CONFIO SOFTWARE

Ignite IT Performance™

# Tuna Helper
# Proven Process for SQL Tuning

**Dean Richards**
**Senior DBA, Confio Software**

# Tuna Helper – Proven Process for SQL Tuning


Give a man a fish and you feed him for a day.
Teach a man to fish and you feed him for a lifetime.
Chinese Proverb

# Who Am I?

CONFIO SOFTWARE
Ignite IT Performance™

- Senior DBA for Confio Software
  - DeanRichards@confio.com
- Current – 20+ Years in Oracle, SQL Server
- Former – 15+ Years in Oracle Consulting
- Specialize in Performance Tuning
- Review Performance of 100's of Databases for Customers and Prospects
- Common Thread – Paralyzed by Tuning

# Agenda

CONFIO SOFTWARE
Ignite IT Performance™

- Introduction
- Challenges
- Identify - Which SQL and Why
- Gather – Details about SQL
- Tune – Case Study
- Monitor – Make sure it stays tuned
- SQL Tuning Myths

# Introduction

CONFIO SOFTWARE
Ignite IT Performance™

- SQL Tuning is Hard
- This Presentation is an Introduction
  - 3-5 day detailed classes are typical
- Providing a Framework
  - Helps develop your own processes
  - There is no magic tool
  - Tools cannot reliably tune SQL statements
  - Tuning requires the involvement of you and other technical and functional members of team

# Challenges

- SQL Tuning is Hard – did I mention that?
- Requires Expertise in Many Areas
  - Technical – Plan, Data Access, SQL Design
  - Business – What is the Purpose of SQL?
- Tuning Takes Time
  - Large Number of SQL Statements
  - Each Statement is Different
- Low Priority in Some Companies
  - Vendor Applications
  - Focus on Hardware or System Issues
- Never Ending

# Identify – Which SQL

- Highest Wait Times (Ignite, AWR, etc)
- Tracing a Session / Process
- User / Batch Job Complaints
- Highest I/O (LIO, PIO)
- SQL Performing Full Table Scans
- Known Poorly Performing SQL

