# Dissecting SQL Plan Management Options

A SQL Troubleshooting, Stability and Performance Talk

Nov 2018

**Faisal Ijaz**

Fidelity International

For investment professional use only and not for general public distribution

# Contents

SQL Plan Stability in Perspective

SQL Plan Management & Stability Options

What's Inside !!

SQL Plan Baselines

Managing SQL Plan Baselines

Some Challenges

Q&A

# Dissecting SQL Plan Management Options

A SQL Troubleshooting, Stability and Performance Talk

- SQL Plan Stability In Perspective
  - Prevents certain DB environment changes from affecting Application Query Performance
  - Facilitates Database migrations and Version upgrades
  - Help reduce risks from performance changes to critical business operations
  - Has been around for a while and evolved over time
  - Needed more and more as the Optimizer becomes ever more intelligent ☹
  - Query Execution Plan change for a variety of Reasons
    - Optimizer Statistics and Optimizer Parameter
    - Hybrid Workload types
    - Database Upgrades
    - Code or Structural Changes

# Dissecting SQL Plan Management Options

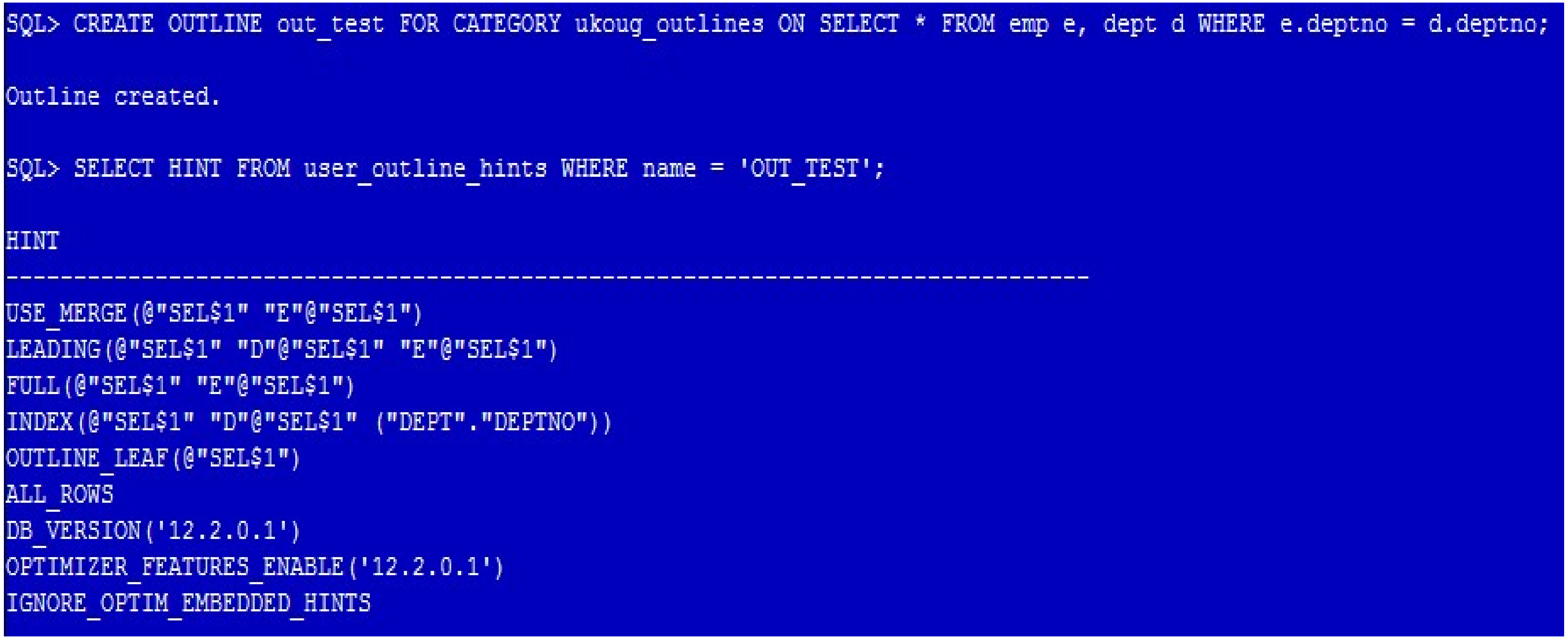
A SQL Troubleshooting, Stability and Performance Talk

