Dec. 6th, 2007

DATA SOFTECH INC.

Complete Database Solution Provider

ORACLE 11g SQL Plan Management

Inderpal S. Johal,
President

Agenda

- Execution plan and factor affecting it
- How optimizer behaves without 11g Sql Plan Management
- Why SQL Plan Management
- 11g SQL PLAN Management Pre-Requisites
- SQL Plan Management Architecture
- SPM flow
- Enabling SQL Plan Baselines
- Loading/Dropping/Displaying SQL Plan Baseline
- Space Usage/Purging for SPM
- SPM View
- Q&A

Execution Plan and factor affecting it

- Execution plan define how Oracle find or writes the data to satisfy DML statement
- Execution plan define whether Oracle will use Index or Which Index to be used and the like

Factor affecting Execution Plan and then the Business

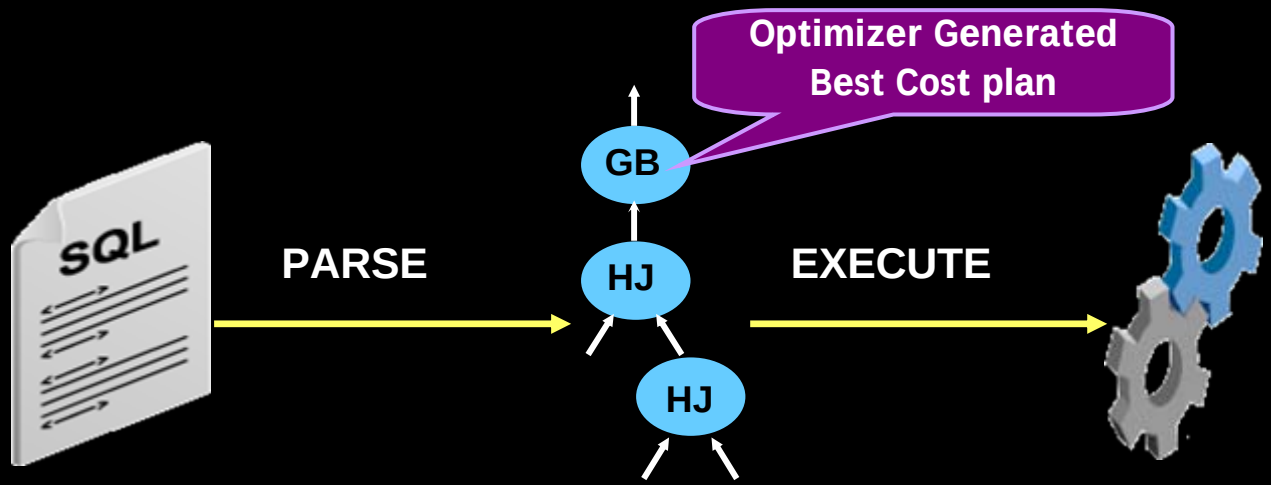
- Optimizer Version due to Database upgrade
- Optimizer Statistics
- Optimizer parameters
- Schema definition Changes

Solution : Prior to 11g

- Freeze critical plans using Stored Outlines/Profile
- Freeze Statistics

Without SQL Plan Management ???

First Time SQL is parsed and an execution plan is generated

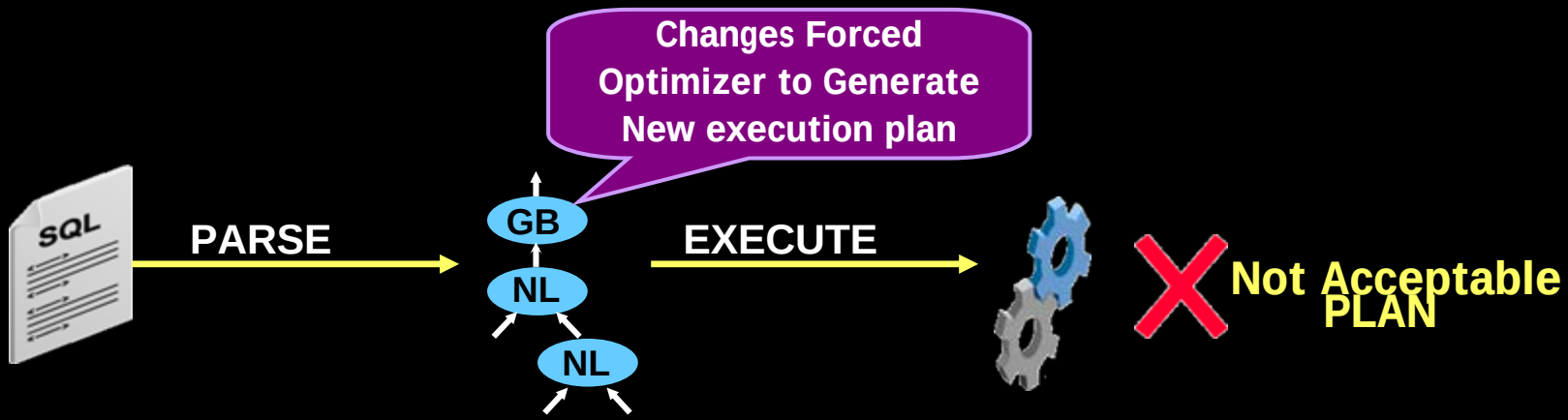


Verified that Plan is giving Acceptable performance

Without SQL Plan Management ???

Database Server Environment Changed

- Upgrade Database or Stats re-Gathered or Parameter Change or so
- New Plan generated regardless of performance impact
- Does new Plan meet required performance criteria ???



Why SQL Plan Management

- Provide Guaranteed Plan Stability
- Maintain good execution plan in SQL Baseline
- Only known or verified plans are used
- Plans are verified
  - Manually
  - Automatically by daily maintenance Job
- Only comparable or better plan are included to SQL Baseline
- Main benefit is the performance stability of the system

11g SQL Plan Management Pre-Requisites

- Initialization Parameter Setting

  1. optimizer_use_sql_plan_baselines=TRUE

     [Default is TRUE]

     Enable Optimizer to use SQL Plan baselines

  2. optimizer_capture_sql_plan_baselines=TRUE

     [Default is FALSE]

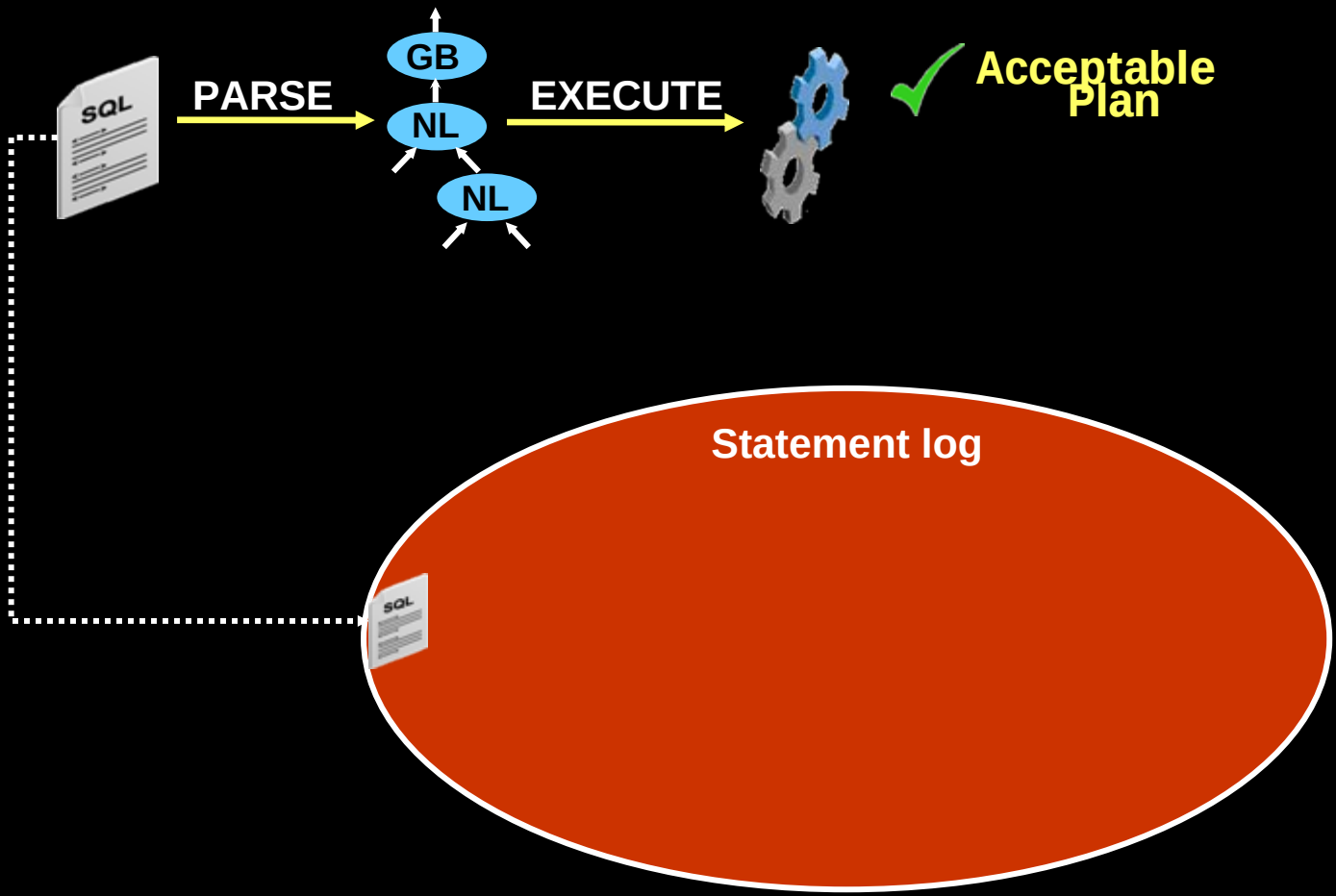
     [Require Oracle Tuning Pack]