# Example SQL Statement

CONFIO SOFTWARE
Ignite IT Performance™

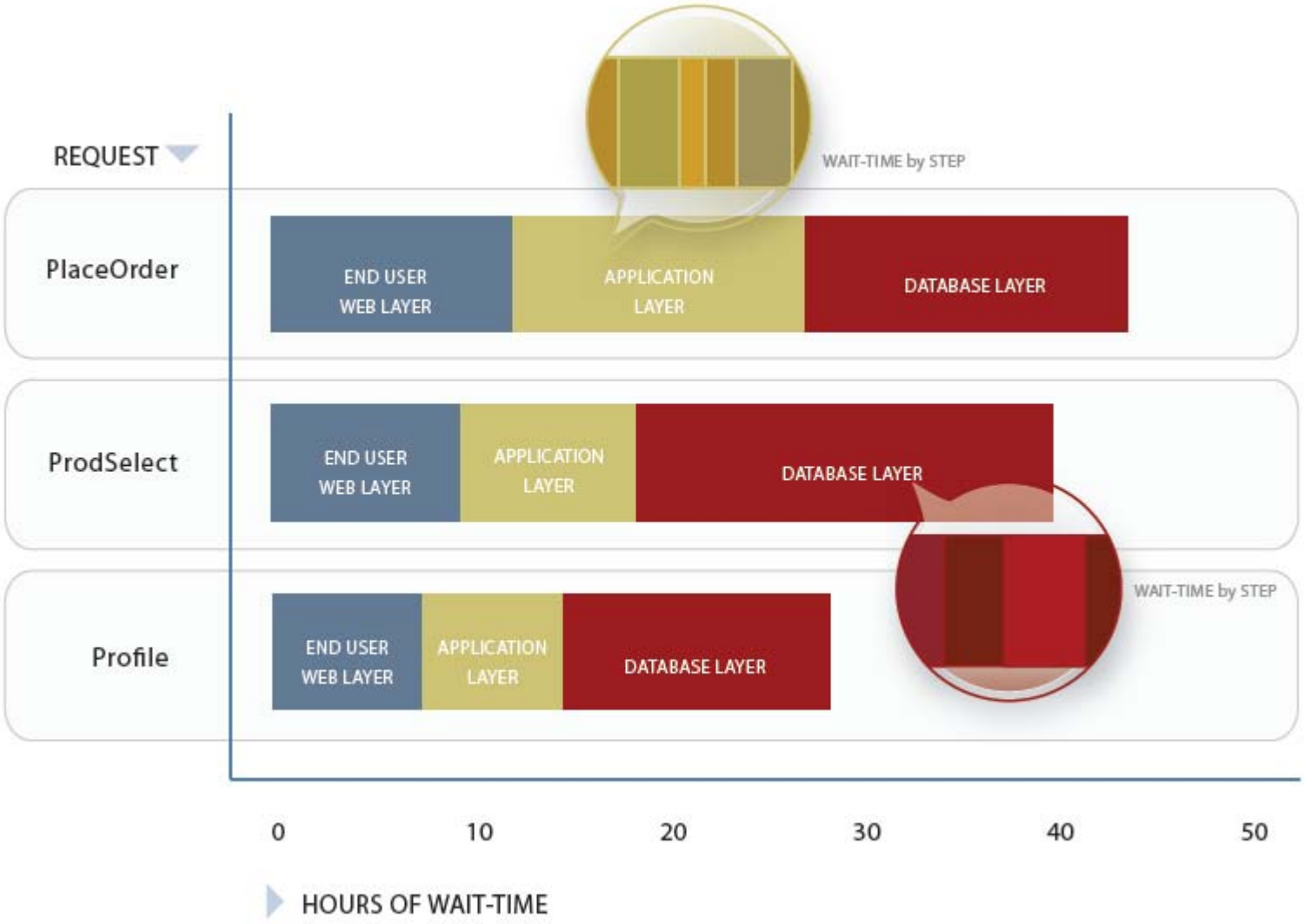
- Who registered yesterday for SQL Tuning

```
SELECT s.fname, s.lname, r.signup_date
FROM   student s, active_registrations r, class c
WHERE  s.student_id = r.student_id
AND    r.class_id     = c.class_id
AND    UPPER(c.name)  = 'SQL TUNING'
AND    c.class_level  = 101
AND    r.signup_date BETWEEN
          TRUNC(SYSDATE) AND TRUNC(SYSDATE-1)
```

# Identify – End-to-End

- Know the business reason for statement
  - Who registered yesterday for SQL Tuning
  - Why does the business need to know this
  - How often is the information needed
  - Who uses this information
- Understand the technical aspects
  - Review ERD
  - Understand application architecture
  - Understand tables and the data (at a high level)
- Understand the entire process
  - What portion of the total time is database
  - Where is it called from in the application

# Identify – End-to-End Time



# Wait Event Information

CONFIO SOFTWARE
Ignite IT Performance™

**V$SESSION**

SID
USERNAME
SQL_ID
PROGRAM
MODULE
ACTION
PLAN_HASH_VALUE

**V$SESSION_WAIT**

SID
EVENT
P1, P1RAW, P2, P2RAW, P3, P3RAW
STATE (WAITING, WAITED…)

• Oracle 10g added this info to V$SESSION

**V$SQL**

SQL_ID
SQL_FULLTEXT

**V$SQLAREA**

SQL_ID
EXECUTIONS
PARSE_CALLS
BUFFER_GETS
DISK_READS

**V$SQL_PLAN**

SQL_ID
PLAN_HASH_VALUE

**DBA_OBJECTS**

OBJECT_ID
OBJECT_NAME
OBJECT_TYPE

# Wait Time Scenario

- Which scenario is worse?
- SQL Statement 1
  - Executed 1000 times
  - Caused 10 minutes of wait time for end user
  - Waited 99% of time on “db file sequential read”
- SQL Statement 2
  - Executed 1 time
  - Caused 10 minutes of wait time for end user
  - Waited 99% on “enq: TX – row lock contention”

# Identify – Simplification

- Break Down SQL Into Simplest Forms
  - Complex SQL becomes multiple SQL
  - Sub-Queries Should be Tuned Separately
  - UNION'ed SQL Tuned Separately
  - Get the definition of views
  - Are synonyms being used