- **API SQL Plan Stability Options**
  - Oracle Provides API for SPM Options
    - DBMS_SPM for SQL Baselines
    - DBMS_SQLTUNE for SQL Profiles
    - DBMS_OUTLN for Stored Outlines → Deprecated starting Oracle12c
    - DBMS_SQLDIAG for SQL Patches → More a tool for influencing a plan
    - Hints
  - Enterprise Manager provide GUI to some the above mentioned APIs

# Dissecting SQL Plan Management Options

A SQL Troubleshooting, Stability and Performance Talk

- **What's Inside**
  - Stored Outlines
  - Source DBA|ALL|USER_OUTLINE_HINTS

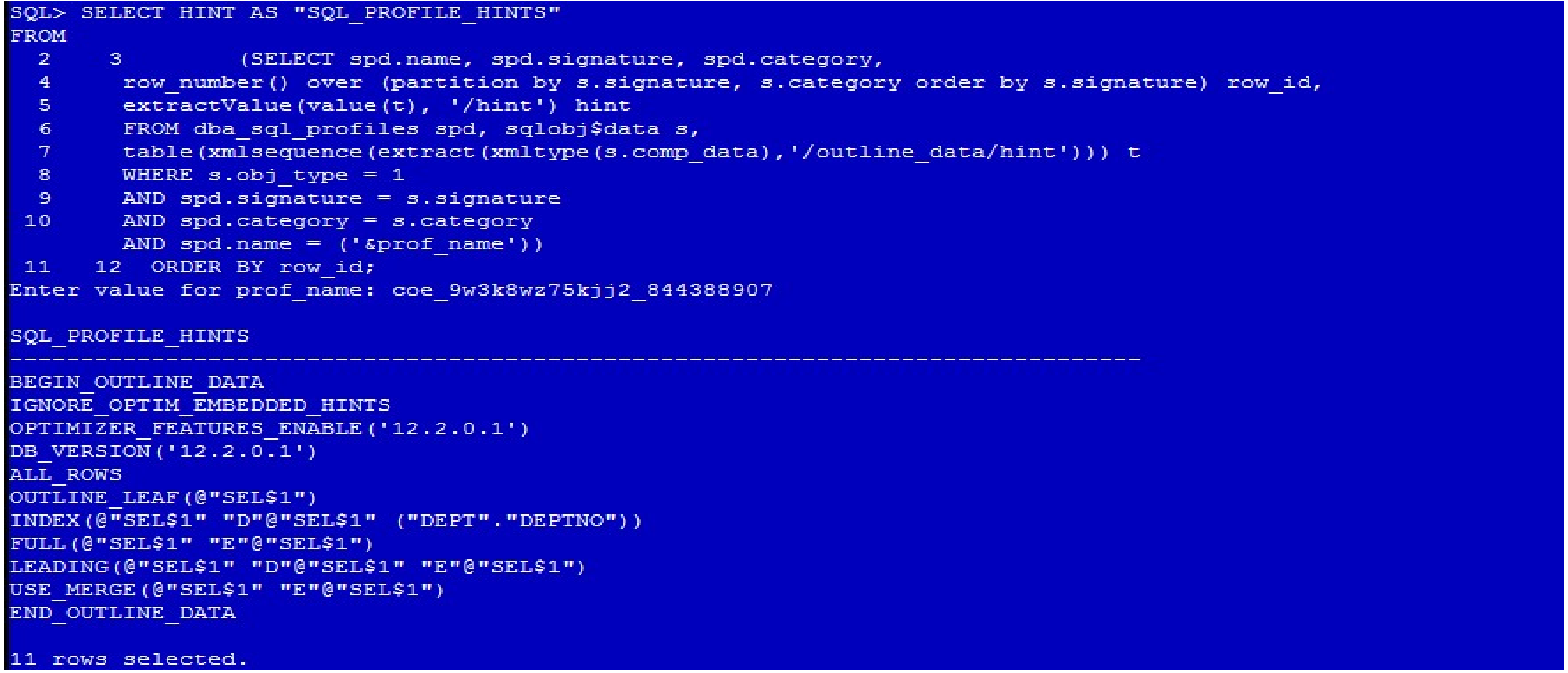


```
SQL> CREATE OUTLINE out_test FOR CATEGORY ukoug_outlines ON SELECT * FROM emp e, dept d WHERE e.deptno = d.deptno;

Outline created.

SQL> SELECT HINT FROM user_outline_hints WHERE name = 'OUT_TEST';

HINT
--------------------------------------------------------------------------------
USE_MERGE(@"SEL$1" "E"@"SEL$1")
LEADING(@"SEL$1" "D"@"SEL$1" "E"@"SEL$1")
FULL(@"SEL$1" "E"@"SEL$1")
INDEX(@"SEL$1" "D"@"SEL$1" ("DEPT"."DEPTNO"))
OUTLINE_LEAF(@"SEL$1")
ALL_ROWS
DB_VERSION('12.2.0.1')
OPTIMIZER_FEATURES_ENABLE('12.2.0.1')
IGNORE_OPTIM_EMBEDDED_HINTS
```