     Enable Oracle to Capture SQL Plan to be included in baseline

- SYSAUX Tablespace should be ONLINE

11g SQL Plan Management Architecture

OPTIMIZER_CAPTURE_SQL_PLAN_BASELINES=TRUE



SQL Plan Management Architecture contd.

Execute the SQL query First time from SCOTT Schema

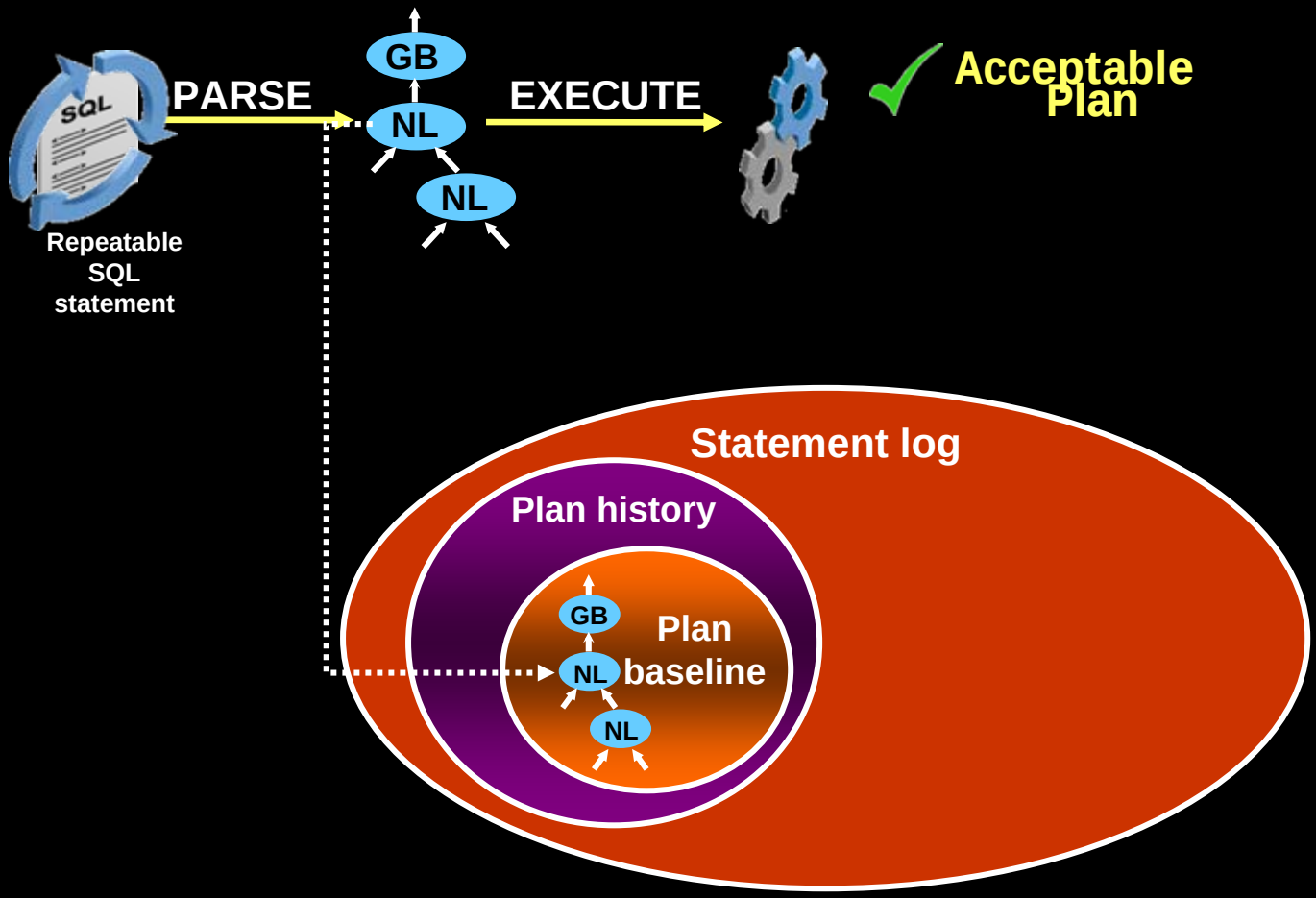
```
SQL> SELECT  sal TESTING
        FROM emp ;
```

Check the SQL is available in SQL PLAN Baseline

```
SQL> SELECT sql_text,  sql_handle,  optimizer_cost,  enabled, fixed, accepted
        FROM  DBA_SQL_PLAN_BASELINES
        WHERE sql_text LIKE '%TESTING%'
no rows selected
```

SQL Plan Management Architecture

OPTIMIZER_CAPTURE_SQL_PLAN_BASELINES=TRUE



SQL Plan Management Architecture contd.

Execute the SQL query SECOND time from SCOTT Schema

```
SQL> SELECT  sal TESTING
       FROM emp ;
```

Check the SQL is available in SQL PLAN Baseline

```
SQL> SELECT plan_name, sql_text,  sql_handle,  optimizer_cost,
       enabled, fixed, accepted
       FROM  DBA_SQL_PLAN_BASELINES
       WHERE sql_text LIKE '%TESTING%'
```

PLAN_NAME	SQL_TEXT	SQL_HANDLE	OPTIMIZER_COST	ENA	FIX	ACC
SYS_SQL_PLAN_5d47d3d	select sal TESTING from emp;	SYS_SQL_f9ec8ef45d	2	YES	NO	YES