# Identify – High Level Analysis

CONFIO SOFTWARE
Ignite IT Performance™

- Look for obvious limiting factors
  - column = :1 - (>, <, BETWEEN)
  - column IN (‘A’, ‘B’) - EXISTS
  - column LIKE ‘ABCD%’
- Match up with existing indexes
- Avoid obvious non-limiting factors
  - <>, NOT LIKE, LIKE ‘%ABCD%’

# Identify – Summary

- Determine the SQL
- Understand End-to-End
- Understand Database Wait Time
- Simplify Statement
- High-Level Analysis

# Gather - Metrics

CONFIO SOFTWARE
Ignite IT Performance™

- Get baseline metrics
  - How long does it take now
  - What is acceptable (10 sec, 2 min, 1 hour)
- Collect Wait Time information
  - Locking / Blocking
  - I/O problem
  - Latch contention
  - Network slowdown
  - May be multiple issues
  - All have different resolutions
- Document everything in simple language

# Gather – Execution Plan

- EXPLAIN PLAN
  - Estimated execution plan - can be wrong for many reasons
- V$SQL_PLAN (Oracle 9i+)
  - Real execution plan
  - Use DBMS_XPLAN for display
- Tracing (all versions)
  - Get all sorts of good information
  - Works when you know a problem will occur
- Historical – AWR, Confio Ignite

# Plans Not Created Equal

CONFIO SOFTWARE
Ignite IT Performance™

```
SELECT company, attribute FROM data_out WHERE segment = :B1
```

- Wait Time – 100% on “db file scattered read”
- Plan from EXPLAIN PLAN

```
SELECT STATEMENT Optimizer=ALL_ROWS (Cost=1 Card=1 Bytes=117)
   TABLE ACCESS (BY INDEX ROWID) OF 'DATA_OUT' (TABLE) (Cost=1 Card=1 Bytes=117)
     INDEX (UNIQUE SCAN) OF 'IX1_DATA_OUT' (INDEX (UNIQUE)) (Cost=1 Card=1)
```

- Plan from V$SQL_PLAN using DBMS_XPLAN

```
select * from table(dbms_xplan.display_cursor('az7r9s3wpqg7n',0));

-------------------------------------------------------------------------------
| Id  | Operation          | Name     | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------
|   0 | SELECT STATEMENT   |          |       |       |   370 (100)|          |
|*  1 |  TABLE ACCESS FULL| DATA_OUT |     1 |   117 |   370   (4)| 00:00:05 |
-------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   1 - filter(TO_BINARY_DOUBLE("SEGMENT")=:B1)
```

# Gather – Bind Values

- V$SQL_BIND_CAPTURE
  - STATISTICS_LEVEL = TYPICAL or ALL
  - Collected at 15 minute intervals

```
SELECT name, position, datatype_string, value_string
FROM   v$sql_bind_capture
WHERE  sql_id = '15uughacxfh13';

NAME    POSITION DATATYPE_STRING VALUE_STRING
----- ---------- --------------- ------------
:B1             1 BINARY_DOUBLE
```

- Bind Values also provided by tracing
  - Level 4 – bind values
  - Level 8 – wait information
  - Level 12 – bind values and wait information

# Gather – Table / Index Stats

CONFIO SOFTWARE
Ignite IT Performance™

- Provides data on objects in execution plans.
  - Table sizes
  - Existing indexes
  - Cardinality of columns
  - Segment sizes
  - Histograms and Data Skew
  - Many things the CBO uses
- Use TuningStats.sql
  - http://support.confio.com/kb/1534
- Run it for expensive data access targets