# Dissecting SQL Plan Management Options

A SQL Troubleshooting, Stability and Performance Talk

- **What's Inside**
  - SQL Profiles
  - Source SQLOBJ$DATA  & DBA_SQL_PROFILES

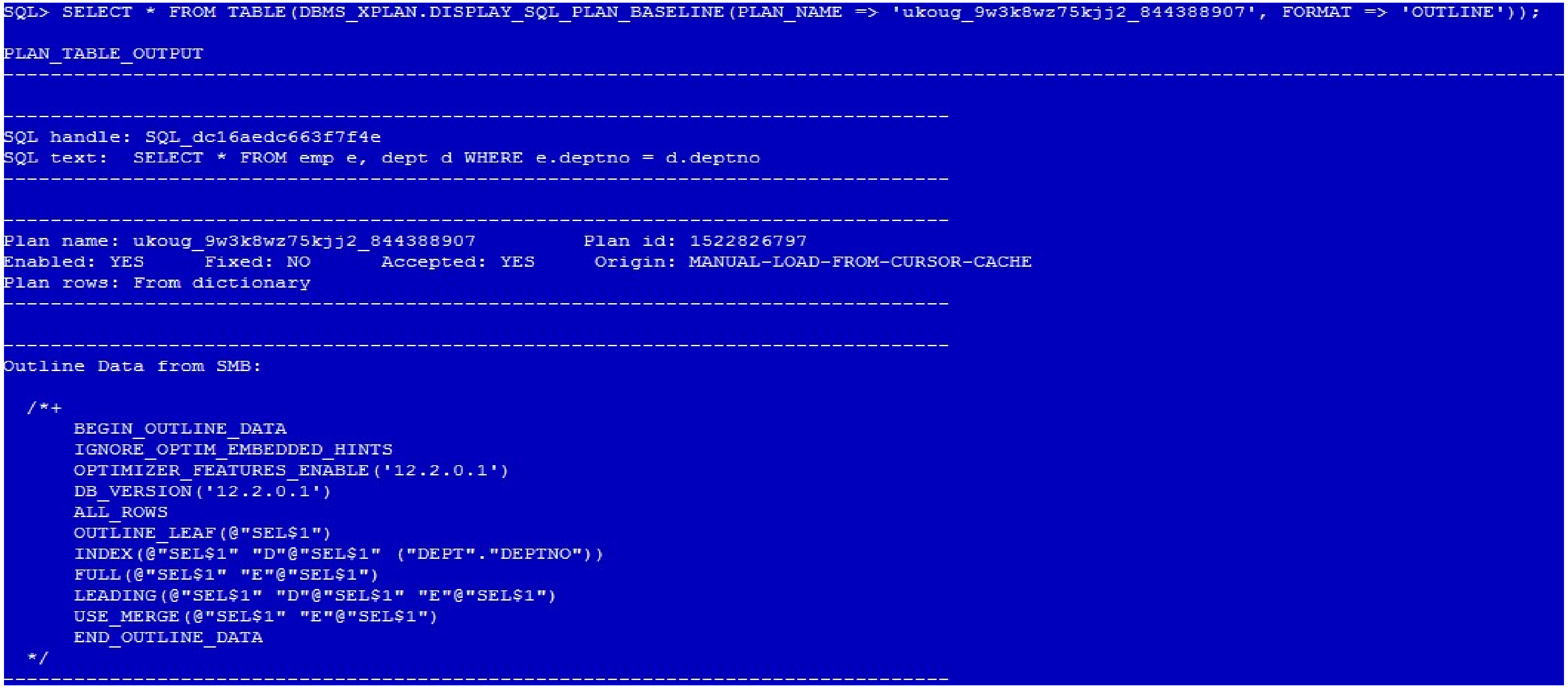


```
SQL> SELECT HINT AS "SQL_PROFILE_HINTS"
FROM
  2    3         (SELECT spd.name, spd.signature, spd.category,
  4      row_number() over (partition by s.signature, s.category order by s.signature) row_id,
  5      extractValue(value(t), '/hint') hint
  6      FROM dba_sql_profiles spd, sqlobj$data s,
  7      table(xmlsequence(extract(xmltype(s.comp_data),'/outline_data/hint'))) t
  8      WHERE s.obj_type = 1
  9      AND spd.signature = s.signature
 10      AND spd.category = s.category
         AND spd.name = ('&prof_name'))
 11   12   ORDER BY row_id;
Enter value for prof_name: coe_9w3k8wz75kjj2_844388907

SQL_PROFILE_HINTS
--------------------------------------------------------------------------------
BEGIN_OUTLINE_DATA
IGNORE_OPTIM_EMBEDDED_HINTS
OPTIMIZER_FEATURES_ENABLE('12.2.0.1')
DB_VERSION('12.2.0.1')
ALL_ROWS
OUTLINE_LEAF(@"SEL$1")
INDEX(@"SEL$1" "D"@"SEL$1" ("DEPT"."DEPTNO"))
FULL(@"SEL$1" "E"@"SEL$1")
LEADING(@"SEL$1" "D"@"SEL$1" "E"@"SEL$1")
USE_MERGE(@"SEL$1" "E"@"SEL$1")
END_OUTLINE_DATA

11 rows selected.
```

