12

SQL Plan Management

Objectives

After completing this lesson, you should be able to:

- Manage SQL performance through changes
- Set up SQL Plan Management
- Set up various SQL Plan Management scenarios
- Load hinted plans into SQL Plan Management
- Migrate stored outlines to SQL plan baselines

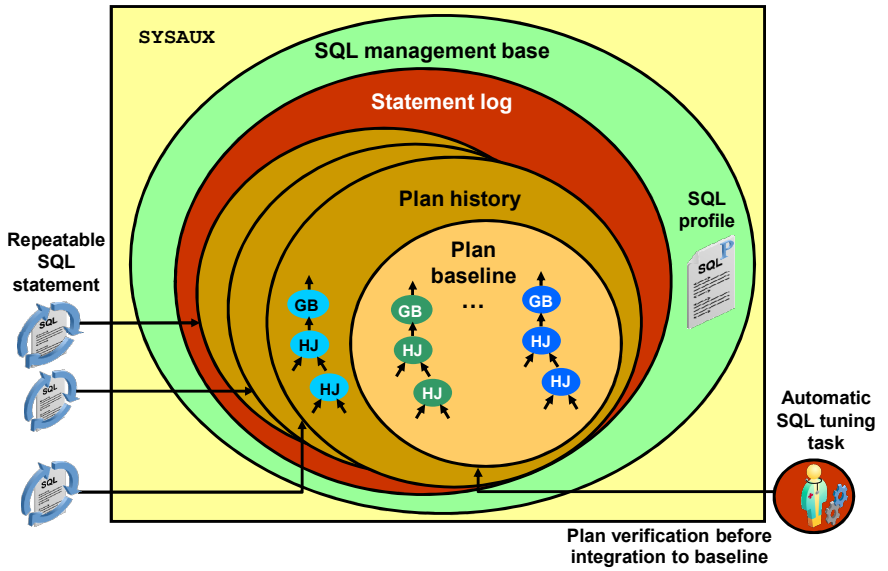
Maintaining SQL Performance

Maintaining performance may require using SQL plan baselines.

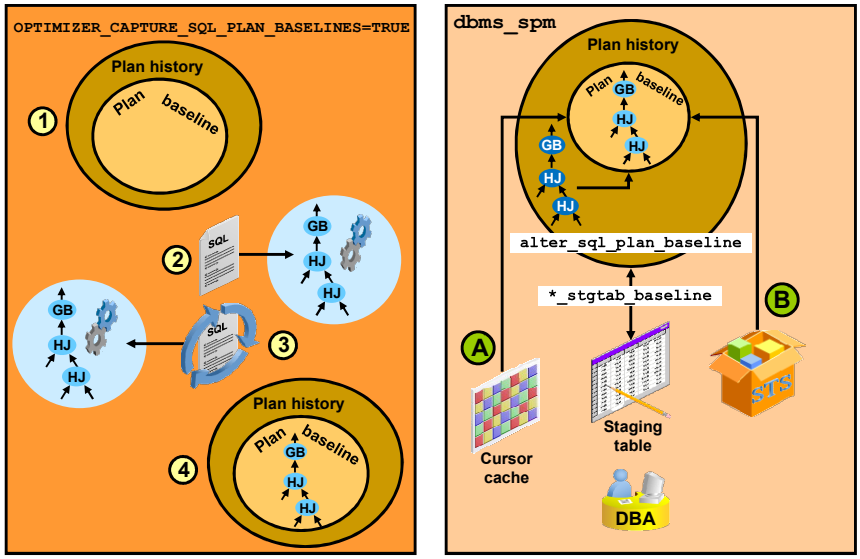
SQL Plan Management: Overview

- SQL Plan Management is automatically controlled SQL plan evolution.
- Optimizer automatically manages SQL plan baselines.
  - Only known and verified plans are used.
- Plan changes are automatically verified.
  - Only comparable or better plans are subsequently used.
- The plan baseline can be seeded for critical SQL with SQL tuning set (STS) from SQL Performance Analyzer.