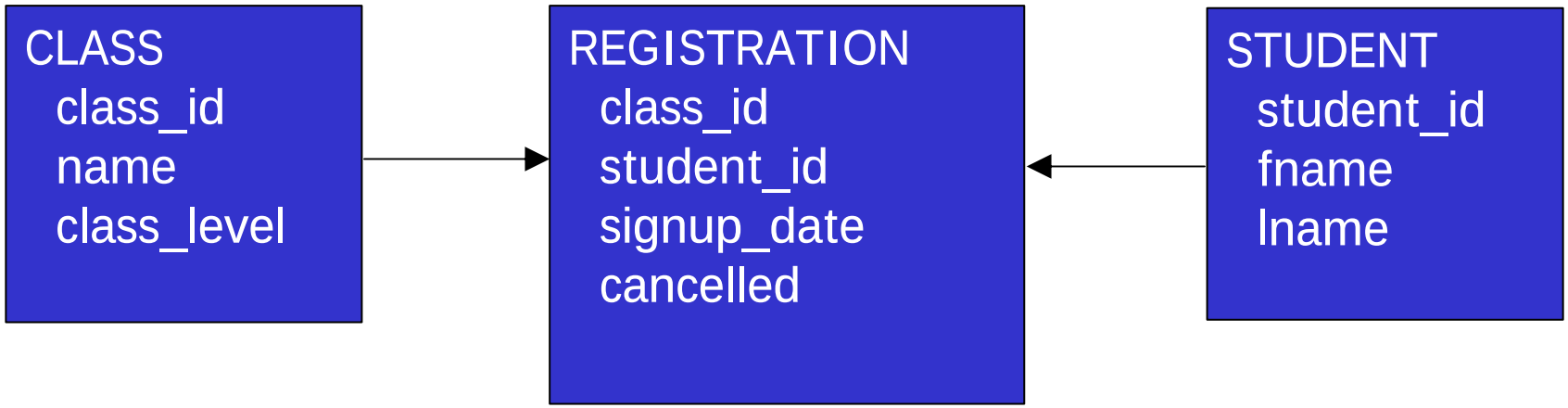
# Example SQL Statement

CONFIO SOFTWARE
Ignite IT Performance™

- Who registered yesterday for SQL Tuning

```
SELECT s.fname, s.lname, r.signup_date
FROM   student s, active_registrations r, class c
WHERE  s.student_id = r.student_id
AND    r.class_id     = c.class_id
AND    UPPER(c.name)  = 'SQL TUNING'
AND    c.class_level  = 101
AND    r.signup_date BETWEEN
          TRUNC(SYSDATE) AND TRUNC(SYSDATE-1)
```

# Relationship



# Gather – Summary

- Execution Plan
  - V$SQL_PLAN
  - Do not use EXPLAIN PLAN
  - DBMS_XPLAN
- Bind Values
  - V$SQL_BIND_CAPTURE
  - Tracing
- Table and Index Statistics
- ERD

# Tune - Execution Plan

CONFIO SOFTWARE
Ignite IT Performance™

- Find the Expensive Steps
- Review Predicates for these Steps
- Evaluate Object Stats
  - Table Definitions
  - Sizes and Row Counts
- Determine Existing Indexes
  - Index Definitions
  - Index Selectivity
- Evaluate Column Stats
  - Limiting Factors from WHERE Clause
- Ensure Join Columns are Indexed

# Execution Plan

CONFIO SOFTWARE
Ignite IT Performance™

```
-------------------------------------------------------------------------------
| Id  | Operation                    | Name         | Rows  | Bytes | Cost |
-------------------------------------------------------------------------------
|   0 | SELECT STATEMENT             |              |       |       |   79|
|   1 |  NESTED LOOPS                |              |     1 |   167 |   79|
|   2 |   NESTED LOOPS               |              |     1 |    81 |   78|
|   3 |    NESTED LOOPS              |              |     1 |    51 |   77|
|   4 |     VIEW                     | VW_SQ_1      |     1 |    35 |   77|
|*  5 |      FILTER                  |              |       |       |     |
|   6 |       HASH GROUP BY          |              |     1 |    17 |   77|
|*  7 |        FILTER                |              |       |       |     |
|*  8 |         TABLE ACCESS FULL    | REGISTRATION |     1 |    17 |   76|
|*  9 |     INDEX UNIQUE SCAN        | SYS_C0020876 |     1 |    16 |    0|
|  10 |    TABLE ACCESS BY INDEX ROWID| STUDENT      |     1 |    30 |    1|
|* 11 |     INDEX UNIQUE SCAN        | SYS_C0020874 |     1 |       |    0|
|* 12 |   TABLE ACCESS BY INDEX ROWID | CLASS        |     1 |    86 |    1|
|* 13 |    INDEX UNIQUE SCAN         | SYS_C0020875 |     1 |       |    0|
-------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------
   5 - filter((MAX("SIGNUP_DATE")>=SYSDATE@! AND MAX("SIGNUP_DATE")<=TRUNC(SYSDATE@!-1)))
   7 - filter(SYSDATE@!<=TRUNC(SYSDATE@!-1))
   8 - filter("CANCELLED"='N')
   9 - access("R1"."STUDENT_ID"="STUDENT_ID" AND "R1"."CLASS_ID"="CLASS_ID" AND
    "SIGNUP_DATE"="VW_COL_1")
       filter(("SIGNUP_DATE">=SYSDATE@! AND "SIGNUP_DATE"<=TRUNC(SYSDATE@!-1)))
  11 - access("S"."STUDENT_ID"="STUDENT_ID")
  12 - filter(("C"."CLASS_LEVEL"=101 AND UPPER("C"."NAME")='SQL TUNING'))
  13 - access("CLASS_ID"="C"."CLASS_ID")
```