# Dissecting SQL Plan Management Options

A SQL Troubleshooting, Stability and Performance Talk

- **What's Inside**
  - SQL Plan Baselines
  - Source ⟶ DBMS_XPLAN with FORMAT => ‘OUTLINE’ or FORMAT =>’ADVANCED’ parameter

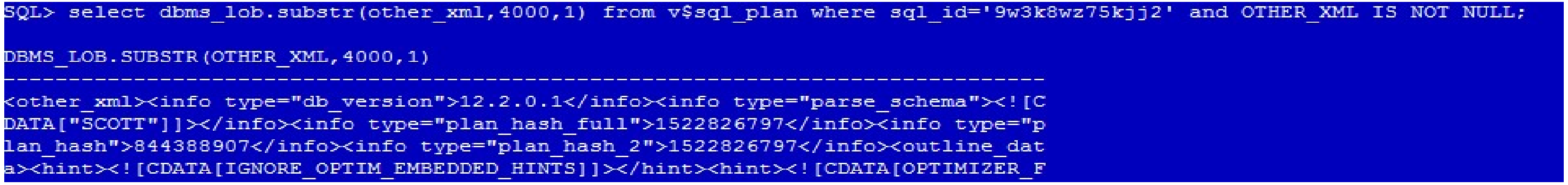


```
SQL> SELECT * FROM TABLE(DBMS_XPLAN.DISPLAY_SQL_PLAN_BASELINE(PLAN_NAME => 'ukoug_9w3k8wz75kjj2_844388907', FORMAT => 'OUTLINE'));

PLAN_TABLE_OUTPUT
------------------------------------------------------------------------------------------------------------------------------------

--------------------------------------------------------------------------------
SQL handle: SQL_dc16aedc663f7f4e
SQL text:   SELECT * FROM emp e, dept d WHERE e.deptno = d.deptno
--------------------------------------------------------------------------------

--------------------------------------------------------------------------------
Plan name: ukoug_9w3k8wz75kjj2_844388907         Plan id: 1522826797
Enabled: YES     Fixed: NO      Accepted: YES     Origin: MANUAL-LOAD-FROM-CURSOR-CACHE
Plan rows: From dictionary
--------------------------------------------------------------------------------

--------------------------------------------------------------------------------
Outline Data from SMB:

  /*+
      BEGIN_OUTLINE_DATA
      IGNORE_OPTIM_EMBEDDED_HINTS
      OPTIMIZER_FEATURES_ENABLE('12.2.0.1')
      DB_VERSION('12.2.0.1')
      ALL_ROWS
      OUTLINE_LEAF(@"SEL$1")
      INDEX(@"SEL$1" "D"@"SEL$1" ("DEPT"."DEPTNO"))
      FULL(@"SEL$1" "E"@"SEL$1")
      LEADING(@"SEL$1" "D"@"SEL$1" "E"@"SEL$1")
      USE_MERGE(@"SEL$1" "E"@"SEL$1")
      END_OUTLINE_DATA
  */
--------------------------------------------------------------------------------
```

