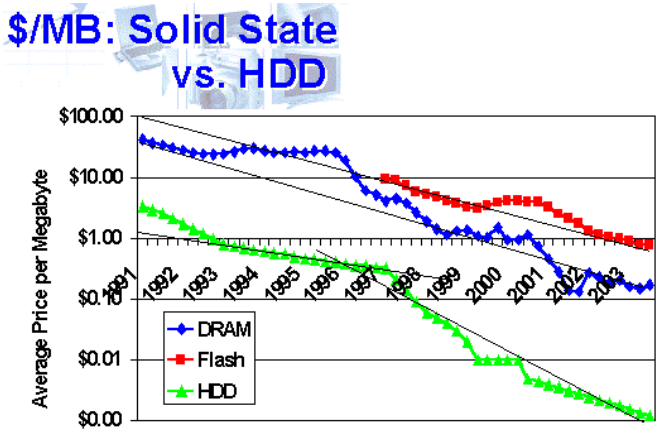
RAM ~10x > SSD ~10x > HDD

Hard Drive

SSD



RAM Disk



Keeping Objects in Memory

SETOBJACC, KEEPINMEM (V7R1)
`SETOBJACC OBJ(IMPORTANTF) OBJTYPE(*FILE) POOL(*BASE)`
`CHGPF IMPORTANTF KEEPINMEM(*YES)`
`CHGLF IMPORTANTLF KEEPINMEM(*YES)`

`CREATE/ALTER TABLE SCHEMA.IMPORTANTF`
`  UNIT SSD`
`  KEEP IN MEMORY YES`

SQE Limitations

Can’t Select From Multi-Member/Format Files

For Multi-Member Access:

Create an alias (using SQL) that selects from that file [member or format]:

```
CREATE ALIAS LIBRARY.FILE_MEMBER FOR LIBRARY.FILE(MEMBER)
```

...and then simply select from that alias:

```
SELECT * FROM LIBRARY.FILE_MEMBER
```

For Multi-Format Logicals:

Select data from the physical file that the format is associated with—not the single format of a multi-format logical file

Instead of :

```
SELECT * FROM MULT_FORMAT_LF(FORMAT)
```

...simply specify :

```
SELECT * FROM PF_THAT_FORMAT_IS_BUILT_OVER
```

Avoid If Possible

Different order of GROUP BY and ORDER BY clauses

```
Select CUSNO, CNAME, SUM(ORVAL)
FROM CUSTMAST, ORDHEAD
WHERE CUSTMAST.CUSNO=ORDHEAD.CUSNO
GROUP BY CUSNO, CNAME
ORDER BY CNAME, CUSNO
```

Excluding ORDER BY from SELECT

```
Select CNAME
FROM CUSTMAST
ORDER BY CUSNO
```

UDFs that are not deterministic

```
create function foo() returns float not deterministic
```

VARCHAR fields with ALLOCATE(0)

Avoid If Possible

Eliminate long fields from GROUP BY

```
Select CUSNO, CNAME, SUM(ORVAL)
FROM CUSTMAST, ORDHEAD
WHERE CUSTMAST.CUSNO=ORDHEAD.CUSNO
GROUP BY CUSNO, CNAME
```

Replace with CTE and join after grouping result complete

```
WITH GGG AS (
Select CUSNO, SUM(ORVAL)
FROM CUSTMAST, ORDHEAD
WHERE CUSTMAST.CUSNO=ORDHEAD.CUSNO
GROUP BY CUSNO)
Select CCC.CUSNO, CNAME
FROM GGG
INNER JOIN CUSTMAST ON GGG.CUSNO=CCC.CUSNO)
```

Date Indices

- Create UDF to translate “your” date types:

```
CREATE FUNCTION RTA/YYMDTODATE(YYMD DECIMAL(8,0))
 RETURNS DATE
 LANGUAGE SQL
 DETERMINISTIC
 RETURNS NULL ON NULL INPUT
 NO EXTERNAL ACTION
 NOT FENCED
BEGIN
  RETURN DATE('01/01/0001') +
   (INT(YYMD/10000)-1) YEARS +
   (INT(MOD(YYMD,10000)/100)-1) MONTHS +
   (MOD(YYMD,100)-1) DAYS;
END
```