# Expensive Steps

CONFIO SOFTWARE
Ignite IT Performance™

```
-------------------------------------------------------------------------------
| Id  | Operation                    | Name         | Rows  | Bytes | Cost |
-------------------------------------------------------------------------------
|   0 | SELECT STATEMENT             |              |       |       |   79|
|   1 |  NESTED LOOPS                |              |     1 |   167 |   79|
|   2 |   NESTED LOOPS               |              |     1 |    81 |   78|
|   3 |    NESTED LOOPS              |              |     1 |    51 |   77|
|   4 |     VIEW                     | VW_SQ_1      |     1 |    35 |   77|
|*  5 |      FILTER                  |              |       |       |     |
|   6 |       HASH GROUP BY          |              |     1 |    17 |   77|
|*  7 |        FILTER                |              |       |       |     |
|*  8 |         TABLE ACCESS FULL    | REGISTRATION |     1 |    17 |   76|
|*  9 |     INDEX UNIQUE SCAN        | SYS_C0020876 |     1 |    16 |    0|
|  10 |    TABLE ACCESS BY INDEX ROWID| STUDENT      |     1 |    30 |    1|
|* 11 |     INDEX UNIQUE SCAN        | SYS_C0020874 |     1 |       |    0|
|* 12 |   TABLE ACCESS BY INDEX ROWID | CLASS        |     1 |    86 |    1|
|* 13 |    INDEX UNIQUE SCAN         | SYS_C0020875 |     1 |       |    0|
-------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------
   5 - filter((MAX("SIGNUP_DATE")>=SYSDATE@! AND MAX("SIGNUP_DATE")<=TRUNC(SYSDATE@!-1)))
   7 - filter(SYSDATE@!<=TRUNC(SYSDATE@!-1))
   8 - filter("CANCELLED"='N')
   9 - access("R1"."STUDENT_ID"="STUDENT_ID" AND "R1"."CLASS_ID"="CLASS_ID" AND
       "SIGNUP_DATE"="VW_COL_1")
       filter(("SIGNUP_DATE">=SYSDATE@! AND "SIGNUP_DATE"<=TRUNC(SYSDATE@!-1)))
  11 - access("S"."STUDENT_ID"="STUDENT_ID")
  12 - filter(("C"."CLASS_LEVEL"=101 AND UPPER("C"."NAME")='SQL TUNING'))
  13 - access("CLASS_ID"="C"."CLASS_ID")
```

# Example SQL Statement

CONFIO SOFTWARE
Ignite IT Performance™

- Who registered for SQL Tuning within last day

```
SELECT s.fname, s.lname, r.signup_date
FROM student s, active_registrations r, class c
WHERE s.student_id = r.student_id
AND r.class_id     = c.class_id
AND UPPER(c.name)  = 'SQL TUNING'
AND c.class_level  = 101
AND r.signup_date BETWEEN
       TRUNC(SYSDATE) AND TRUNC(SYSDATE-1)
```

# ACTIVE_REGISTRATIONS

CONFIO SOFTWARE
Ignite IT Performance™

```
set long 8000
select text from user_views where
view_name='ACTIVE_REGISTRATIONS';

TEXT
----------------------------------------
SELECT student_id, class_id, signup_date
FROM registration r1
WHERE signup_date = (
   SELECT MAX(signup_date)
   FROM  registration r2
   WHERE r1.class_id   = r2.class_id
   AND   r1.student_id = r2.student_id
   AND   r2.cancelled  = 'N')
```