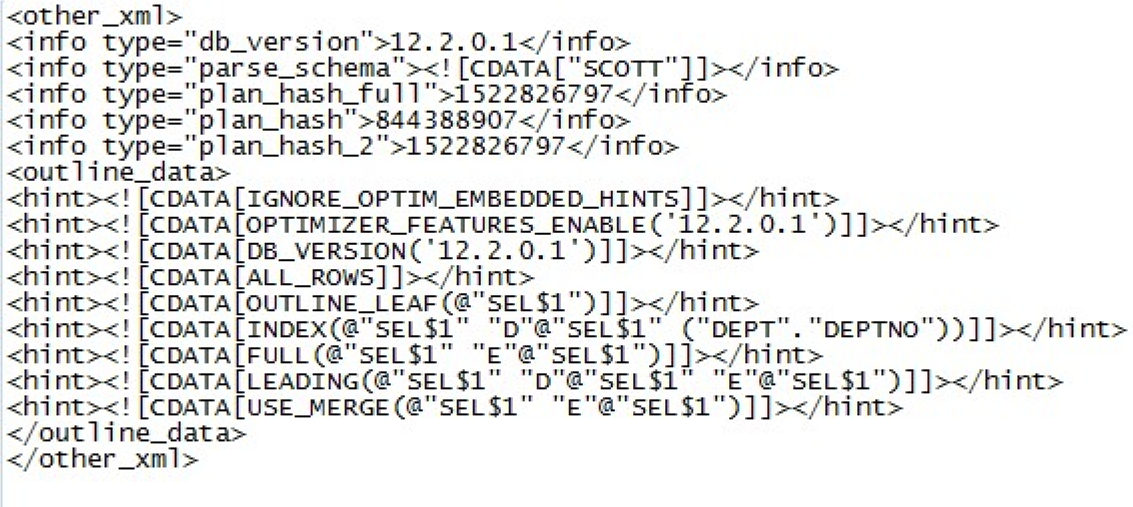
# Dissecting SQL Plan Management Options

A SQL Troubleshooting, Stability and Performance Talk

- **What's Inside**
  - SQL Plan Baselines
  - Source → OTHER_XML columns in V$SQL_PLAN
  - Source → OTHER_XML columns in Staging Table



```
SQL> select dbms_lob.substr(other_xml,4000,1) from v$sql_plan where sql_id='9w3k8wz75kjj2' and OTHER_XML IS NOT NULL;

DBMS_LOB.SUBSTR(OTHER_XML,4000,1)
--------------------------------------------------------------------------------
<other_xml><info type="db_version">12.2.0.1</info><info type="parse_schema"><![C
DATA["SCOTT"]]></info><info type="plan_hash_full">1522826797</info><info type="p
lan_hash">844388907</info><info type="plan_hash_2">1522826797</info><outline_dat
a><hint><![CDATA[IGNORE_OPTIM_EMBEDDED_HINTS]]></hint><hint><![CDATA[OPTIMIZER_F
```



```
<other_xml>
<info type="db_version">12.2.0.1</info>
<info type="parse_schema"><![CDATA["SCOTT"]]></info>
<info type="plan_hash_full">1522826797</info>
<info type="plan_hash">844388907</info>
<info type="plan_hash_2">1522826797</info>
<outline_data>
<hint><![CDATA[IGNORE_OPTIM_EMBEDDED_HINTS]]></hint>
<hint><![CDATA[OPTIMIZER_FEATURES_ENABLE('12.2.0.1')]]></hint>
<hint><![CDATA[DB_VERSION('12.2.0.1')]]></hint>
<hint><![CDATA[ALL_ROWS]]></hint>
<hint><![CDATA[OUTLINE_LEAF(@"SEL$1")]]></hint>
<hint><![CDATA[INDEX(@"SEL$1" "D"@"SEL$1" ("DEPT"."DEPTNO"))]]></hint>
<hint><![CDATA[FULL(@"SEL$1" "E"@"SEL$1")]]></hint>
<hint><![CDATA[LEADING(@"SEL$1" "D"@"SEL$1" "E"@"SEL$1")]]></hint>
<hint><![CDATA[USE_MERGE(@"SEL$1" "E"@"SEL$1")]]></hint>
</outline_data>
</other_xml>
```