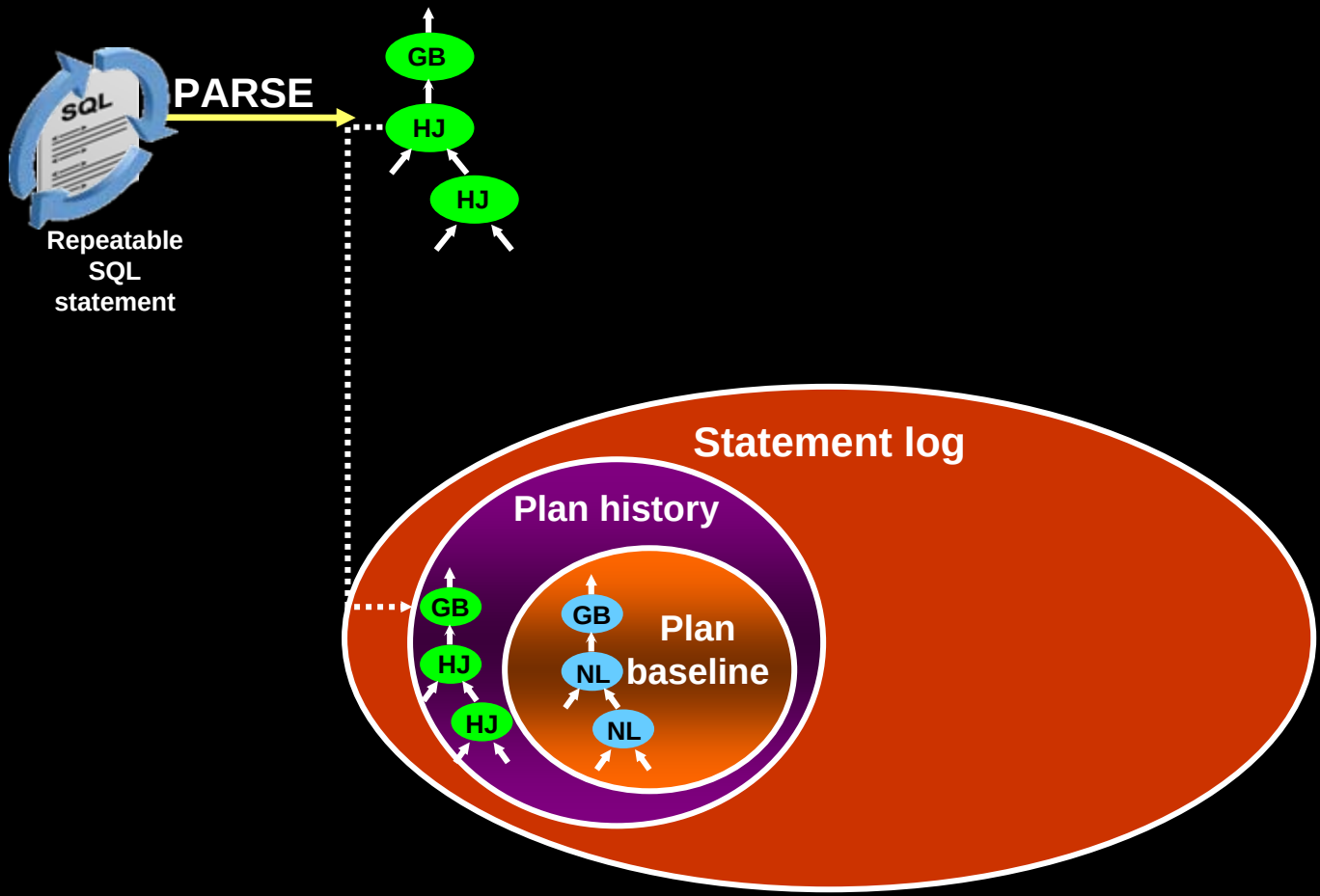
SQL Plan Management Architecture contd.

Execute the SQL query SECOND time from SCOTT Schema

```
SQL> SELECT  *  FROM  TABLE ( DBMS_PLAN.DISPLAY_SQL_PLAN_BASELINE(
        SQL_HANDLE=>' SYS_SQL_f9ec8ef45d47d320',
       FORMAT=> 'BASIC'));
PLAN_TABLE_OUTPUT
--------------------------------------------------------------------------------
SQL handle: SYS_SQL_f9ec8ef45d
SQL text: select sal TESTING from emp
--------------------------------------------------------------------------------
Plan name: SYS_SQL_PLAN_5d47d3d
Enabled: YES     Fixed: NO      Accepted: NO      Origin: AUTO-CAPTURE
--------------------------------------------------------------------------------
| Id  | Operation         | Name |
---------------------------------
|   0 | SELECT STATEMENT  |      |
|   1 |  SORT AGGREGATE   |      |
|   2 |   INDEX RANGE SCAN| III  |
--------------------------------------------------------------------------------
```

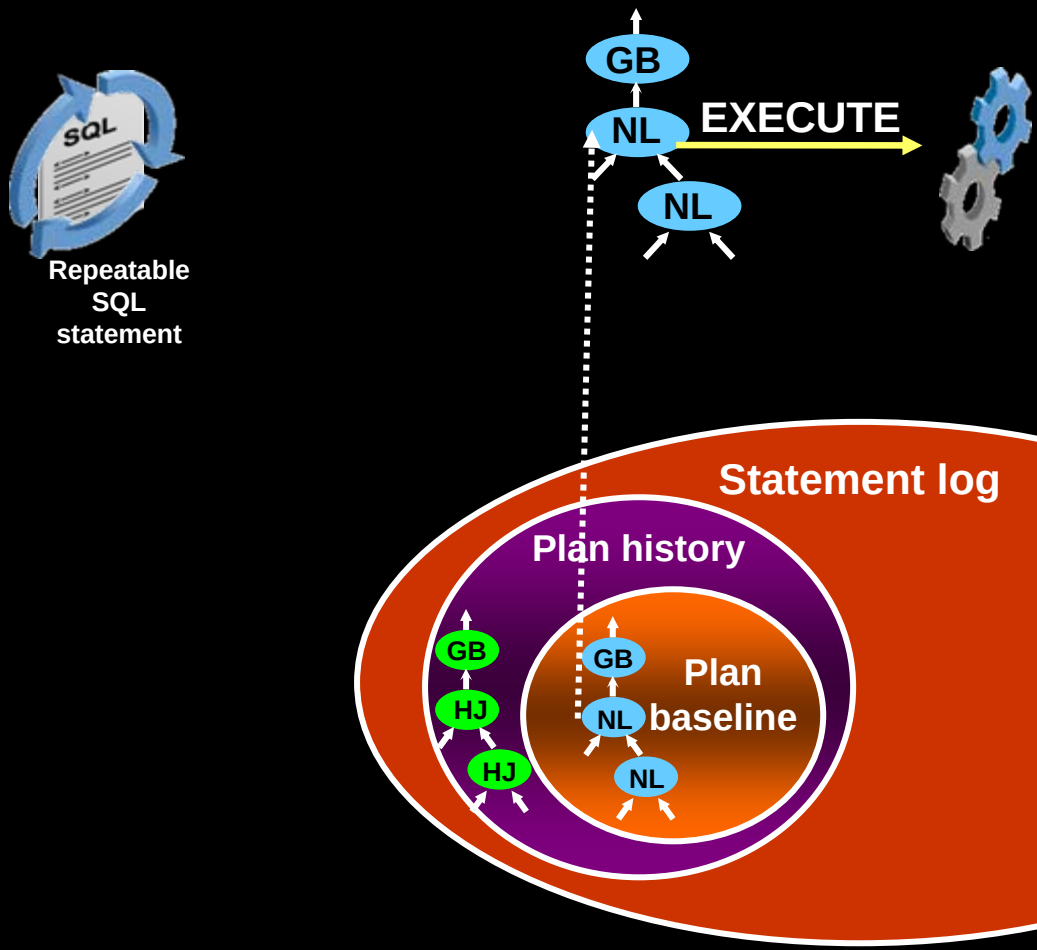
SQL Plan Management Architecture

Make the Change in System like Index or Collect Stats or Change Optimizer version



SQL Plan Management Architecture

Make the Change in System like Index or Collect Stats or Change Optimizer version



SQL Plan Management Architecture contd.