# REGISTRATION Table Data

CONFIO SOFTWARE
Ignite IT Performance™

```
Name                           Null?    Type
------------------------------ -------- ----------------
STUDENT_ID                     NOT NULL NUMBER
CLASS_ID                       NOT NULL NUMBER
SIGNUP_DATE                    NOT NULL DATE
CANCELLED                               CHAR(1)


INDEX_NAME                       UNIQUENES COLUMN_NAME     COLUMN_POSITION
-------------------------------- --------- --------------- ---------------
SYS_C0020876                     UNIQUE    STUDENT_ID                    1
SYS_C0020876                     UNIQUE    CLASS_ID                      2
SYS_C0020876                     UNIQUE    SIGNUP_DATE                   3


COLUMN_NAME                  NUM_DISTINCT  NUM_NULLS    DENSITY SAMPLE_SIZE
---------------------------- ------------ ---------- ---------- -----------
CANCELLED                               2          0         .5        5443
CLASS_ID                              998          0 .001002004        5443
SIGNUP_DATE                         32817          0 .000030472       79983
STUDENT_ID                           9999          0  .00010001       79983
```

# Column Contents

CONFIO SOFTWARE
Ignite IT Performance™

```
select cancelled, count(1)
from registration group by cancelled;

C   COUNT(1)
- ----------
Y        638
N      79345

-------------------------------------------------------------------------
select trunc(signup_date), count(1)
from registration group by trunc(signup_date)

TRUNC(SIGNUP_D   COUNT(1)
-------------- ----------
01/01/09 00:00        100
01/02/09 00:00        290
01/03/09 00:00        107
01/04/09 00:00        845
01/05/09 00:00       3190
01/06/09 00:00       2727
…
01/29/09 00:00       2693

45 Buckets with fairly even distribution.
```

# New Plan

CONFIO SOFTWARE
Ignite IT Performance™

create index reg_sudt on registration(signup_date)

| Id | Operation | Name | Rows | Bytes | Cost |
|---|---|---|---|---|---|
| 0 | SELECT STATEMENT | | | | 10 |
| * 1 | FILTER | | | | |
| 2 | HASH GROUP BY | | 1 | 174 | 10 |
| * 3 | FILTER | | | | |
| * 4 | TABLE ACCESS BY INDEX ROWID | REGISTRATION | 1 | 17 | 3 |
| 5 | NESTED LOOPS | | 1 | 174 | 9 |
| 6 | NESTED LOOPS | | 1 | 157 | 6 |
| 7 | NESTED LOOPS | | 1 | 59 | 5 |
| **8** | **TABLE ACCESS BY INDEX ROWID** | **REGISTRATION** | **1** | **17** | **4** |
| *** 9** | **INDEX RANGE SCAN** | **REG_SUDT** | **2** | | **2** |
| 10 | TABLE ACCESS BY INDEX ROWID | STUDENT | 1 | 42 | 1 |
| * 11 | INDEX UNIQUE SCAN | SYS_C0020874 | 1 | | 0 |
| * 12 | TABLE ACCESS BY INDEX ROWID | CLASS | 1 | 98 | 1 |
| * 13 | INDEX UNIQUE SCAN | SYS_C0020875 | 1 | | 0 |
| * 14 | INDEX RANGE SCAN | SYS_C0020876 | 1 | | 2 |