# Dissecting SQL Plan Management Options

A SQL Troubleshooting, Stability and Performance Talk

- **Hints**
  - Lets us control decisions that optimizer makes
  - Occasions where we understand our data or situations better
- **Challenges with Hints**
  - Lack of proper Understanding
  - Improper Documentation
  - Un Documented Hints
- **Some Information Available in the Database**
  - V$SQL_HINT
  - V$SQL_FEATURE
  - V$SQL_FEATURE_HIERARCHY

# Dissecting SQL Plan Management Options

A SQL Troubleshooting, Stability and Performance Talk

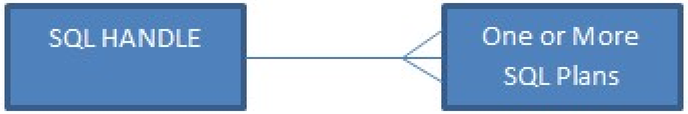
- **SQL Plan Baselines**
  - Core of Oracle SQL Plan Management
  - Ensure Plan Stability (almost…perhaps not always)
  - SQL Plan Baseline Source
    - Capture Automatically if OPTIMZIER_SQL_PLAN_BASELINES is set to TRUE
    - Capture Manually from
      - ➢ Cursor Cache → DBMS_SPM.LOAD_PLANS_FROM_CURSOR_CACHE
      - ➢ SQL Tuning Set → DBMS_SPM.LOAD_PLANS_FROM_SQLSET
      - ➢ Baseline Staging → DBMS_SPM.UNPACK_STGTAB_BASELINE
      - ➢ AWR → DBMS_SPM.LOAD_PLANS_FROM_AWR **(New in 12.2)**

# Dissecting SQL Plan Management Options

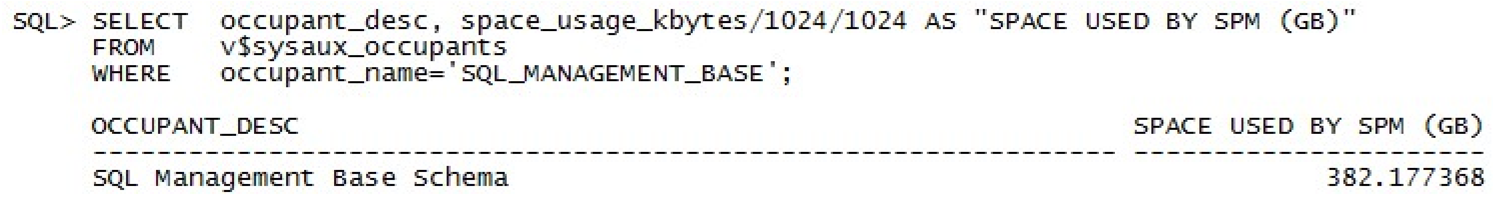
A SQL Troubleshooting, Stability and Performance Talk

- **Managing SQL Plan Baselines – Things we know**

  - A Baseline can have multiple plans stored in it



  - SQL for which Baseline is once created will keep getting new Plans in the SQL Handle
  - Baselines are kept in SYSAUX
  - How space is SPM consuming in SYSAUX



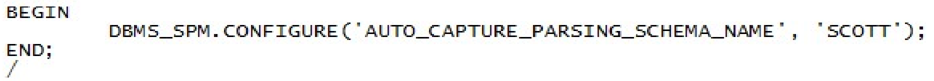
```
SQL> SELECT   occupant_desc, space_usage_kbytes/1024/1024 AS "SPACE USED BY SPM (GB)"
     FROM     v$sysaux_occupants
     WHERE    occupant_name='SQL_MANAGEMENT_BASE';

     OCCUPANT_DESC                                                    SPACE USED BY SPM (GB)
     ---------------------------------------------------------------- ----------------------
     SQL Management Base Schema                                                   382.177368
```