Execute the SQL query again from SCOTT Schema

```
SQL> SELECT  sal TESTING
       FROM emp ;
```

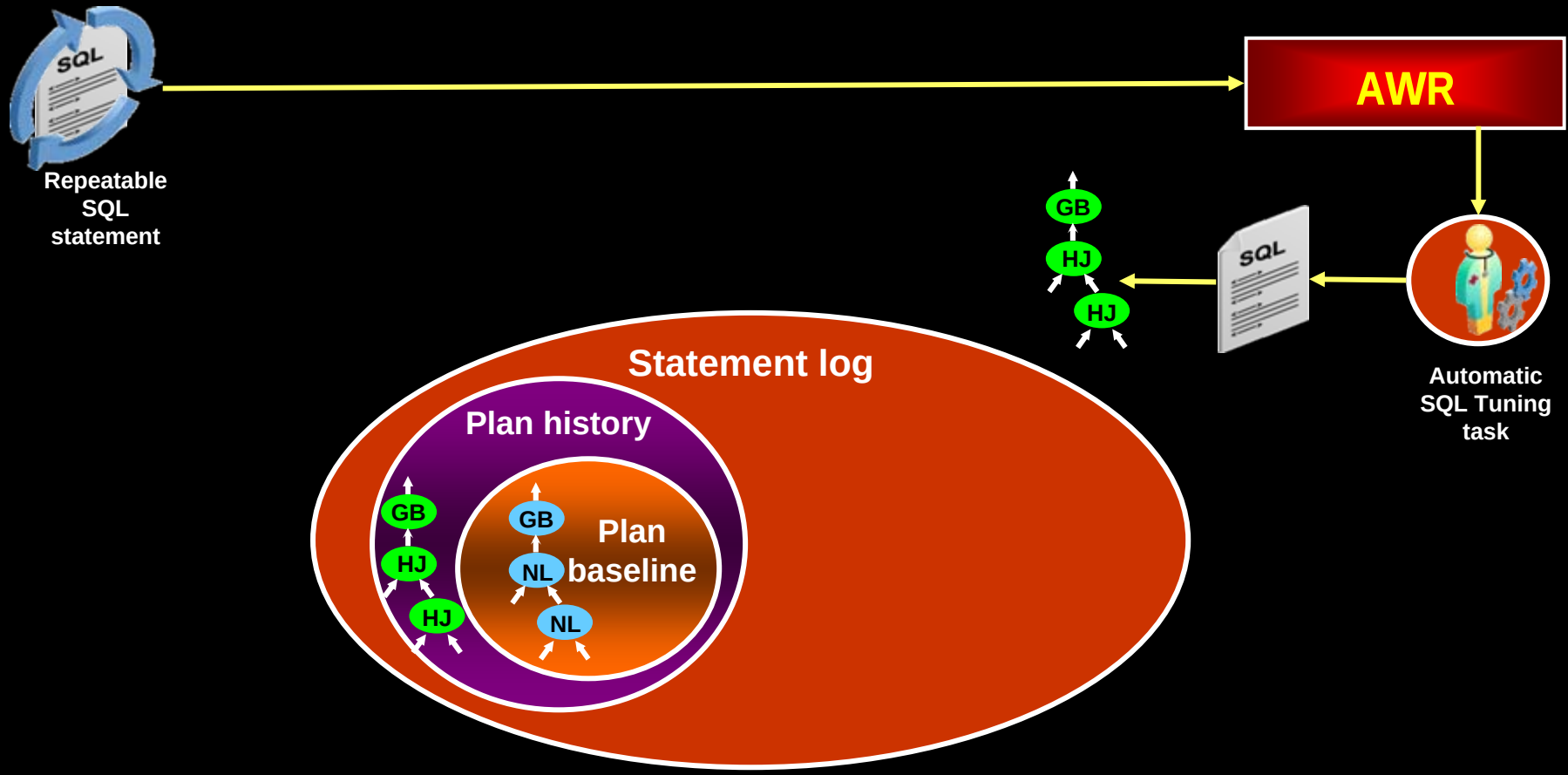
Check the SQL is available in SQL PLAN Baseline

```
SQL> SELECT plan_name, sql_text,  sql_handle,  optimizer_cost,
       enabled, fixed, accepted
       FROM  DBA_SQL_PLAN_BASELINES
       WHERE sql_text LIKE '%TESTING%'
```

PLAN_NAME	SQL_TEXT	SQL_HANDLE	OPTIMIZER_COST	ENA	FIX	ACC
SYS_SQL_PLAN_5d47d3d	select sal TESTING from emp;	SYS_SQL_f9ec8ef45d	2	YES	NO	YES
SYS_SQL_PLAN_6d1231d	select sal TESTING from emp;	SYS_SQL_f8de7fr6dd	1	YES	NO	NO

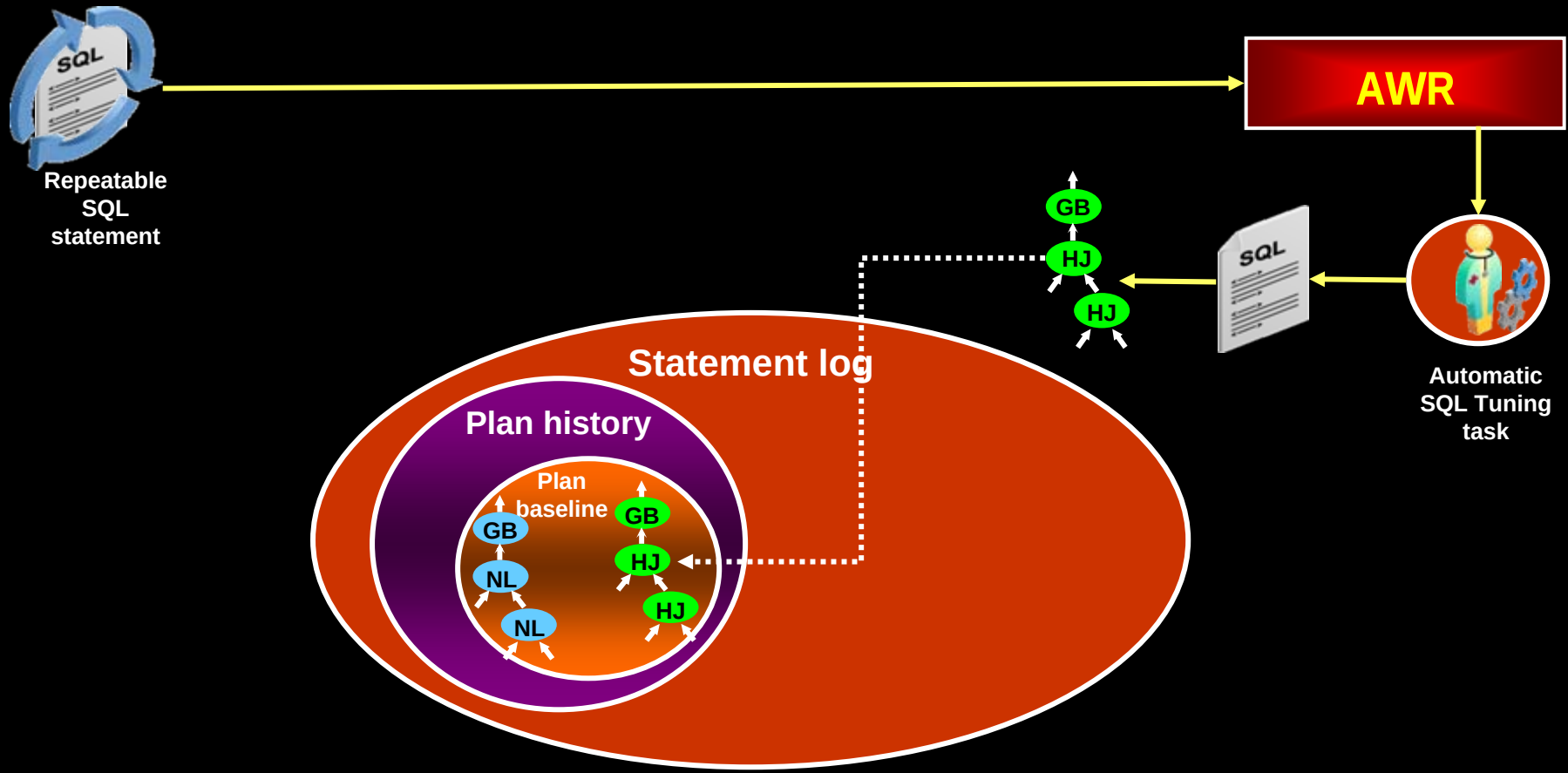
SQL Plan Management Architecture

SQL Repeated Several Times result High Load SQL in AWR



SQL Plan Management Architecture

SQL Repeated Several Times result High Load SQL in AWR



SQL Plan Management Architecture contd.