SQL Plan Baseline: Architecture

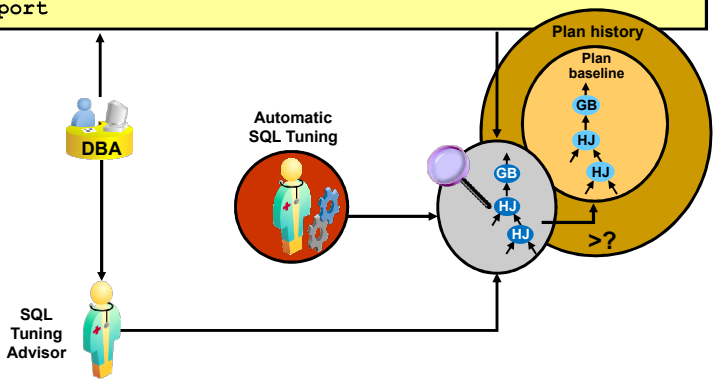


Loading SQL Plan Baselines

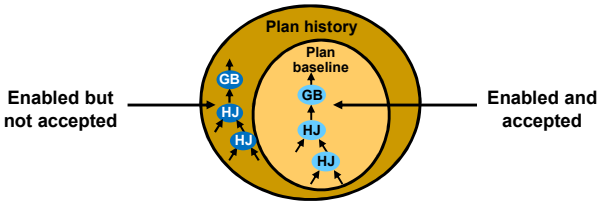


Evolving SQL Plan Baselines

```
variable report clob
exec :report:=DBMS_SPM.EVOLVE_SQL_PLAN_BASELINE(-
sql_handle=>'SYS_SQL_593bc74fca8e6738');
Print report
```



Important Baseline SQL Plan Attributes

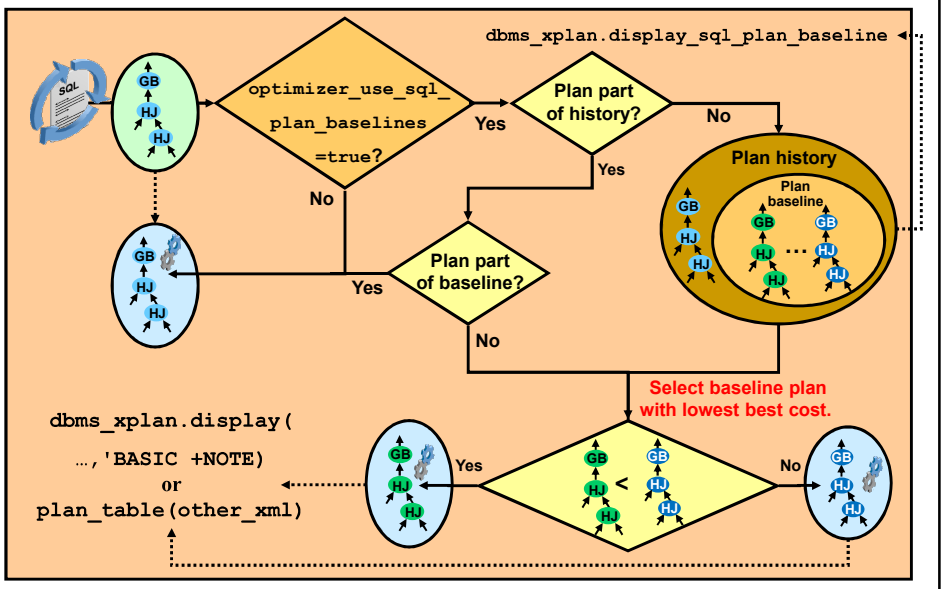



```
select signature, sql_handle, sql_text, plan_name, origin, enabled,
       accepted, fixed, autopurge
from dba_sql_plan_baselines;
```

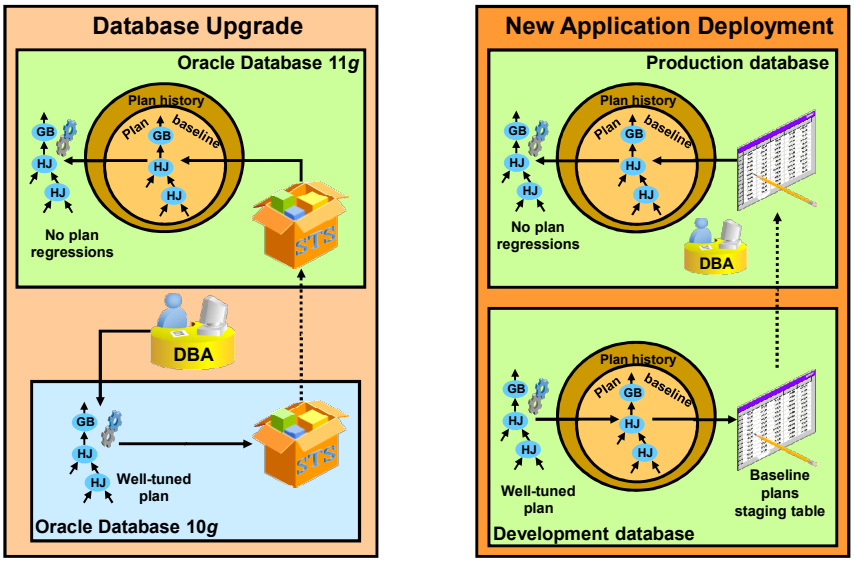
```
SIGNATURE  SQL_HANDLE  SQL_TEXT PLAN_NAME       ORIGIN       ENA ACC FIX AUT
---------- ----------- -------- ---------------- ------------ --- --- --- ---
8.062E+18 SYS_SQL_6fe2 select.. SQL_PLAN_6zsn…   AUTO-CAPTURE YES NO  NO  YES
8.062E+18 SYS_SQL_e23f select.. SQL_PLAN_f4gy…   AUTO-CAPTURE YES YES NO  YES
…
```

```
 exec :cnt := dbms_spm.alter_sql_plan_baseline(-
              sql_handle => 'SYS_SQL_6fe28d438dfc352f', -
              plan_name       => 'SQL_PLAN_6zsnd8f6zsd9g54bc8843',-
              attribute_name  => 'ENABLED', attribute_value => 'NO');
```