  - Good MOS Notes around Space Management
    - SQL Plan Control Page Hangs or Fails to return Results as a Result of Excessive Baselines (Doc ID 1518749.1)
    - How to Drop Plans from the SQL Plan Management (SPM) Repository (Doc ID 790039.1)
    - Reducing the Space Usage of the SQL Management Base in the SYSAUX Tablespace (Doc ID 1499542.1)
    - **baseline_repro_fail** Tag in OTHER_XML hint from Oracle18c (https://blog.dbi-services.com/18c-dbms_xplan-note-about-failed-sql-plan-baseline/)

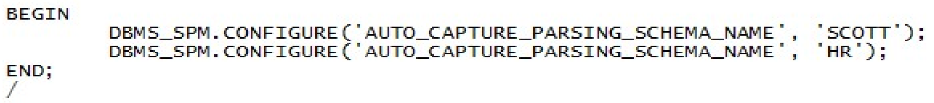
# Dissecting SQL Plan Management Options

A SQL Troubleshooting, Stability and Performance Talk

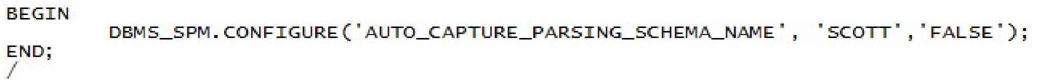
- **Managing SQL Plan Baselines**
- DBMS_SPM.CONFIGURE Procedure manages the baseline Retention and Capture
  - ➢ SPACE_BUDGET_PRECENT
  - ➢ PLAN_RENTEION_WEEKS
  - ➢ AUTO_CAPTURE_PARSING_SCHEMA_NAME **(New in 12.2)**



```
BEGIN
        DBMS_SPM.CONFIGURE('AUTO_CAPTURE_PARSING_SCHEMA_NAME', 'SCOTT');
END;
/
```



```
BEGIN
        DBMS_SPM.CONFIGURE('AUTO_CAPTURE_PARSING_SCHEMA_NAME', 'SCOTT');
        DBMS_SPM.CONFIGURE('AUTO_CAPTURE_PARSING_SCHEMA_NAME', 'HR');
END;
/
```



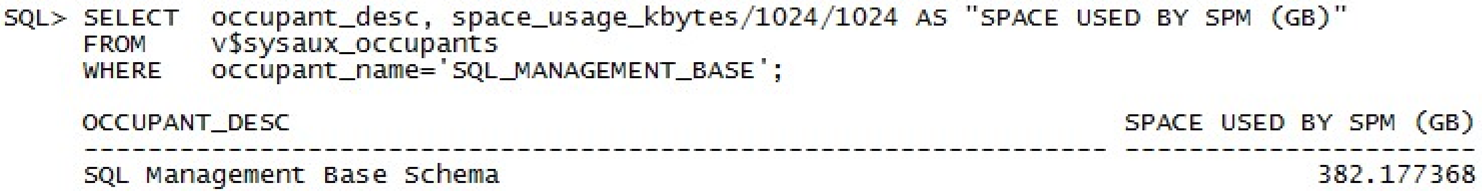
```
BEGIN
        DBMS_SPM.CONFIGURE('AUTO_CAPTURE_PARSING_SCHEMA_NAME', 'SCOTT','FALSE');
END;
/
```

  - ➢ AUTO_CAPTURE_MODULE **(New in 12.2)**
  - ➢ AUTO_CAPTURE_ACTION **(New in 12.2)**
  - ➢ AUTO_CAPTURE_SQL_TEXT **(New in 12.2)**

# Dissecting SQL Plan Management Options

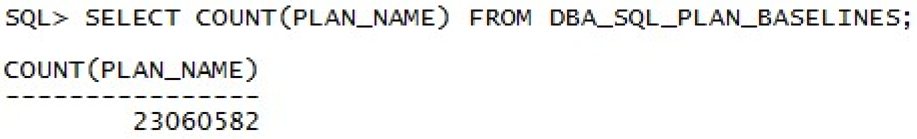
A SQL Troubleshooting, Stability and Performance Talk

- **Some Practical Challenges**
  - Having **OPTIMIZER_CAPTURE_SQL_PLAN_BASELINES**=TRUE
    - ➢ Storage