Execute the SQL query again from SCOTT Schema

```
SQL> SELECT  sal TESTING
       FROM emp ;
```

Check the SQL is available in SQL PLAN Baseline

```
SQL> SELECT plan_name, sql_text,  sql_handle,  optimizer_cost,
       enabled, fixed, accepted
       FROM  DBA_SQL_PLAN_BASELINES
       WHERE sql_text LIKE '%TESTING%'
```

PLAN_NAME	SQL_TEXT	SQL_HANDLE	OPTIMIZER_COST	ENA	FIX	ACC
SYS_SQL_PLAN_5d47d3d	select sal TESTING from emp;	SYS_SQL_f9ec8ef45d	2	YES	NO	YES
SYS_SQL_PLAN_6d1231d	select sal TESTING from emp;	SYS_SQL_f8de7fr6dd	1	YES	NO	YES

SQL Plan Management Architecture contd.

Check the SQL is available in SQL PLAN Baseline

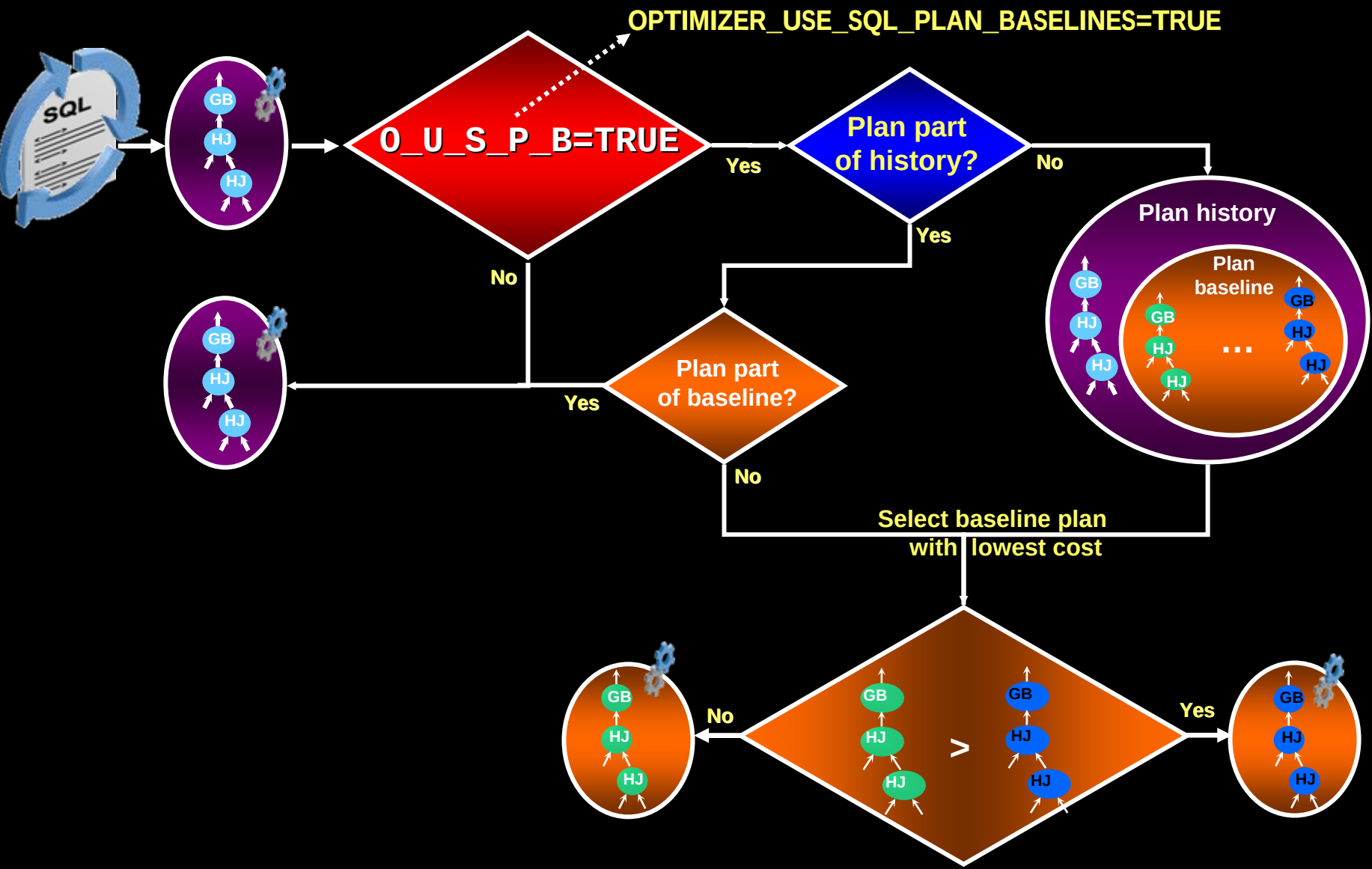
```
SQL> SELECT plan_name, sql_text,  sql_handle,  optimizer_cost,
        enabled, fixed, accepted
        FROM  DBA_SQL_PLAN_BASELINES
        WHERE sql_text LIKE '%TESTING%'
```

PLAN_NAME	SQL_TEXT	SQL_HANDLE	OPTIMIZER_COST	ENA	FIX	ACC
SYS_SQL_PLAN_5d47d3d	select sal TESTING from emp;	SYS_SQL_f9ec8ef45d	2	YES	NO	YES
SYS_SQL_PLAN_6d1231d	select sal TESTING from emp;	SYS_SQL_f8de7fr6dd	1	YES	NO	No

```
SQL> DECLARE
            indy NATURAL;
      BEGIN
            indy := DBMS_SPM.ALTER_SQL_PLAN_BASELINE ('SYS_SQL_f9ec8ef45d47d320',
                        attribute_name => 'ACCEPTED',
                        attribute_value=>'YES');
      END;
/
```

PLAN_NAME	SQL_TEXT	SQL_HANDLE	OPTIMIZER_COST	ENA	FIX	ACC
SYS_SQL_PLAN_5d47d3d	select sal TESTING from emp;	SYS_SQL_f9ec8ef45d	2	YES	NO	YES
SYS_SQL_PLAN_6d1231d	select sal TESTING from emp;	SYS_SQL_f8de7fr6dd	1	YES	NO	YES

11g SQL Plan Selection Flow



More ways to Load SQL Baseline

Already covered SQL Baseline methods

- Using init.ora parameter OPTIMIZER_CAPTURE_SQL_PLAN_BASELINES=TRUE
- Using 11g Daily Automatic SQL Tuning Task
- Using DBMS_SPM.ALTER_SQL_PLAN_BASELINE to move Plan from Plan History to PLAN Baseline

More Manual method to Load SQL Baseline

- Load SQL Plans from Cursor Cache
- Export /Import from one Database to another

Enabling SQL Plan Baseline

Check the SQL is available in SQL PLAN Baseline

```
SQL> SELECT plan_name, sql_text,  sql_handle,  optimizer_cost,
        enabled, fixed, accepted
        FROM  DBA_SQL_PLAN_BASELINES
        WHERE sql_text LIKE '%TESTING%'
```

PLAN_NAME	SQL_TEXT	SQL_HANDLE	OPTIMIZER_COST	ENA	FIX	ACC
SYS_SQL_PLAN_5d47d3d	select sal TESTING from emp;	SYS_SQL_f9ec8ef45d	2	YES	NO	YES
SYS_SQL_PLAN_6d1231d	select sal TESTING from emp;	SYS_SQL_f8de7fr6dd	1	YES	NO	No

```
SQL>
  1  SET SERVEROUTPUT ON
  2  SET LONG 10000
  3  DECLARE
  4      report clob;
  5  BEGIN
  6      report := DBMS_SPM.EVOLVE_SQL_PLAN_BASELINE(
  7                    sql_handle => ' SYS_SQL_f8de7fr6dd ');
  8      DBMS_OUTPUT.PUT_LINE(report);
  9* END;
/
```