# What About Other Criteria

CONFIO SOFTWARE
Ignite IT Performance™

- AND UPPER(c.name) = 'SQL TUNING'
  - Should only return one row (or just a few) from CLASS and join to REGISTRATION table
  - Created a function-based index on UPPER(name)
  - Added another index on registration.class_id

```
-------------------------------------------------------------------------------
| Id  | Operation                      | Name         | Rows  | Bytes | Cost |
-------------------------------------------------------------------------------
|   0 | SELECT STATEMENT               |              |       |       |     7|
|*  1 |  FILTER                        |              |       |       |      |
|   2 |   NESTED LOOPS                 |              |     1 |   132 |     4|
|   3 |    NESTED LOOPS                |              |     1 |   102 |     3|
|*  4 |     TABLE ACCESS BY INDEX ROWID| CLASS        |     1 |    86 |     2|
|*  5 |      INDEX RANGE SCAN          | CL_FUNC      |     1 |       |     1|
|*  6 |     INDEX RANGE SCAN           | REG_ALT      |     1 |    16 |     1|
|   7 |      SORT AGGREGATE            |              |     1 |    18 |      |
|*  8 |       TABLE ACCESS BY INDEX ROWID| REGISTRATION |     1 |    18 |     3|
|*  9 |        INDEX RANGE SCAN        | REG_ALT      |     1 |       |     2|
|  10 |    TABLE ACCESS BY INDEX ROWID | STUDENT      |     1 |    30 |     1|
|* 11 |     INDEX UNIQUE SCAN          | SYS_C0020874 |     1 |       |     0|
-------------------------------------------------------------------------------
```

# Query 2

CONFIO SOFTWARE
Ignite IT Performance™

- Who cancelled classes within the week

```
SELECT s.lname, c.name, r.signup_date cancel_date
FROM   registration r, student s, class c
where  r.signup_date between sysdate and sysdate-7
AND    r.cancelled  = 'Y'
AND    r.student_id = s.student_id
AND    r.class_id   = c.class_id
```

- 30% of rows are dated within last week
- No index on CANCELLED column = FTS
- Will an index on CANCELLED column help?
  - Why or why not?

# Query 2 Column Stats

CONFIO SOFTWARE
Ignite IT Performance™

```
select cancelled, count(1)
from registration group by cancelled;

C   COUNT(1)
- ----------
Y        638
N      79345
```

- Oracle will not use an index on this column
  - Unless it has more information
  - CBO assumes an even data distribution
- Histograms give more information to Oracle
  - Based on skewed data, CBO realizes an index would be beneficial
  - Works best with literal values
  - Bind Variables – Oracle peeks first time only

# Query 2 - Histogram

```
dbms_stats.gather_table_stats(
    ownname    => 'STDMGMT',
    tabname    => 'REGISTRATION',
    method_opt=>'FOR COLUMNS cancelled SIZE AUTO')
```

```
-------------------------------------------------------------------------------
| Id  | Operation                    | Name         | Rows  | Bytes | Cost |
-------------------------------------------------------------------------------
|   0 | SELECT STATEMENT             |              |       |       |     7|
|*  1 |  FILTER                      |              |       |       |      |
|*  2 |   TABLE ACCESS BY INDEX ROWID| REGISTRATION |     1 |    17 |     7|
|*  3 |    INDEX RANGE SCAN          | REG_CAN      |   754 |       |     2|
-------------------------------------------------------------------------------
```

# Monitor

CONFIO SOFTWARE
Ignite IT Performance™

- Monitor the improvement
  - Be able to prove that tuning made a difference
  - Take new metrics measurements
  - Compare them to initial readings
  - Brag about the improvements – no one else will
- Monitor for next tuning opportunity
  - Tuning is iterative
  - There are always room for improvements
  - Make sure you tune things that make a difference
- Shameless Product Pitch - Ignite

# Summary

- Identify
  - What is the Bottleneck
  - End-to-End view of performance
  - Simplify
- Gather
  - Metrics – Current Performance
  - Wait Time
  - Execution Plan
  - Object Definitions and Statistics
- Tune
- Monitor
  - New Metrics, Wait Time Profile, Execution Plan

# Myth 1 – Option 1

- Use outer joins vs. NOT IN / NOT EXISTS
- Which class is currently empty?

```
select class_id, name
from class
where class_id not in (
   select class_id from registration)
```

```
-------------------------------------------------------------------------
| Id  | Operation          | Name         | Rows  | Bytes | Cost |
-------------------------------------------------------------------------
|   0 | SELECT STATEMENT   |              |       |       |    88|
|*  1 |  HASH JOIN ANTI    |              |     1 |    68 |    88|
|   2 |   TABLE ACCESS FULL| CLASS        |  1000 | 64000 |    14|
|   3 |   TABLE ACCESS FULL| REGISTRATION | 80056 |   312K|    72|
-------------------------------------------------------------------------
```