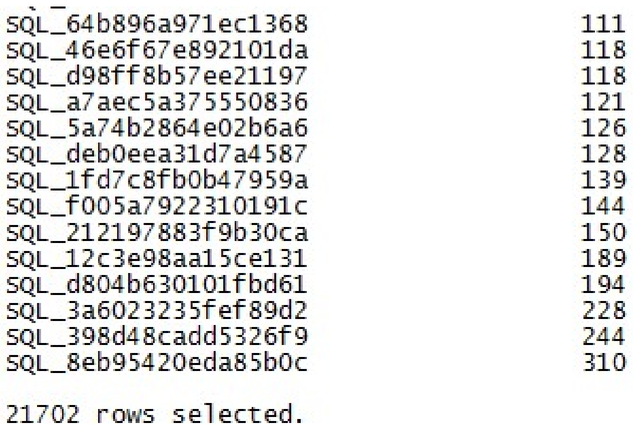


```
SQL> SELECT  occupant_desc, space_usage_kbytes/1024/1024 AS "SPACE USED BY SPM (GB)"
     FROM    v$sysaux_occupants
     WHERE   occupant_name='SQL_MANAGEMENT_BASE';

     OCCUPANT_DESC                                                    SPACE USED BY SPM (GB)
     ---------------------------------------------------------------- ----------------------
     SQL Management Base Schema                                                   382.177368
```

- ➢ Large Number of Baseline Plans



```
SQL> SELECT COUNT(PLAN_NAME) FROM DBA_SQL_PLAN_BASELINES;

COUNT(PLAN_NAME)
----------------
        23060582
```



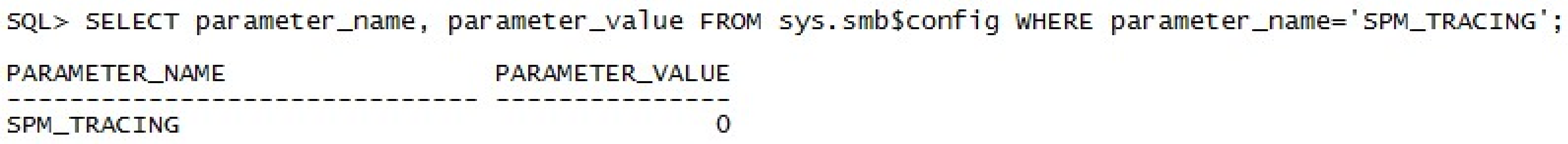
```
SQL_64b896a971ec1368                 111
SQL_46e6f67e892101da                 118
SQL_d98ff8b57ee21197                 118
SQL_a7aec5a375550836                 121
SQL_5a74b2864e02b6a6                 126
SQL_deb0eea31d7a4587                 128
SQL_1fd7c8fb0b47959a                 139
SQL_f005a7922310191c                 144
SQL_212197883f9b30ca                 150
SQL_12c3e98aa15ce131                 189
SQL_d804b630101fbd61                 194
SQL_3a6023235fef89d2                 228
SQL_398d48cadd5326f9                 244
SQL_8eb95420eda85b0c                 310

21702 rows selected.
```

- ➢ Plan Handles with numerous plans
- ➢ Wait due large number of BLs

# Dissecting SQL Plan Management Options

A SQL Troubleshooting, Stability and Performance Talk

- **Some Practical Challenges**
  - Why was my Baseline Not Used
    - ➢ Usual cases where Plan was not Re-Producible
    - ➢ Few bug which one should be aware of
    - ➢ Things to Consider When Upgrading From 12.1.0.1 to Avoid Problems with SQL Plan Management (SPM) (Doc ID 2035897.1)
    - ➢ Things to Consider for 12.1.0.2 to Avoid Problems with SQL Plan Management (SPM) (Doc ID 2035898.1)
  - How to Diagnose Issues where a SQL Plan Management (SPM) Baseline is Not Used (Doc ID 1663691.1)
    - ➢ Enable SPM Trace → DBMS_SPM.CONFIGURE(‘SPM_TRACING',0/1);



```
SQL> SELECT parameter_name, parameter_value FROM sys.smb$config WHERE parameter_name='SPM_TRACING';

PARAMETER_NAME                 PARAMETER_VALUE
------------------------------ ---------------
SPM_TRACING                                  0
```

- ➢ Optimizer Trace → alter session set events 'trace[sql_optimizer.*]';
- ➢ SQLT or SQLHC

# Dissecting SQL Plan Management Options

A SQL Troubleshooting, Stability and Performance Talk