Loading SQL Plan from Cursor Cache

```
SQL> SELECT sql_id,sql_text  FROM v$sql  WHERE sql_text LIKE  '%INDYTEST%';
SQL_ID          SQL_TEXT
364s6t88vhd     select sql_id,sql_text from v$sql where sql_text like '%INDYTEST%'
```

Load SQL Plan from Cursor Cache

```
DECLARE
      my_plans pls_integer;
BEGIN
      my_plans := DBMS_SPM.LOAD_PLANS_FROM_CURSOR_CACHE( sql_id => 364s6t88vhd ');
END;
/
```

Check the SQL is available in SQL PLAN Baseline

```
SQL> SELECT plan_name, sql_text,  sql_handle,  optimizer_cost,
       enabled, fixed, accepted
       FROM  DBA_SQL_PLAN_BASELINES
       WHERE sql_text LIKE '%TESTING%'
```

PLAN_NAME	SQL_TEXT	SQL_HANDLE	ORIGIN	ENA	ACC
SYS_SQL_PLAN_5d47d3d	select sal TESTING from emp;	SYS_SQL_f9ec8ef45d	MANUAL-LOAD	YES	YES
SYS_SQL_PLAN_6d1231d	select sal TESTING from emp;	SYS_SQL_f8de7fr6dd	AUTO-CAPTURE	YES	YES

Drop SQL Plan from Baselines

Check the SQL is available in SQL PLAN Baseline

```
SQL> SELECT plan_name, sql_text,  sql_handle,  optimizer_cost,
        enabled, fixed, accepted
        FROM  DBA_SQL_PLAN_BASELINES
        WHERE sql_text LIKE '%TESTING%'
```

PLAN_NAME	SQL_TEXT	SQL_HANDLE	ORIGIN	ENA	ACC
SYS_SQL_PLAN_5d47d3d	select sal TESTING from emp;	SYS_SQL_f9ec8ef45d	MANUAL-LOAD	YES	YES

Drop the Above SQL plan from the Baseline

```
DECLARE
      drp binary_integer;
BEGIN
      drp :=DBMS_SPM.DROP_SQL_PLAN_BASELINE(
                  SQL_HANDLE => 'SYS_SQL_f9ec8ef45d ',
                  PLAN_NAME => 'SYS_SQL_PLAN_5d47d3d ');
END;
/
```

Displaying SQL Plan Baselines

Check the SQL is available in SQL PLAN Baseline

```
SQL> SELECT plan_name, sql_text,  sql_handle,  optimizer_cost,
        enabled, fixed, accepted
        FROM  DBA_SQL_PLAN_BASELINES
        WHERE sql_text LIKE '%TESTING%'
```

PLAN_NAME	SQL_TEXT	SQL_HANDLE	OPTIMIZER_COST	ENA	FIX	ACC
SYS_SQL_d0848cb7105fffdf	select sal TESTING from emp;	SYS_SQL_f9ec8ef45d	2	YES	NO	YES
SYS_SQL_d0848cb7105fffdf	select sal TESTING from emp;	SYS_SQL_f8de7fr6dd	1	YES	NO	No

```
SQL> SELECT  * FROM  TABLE(
       DBMS_XPLAN.DISPLAY_SQL_PLAN_BASELINE(
       sql_handle=> ' SYS_SQL_d0848cb7105fffdf ',format=>'basic'));
```

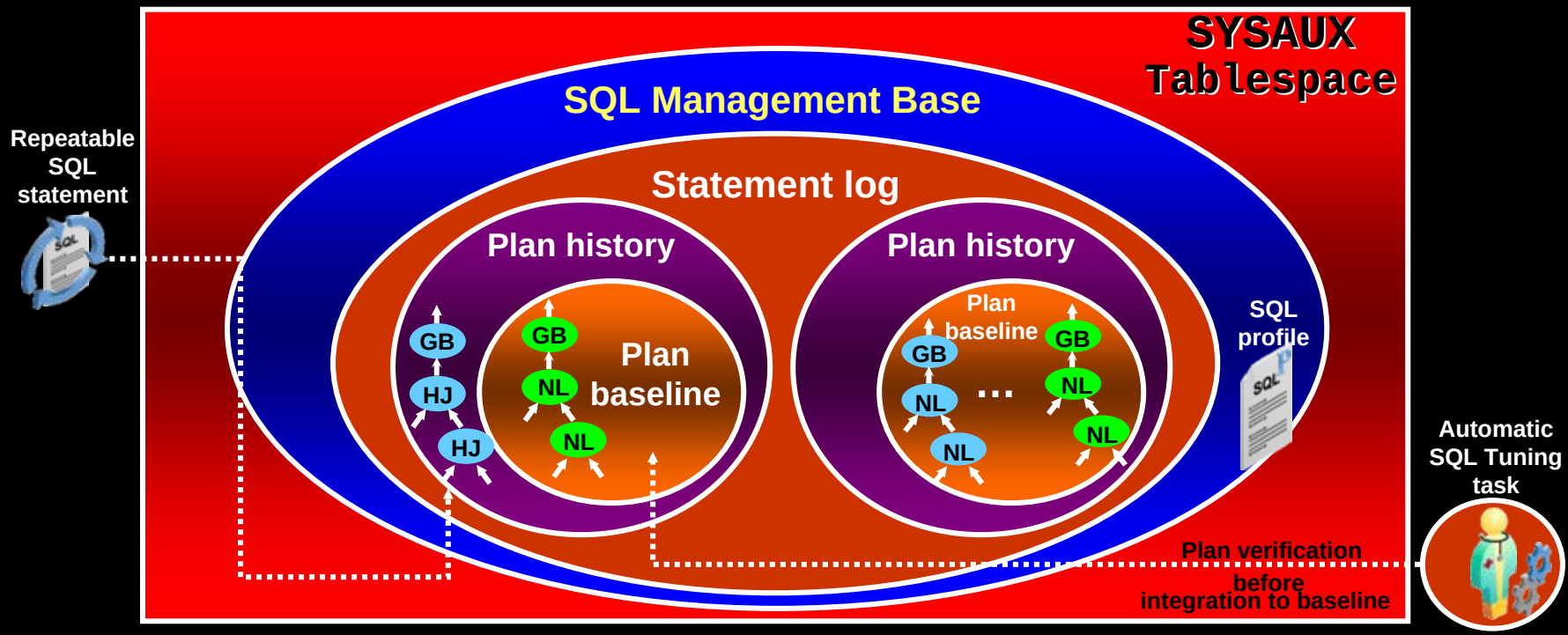
Disaplaying SQL Plan Baselines

```
PLAN_TABLE_OUTPUT
--------------------------------------------------------------------------------
SQL handle: SYS_SQL_c45581c7730096e3
SQL text: select /* INDYTEST */ * from emp where job='DBA' and rownum < 50
--------------------------------------------------------------------------------
Plan name: SYS_SQL_PLAN_730096e31877483c
Enabled: YES     Fixed: NO     Accepted: NO     Origin: AUTO-CAPTURE
--------------------------------------------------------------------------------

Plan hash value: 4187151245
-----------------------------------------
| Id  | Operation                  | Name   |
-----------------------------------------
|   0 | SELECT STATEMENT           |        |
|   1 |  COUNT STOPKEY             |        |
|   2 |   TABLE ACCESS BY INDEX ROWID| EMP    |
|   3 |    INDEX RANGE SCAN        | EMPJOB |
-----------------------------------------
Plan name: SYS_SQL_PLAN_730096e38694f11d
Enabled: YES     Fixed: NO     Accepted: YES     Origin: AUTO-CAPTURE
--------------------------------------------------------------------------------

Plan hash value: 1973284518
----------------------------------
| Id  | Operation         | Name |
----------------------------------
|   0 | SELECT STATEMENT  |      |
|   1 |  COUNT STOPKEY    |      |
|   2 |   TABLE ACCESS FULL| EMP  |
----------------------------------
36 rows selected.
```