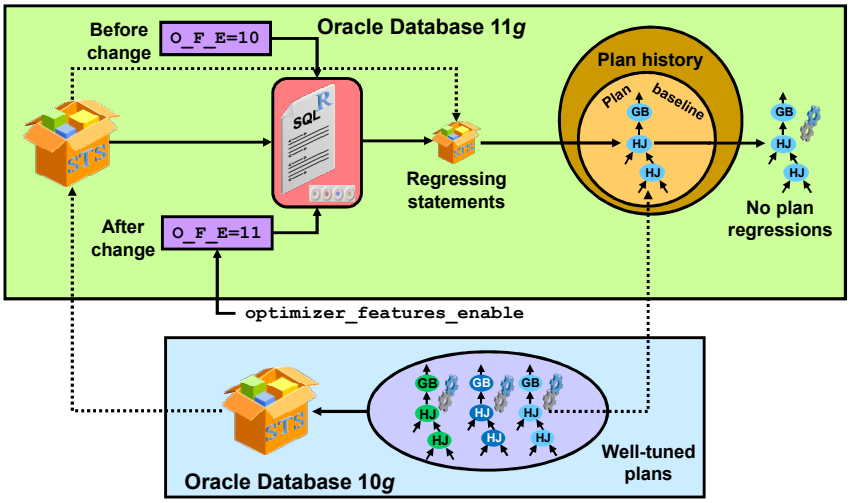
SQL Plan Selection



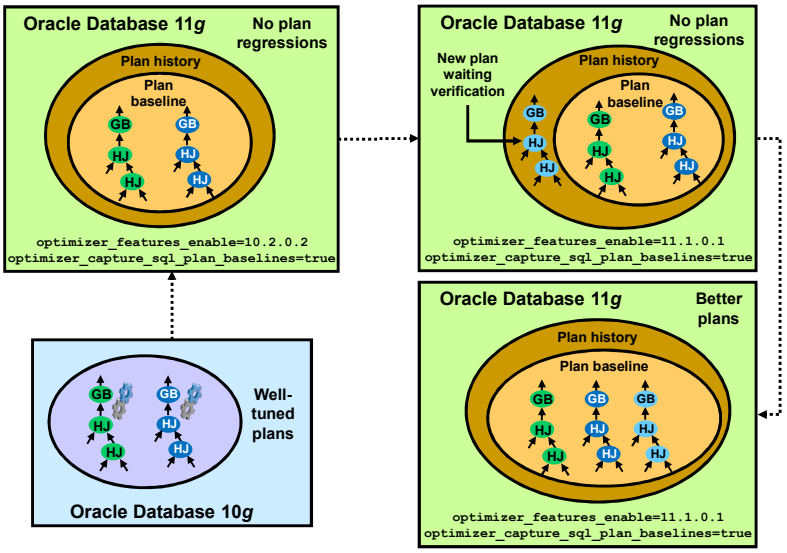
Possible SQL Plan Manageability Scenarios