- Create Index using UDF:

```
CREATE INDEX GL/HIRE_DATE_INDEX ON GL/EMPLOYEE(RTA/YYMDTODATE(HIREYYMD))
```

- Use UDF when writing SQL statement
  (or build SQL View that does the same thing):

```
SELECT *
FROM GL/EMPLOYEE
WHERE RTA/YYMDTODATE(HIREYYMD) BETWEEN '01/01/2001' AND '12/31/2015'
```

Improve Your Performance

- Upgrade to latest IBM i os and TR level
  (~80 performance enhancements since 2010)
  https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/IBM%20i%20Technology%20Updates/page/DB2%20for%20i%20Performance%20Enhancements
- Analyze Logging Messages
- Optimize SQL statements
- Create Appropriate Indices
  - Weigh cost of disk usage/index updating vs implementation time
  - Create Encoded Vector Index when
    - Data is scattered/not uniquely identified
    - Data is fairly static (no transaction files)
    - Queries tend to be “Ad Hoc”
  - Create Materialized Query Table when
    - Huge amount of data is grouped
    - Data is very static (week-end, month-end transactions)
  - Otherwise, create normal (radix) index
- Identify critical files and move to SSD or force into memory

Index Advisor

From Operations Navigator: Databases> Index Advisor

Condensed Index Advice - Serverxyz

File Edit View Help

2 minutes old

Database: Serverxyz Condensed Index Advice for SAMPLE Filter: Include settings applied

Table for Which Index was Advised	Schema	System Schema	System Name	Partition	Keys Advised	Advised Index Type	Last Advised for Query Use	Times Advised for Query Use	Estimated Index Creation Time	Logical Page Size Advised (KB)	Most Expensive Query Estimate	Average of Query Estimates	Rows in Table when Advised	NLSS Table Advised	MTI Used	MTI Creat...	MTI Last Used
MQT1	SAMPLE	SAMPLE	MQT1		JOB	Binary Radix	9/1/09 3:57:14 PM	10	00:00:01	64	1	1	42	*HEX	4	3	9/1/09 2:58:10 PM
EMPLOYEE	SAMPLE	SAMPLE	EMPLOYEE		JOB, WORKDEPT	Binary Radix	9/1/09 4:06:54 PM	8	00:00:01	64	1	1	42	*HEX	0	0	
DEPARTMENT	SAMPLE	SAMPLE	DEPARTMENT		LOCATION, DEPTNAME	Binary Radix	9/1/09 3:53:05 PM	8	00:00:01	64	1	1	14	*HEX	0	0	
DEPARTMENT	SAMPLE	SAMPLE	DEPARTMENT		LOCATION, DEPTNO	Binary Radix	9/1/09 3:53:05 PM	8	00:00:01	64	1	1	14	*HEX	0	0	
DEPARTMENT	SAMPLE	SAMPLE	DEPARTMENT		DEPTNAME, LOCATION	Binary Radix	9/1/09 3:53:12 PM	6	00:00:01	64	1	1	14	*HEX	0	0	
MQT2	SAMPLE	SAMPLE	MQT2		LOCATION, DEPTNAME	Binary Radix	9/1/09 3:53:15 PM	4	00:00:01	64	1	1	8	*HEX	0	0	
MQT2	SAMPLE	SAMPLE	MQT2		DEPTNAME, SUM_SAL	Encoded vector (not unique)	9/1/09 3:53:15 PM	4	00:00:01	64	1	1	8	*HEX	0	0	

1 - 7 of 7 objects

OLAP Features

V5R4

- ROW_NUMBER
- DENSE_RANK
- RANK

```
SELECT empno, lastname, salary+bonus AS
TOTAL_SALARY,
RANK() OVER (ORDER BY salary+bonus DESC) AS
Salary_Rank
FROM employee
WHERE salary + bonus > 30000
ORDER BY lastname
```

SELECT empno, lastname, salary+bonus AS TOT...