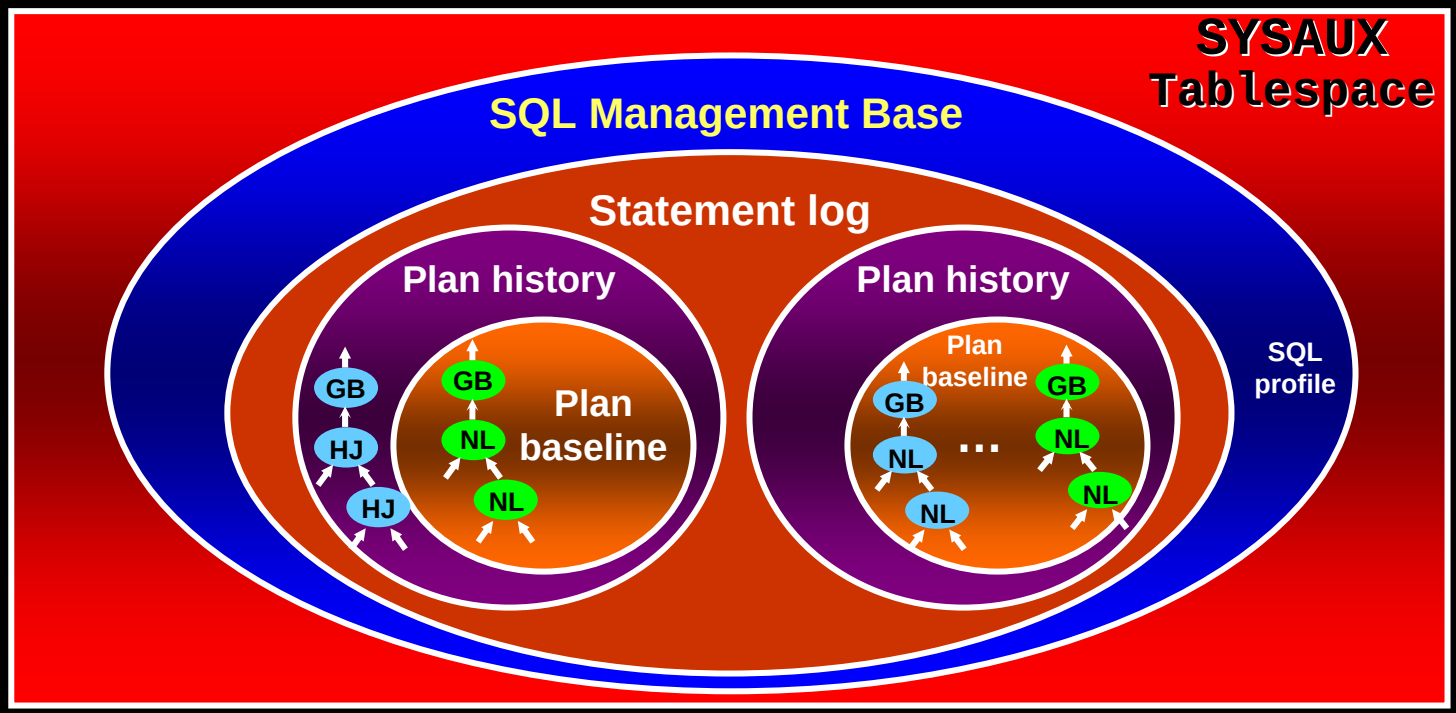
SQL Management Base



SQL Management Base

Three Important maintenance

- Disk Space Usage
- Purging Policy

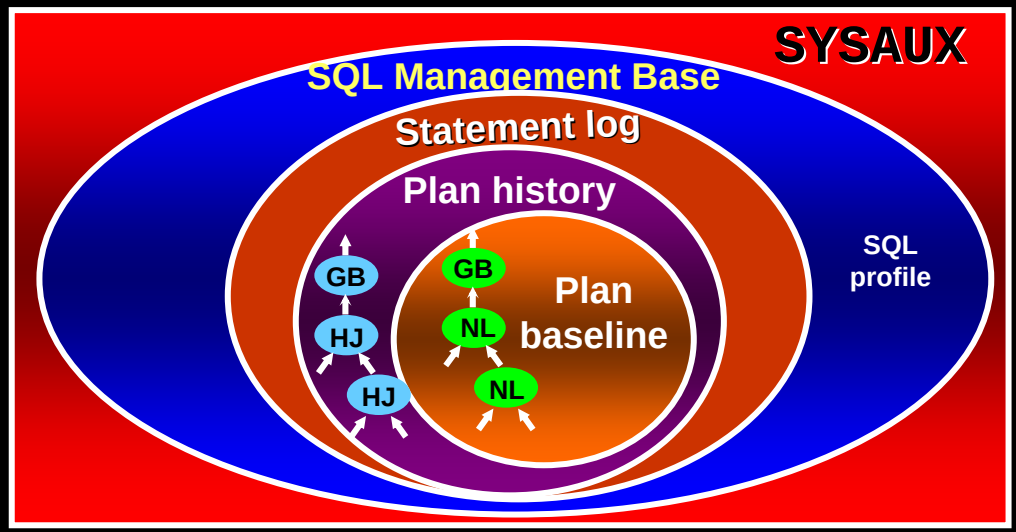


SQL Management Base

DISK Space Usage for Sql Management Base

- By default SMB space limit is 10% of the size of SYSAUX
- Allowable Limit is 1% - 50%
- Warning issued in Alert log by a weekly Background process
- Warning will continue every week until
  - ➢SMB space limit is increased

    DBMS_SPM.CONFIGURE('space_budget_percent',30);
  - ➢Size of SYSAUX is increased
  - ➢Space used by SMB is decreased by purging SQL baseline/Profiles