# Myth 1 – Option 2

CONFIO SOFTWARE
Ignite IT Performance™

- Try NOT EXISTS vs. NOT IN

```
select class_id, name
from class c
where not exists (
   select 1 from registration r
   where c.class_id = r.class_id)
```

```
-------------------------------------------------------------------------
| Id  | Operation          | Name         | Rows  | Bytes | Cost |
-------------------------------------------------------------------------
|   0 | SELECT STATEMENT   |              |       |       |    88|
|*  1 |  HASH JOIN ANTI    |              |     1 |    68 |    88|
|   2 |   TABLE ACCESS FULL| CLASS        |  1000 | 64000 |    14|
|   3 |   TABLE ACCESS FULL| REGISTRATION | 80056 |   312K|    72|
-------------------------------------------------------------------------
```

# Myth 1 – Option 3

CONFIO SOFTWARE
Ignite IT Performance™

- Try OUTER JOIN
- No Differences with 3 options

```
select c.class_id, c.name
from class c, registration r
where c.class_id = r.class_id (+)
and r.class_id is null
```

```
-------------------------------------------------------------------------
| Id  | Operation          | Name         | Rows  | Bytes | Cost |
-------------------------------------------------------------------------
|   0 | SELECT STATEMENT   |              |       |       |    88|
|*  1 |  HASH JOIN ANTI    |              |     1 |    68 |    88|
|   2 |   TABLE ACCESS FULL| CLASS        |  1000 | 64000 |    14|
|   3 |   TABLE ACCESS FULL| REGISTRATION | 80056 |   312K|    72|
-------------------------------------------------------------------------
```

# Myth 2 – Option 1

- Use MINUS vs. NOT IN
- Which students live in DC area but not in 20002 or 20003 zip
- Cost = 15, LIO = 20

```
select student_id from student
where state in ('VA', 'DC', 'MD')
and zip not in (20002, 20003)
```

```
-------------------------------------------------------------------------
| Id  | Operation                     | Name    | Rows  | Bytes | Cost |
-------------------------------------------------------------------------|
|   0 | SELECT STATEMENT              |         |       |       |    15|
|   1 |  INLIST ITERATOR              |         |       |       |      |
|*  2 |   TABLE ACCESS BY INDEX ROWID| STUDENT |    11 |   110 |    15|
|*  3 |    INDEX RANGE SCAN           | ST_ST   |    11 |       |     3|
-------------------------------------------------------------------------
```

# Myth 2 – Option 2

- Try MINUS vs. NOT IN
- Cost = 20, LIO = 23 – Worse Performance

```
select student_id from student
where state in ('VA', 'DC', 'MD')
minus
select student_id from student
where zip in (20002, 20003)
```

```
-------------------------------------------------------------------------------
| Id  | Operation                           | Name    | Rows  | Bytes | Cost |
-------------------------------------------------------------------------------
|   0 | SELECT STATEMENT                    |         |       |       |    20|
|   1 |  MINUS                              |         |       |       |      |
|   2 |   SORT UNIQUE                       |         |    11 |    77 |    16|
|   3 |    INLIST ITERATOR                  |         |       |       |      |
|   4 |     TABLE ACCESS BY INDEX ROWID     | STUDENT |    11 |    77 |    15|
|*  5 |      INDEX RANGE SCAN               | ST_ST   |    11 |       |     3|
|   6 |   SORT UNIQUE                       |         |     2 |    14 |     4|
|   7 |    INLIST ITERATOR                  |         |       |       |      |
|   8 |     TABLE ACCESS BY INDEX ROWID     | STUDENT |     2 |    14 |     3|
|*  9 |      INDEX RANGE SCAN               | ST_ZIP  |     2 |       |     2|
-------------------------------------------------------------------------------
```

# Confio Software - Monitor

CONFIO SOFTWARE
Ignite IT Performance™

- Developer of Wait-Based Performance Tools
- Igniter Suite
  - Ignite for SQL Server, Oracle, DB2, Sybase
- Provides Help With
  - Identify
  - Gather
  - Monitor
- Based in Colorado, worldwide customers
- Free trial at www.confio.com