EMPNO	LASTNAME	TOTAL_SALARY	SALARY_RANK
000050	GEYER	40975.00	5
000010	HAAS	53750.00	1
200010	HEMMINGER	47500.00	2
000090	HENDERSON	30350.00	11
200220	JOHN	30440.00	9
000030	KWAN	39050.00	6
000110	LUCCHESSI	47400.00	3
000220	LUTZ	30440.00	9
000070	PULASKI	36870.00	7
000060	STERN	32750.00	8
000020	THOMPSON	42050.00	4

Dense_Rank() Output

SALARY_RANK
5
1
2
10
9
6
3
9
7
8
4

OLAP Features

V6R1

- CUBE
- GROUPING SETS
- GROUPING
- ROLLUP

```
SELECT country, region, SUM(sales)
FROM trans
GROUP BY ROLLUP (country, region)
```

GROUP BY country,NULL

GROUP BY NULL, NULL

Country	Region	Sum(Sales)
Canada	-	100,000
Canada	NW	100,000
U.S.A.	-	3,250,000
U.S.A.	NE	450,000
U.S.A.	NW	940,000
U.S.A.	SE	550,000
U.S.A.	SW	1,310,000
-	-	3,350,000

V6R1 Features

Full Outer Join

- ```
  SELECT ccc.cusno, ordno
  FROM sequelex/custmast ccc
  FULL OUTER JOIN sequelex/ordhead ooo
  ON ccc.cusno=ooo.cusno
  ```

Fetch First in Subselect

- ```
  SELECT ordno, odrval FROM ordhead WHERE orval >
  (SELECT sum(orval) FROM ordhead
  ORDER BY odate
  FETCH FIRST 10 ROWS ONLY)
  ```

DECFLOAT Data Type

- (Theoretically, at least) massive performance improvement over packed decimal on Power 6+ hardware

V7R1 Features

XML Datatype (*PTF to V7R1)

- **SELECT** U."PO ID", U."Part #", U."Product Name", U."Quantity", U."Price", U."Order Date"
  FROM PURCHASEORDER P,

```
XMLTABLE(XMLNAMESPACES('http://podemo.org' AS "pod"), '$po/PurchaseOrder/itemlist/item' PASSING P.PORDER AS "po" COLUMNS "PO ID" INTEGER PATH '../../@POid', "Part #" CHAR(6) PATH 'product/@pid', ...
```

V7R2 Features

Row and Column Access Control

```
CREATE MASK SSN_MASK ON EMPLOYEE
FOR COLUMN SSN RETURN
CASE WHEN (VERIFY_GROUP_FOR_USER(SESSION_USER,'PAYROLL') = 1)
THEN SSN WHEN (VERIFY_GROUP_FOR_USER(SESSION_USER,'MGR') = 1)
THEN 'XXX-XX-' CONCAT SUBSTR(SSN,8,4) ELSE NULL END ENABLE;
```

Timestamp Precision

```
CREATE TABLE x
(C1 TIMESTAMP(12), -- Additional precision when 6 is not enough
 C2 TIMESTAMP(0)) -- Less precision (and storage) when 6 is too much
```

Summary

Understanding Database Performance on IBM I

SQL Optimization

Taking Advantage of New Features

Questions?

Thank you for attending!

Andrew Clark
CTO
RT Analytics
aclark@rtanalytics.com

References:

Preparing for and Tuning the SQL Query Engine on DB2 for i5/OS *Redbook 2006*
http://www.redbooks.ibm.com/redbooks/pdfs/sg246598.pdf

Optimizing query performance using query optimization tools *Knowledge Center*
https://www-01.ibm.com/support/knowledgecenter/ssw_ibm_i_72/rzajq/queryopt.htm

DB2 for i SQL OLAP Functions *Mike Cain*
http://www.scandevconf.se/db/SQL%20Programming%20DB2%20for%20i%20SQL%20OLAP%20Functions%20%20Mike%20Cain.pdf