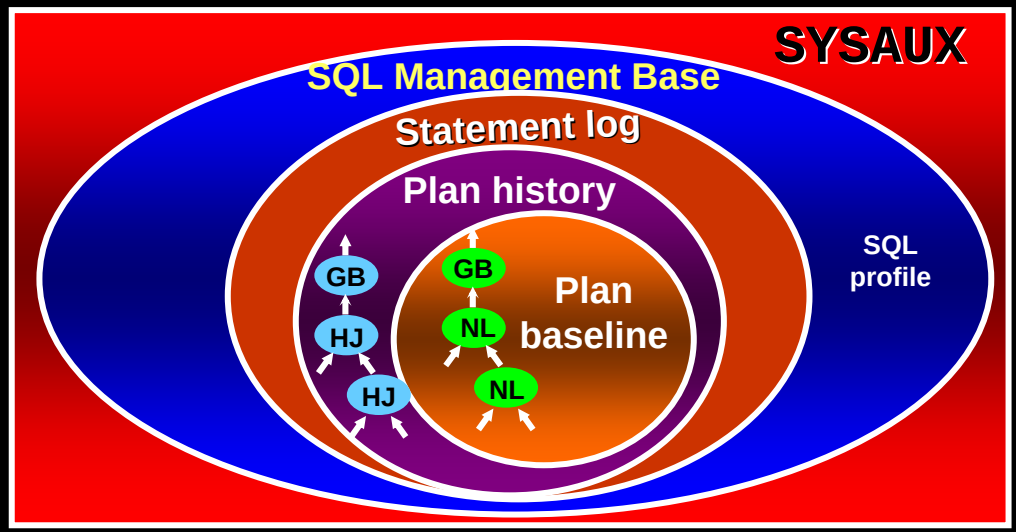
SQL Management Base

SMB Purging Policy

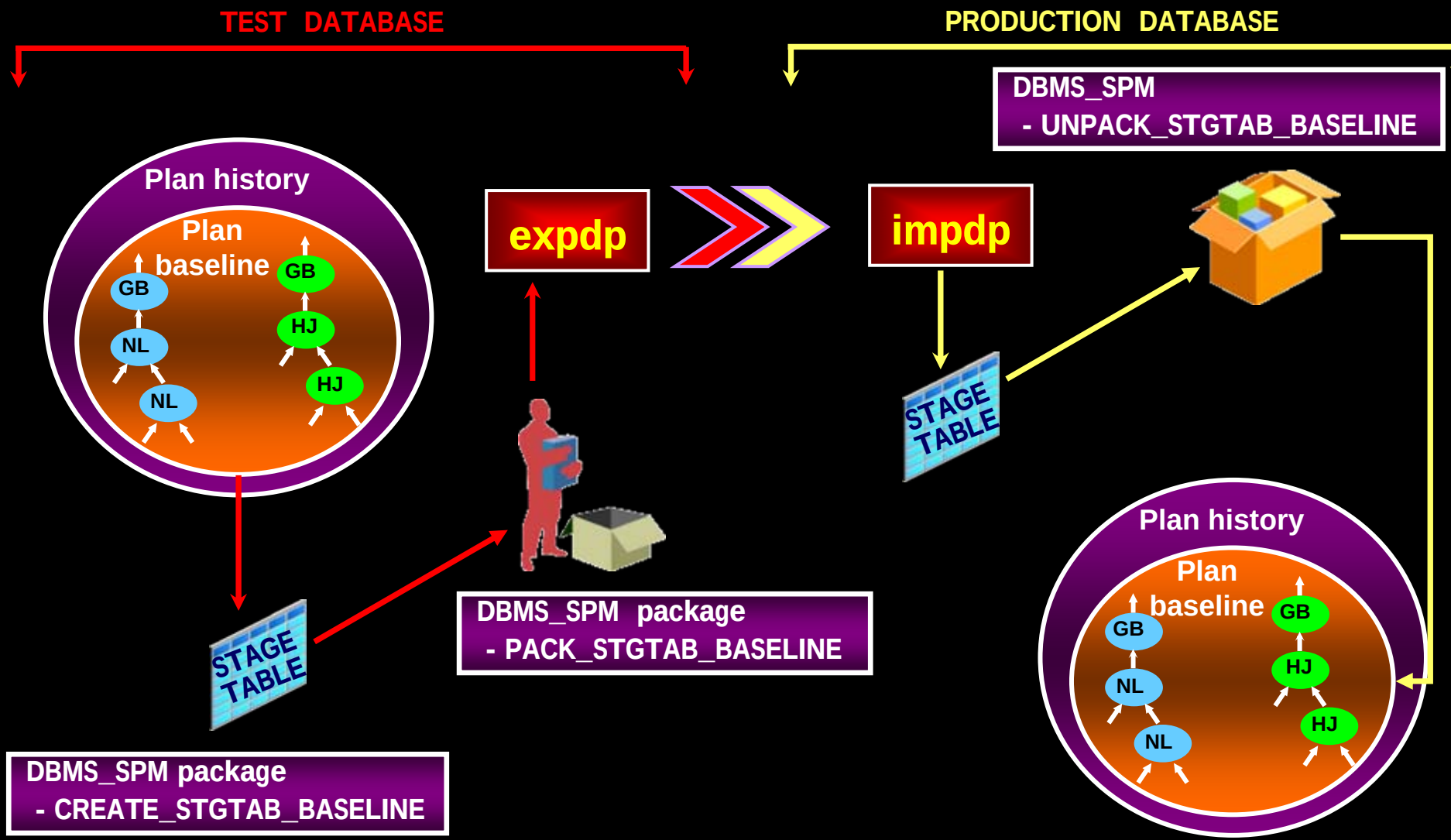
- By default , any PLAN not used for 53 Weeks are purged
- 53 Weeks ensures that Plan is available for any yearly SQL job
- SMB retention range is 5 weeks – 523 weeks [little more than 10yr]
- SMB Retention period can be configured [50 weeks]

```
DBMS_SPM.CONFIGURE( 'plan_retention_weeks',50);
```

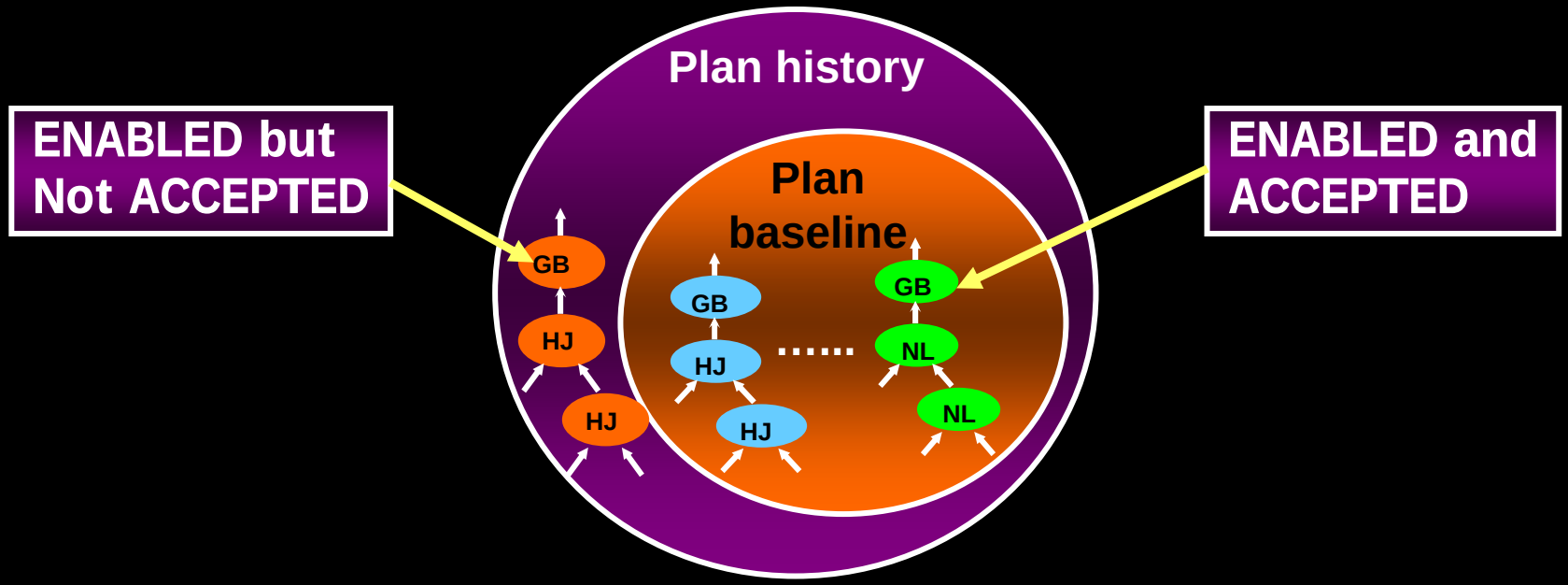
```
select parameter_name, parameter_value
from dba_sql_management_config;
```



Exporting/Importing SQL Baselines



SQL Plan Baseline Main Attributes



DBA_SQL_PLAN_BASELINES

Main Column Name

SQL_TEXT	→ Actual SQL Text
ORIGIN	→ MANUAL-LOAD | AUTO-CAPTURE | MANUAL-SQLTUNE | AUTO-SQLTUNE
ENABLED	→ YES | NO
ACCEPTED	→ YES | NO
FIXED	→ YES | NO
AUTOPURGE	→ YES | NO
OPTIMIZER_COST	

DBA_SQL_PLAN_BASELINE view

Check the SQL is available in SQL PLAN Baseline

```
SQL> SELECT plan_name, sql_text,  sql_handle,  optimizer_cost,
        enabled, fixed, accepted
        FROM  DBA_SQL_PLAN_BASELINES
        WHERE sql_text LIKE '%TESTING%'
```

PLAN_NAME	SQL_TEXT	SQL_HANDLE	ENABLED	ACCEPTED	FIXED
SYS_SQL_d0848cb7105fffdf	select sal TESTING from emp;	SYS_SQL_f9ec8ef45d	YES	NO	YES
SYS_SQL_d0848cb7105fffdf	select sal TESTING from emp;	SYS_SQL_f8de7fr6dd	YES	NO	NO

```
DECLARE
      v_chg              Natural;
BEGIN
      v_chg : dbms_spm.alter_sql_plan_baseline( sql_handle=>'SYS_SQL_f8de7fr6d ', -
                  plan_name=>'SYS_SQL_d0848cb7105fffdf ', -
                  attribute_name => 'AUTOPURGE | ENABLED | ACCEPTED | FIXED', -
                  attribute_value => 'YES | NO');
      dbms_output.put_line(i);
END;
```

Any One Attribute Name or Value


QUESTIONS
&
ANSWERS

THANKS

indy.johal@datasoftech.com