SQL Performance Analyzer and SQL Plan Baseline Scenario



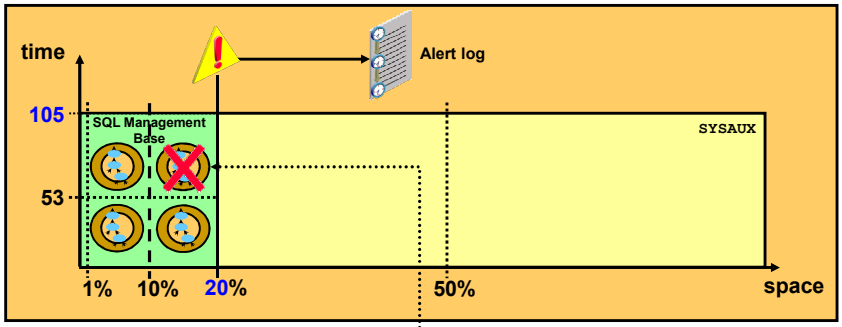
Loading a SQL Plan Baseline Automatically



Purging SQL Management Base Policy

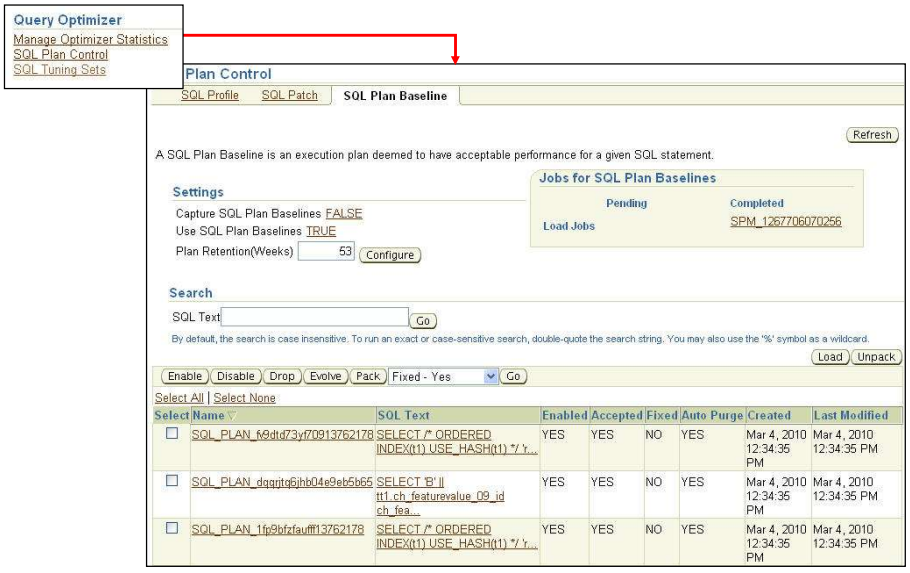
```
SQL> exec dbms_spm.configure('SPACE_BUDGET_PERCENT',20);
SQL> exec dbms_spm.configure('PLAN_RETENTION_WEEKS',105);
```

DBA_SQL_MANAGEMENT_CONFIG

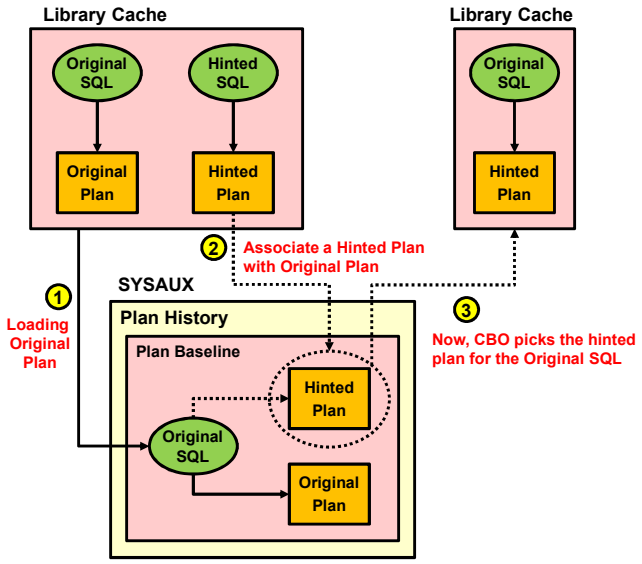



```
SQL> exec :cnt := dbms_spm.drop_sql_plan_baseline('SYS_SQL_37e0168b04e73efe');
```

Enterprise Manager and SQL Plan Baselines



Loading Hinted Plans into SPM: Example



Using the `MIGRATE_STORED_OUTLINE` Functions

- Specify stored outlines to be migrated based on outline name, SQL text, or outline category, or migrate all stored outlines in the system to SQL plan baselines:

```
DBMS_SPM.MIGRATE_STORED_OUTLINE (
attribute_name IN VARCHAR2,
attribute_value IN CLOB,
fixed IN VARCHAR2 := 'NO')
RETURN CLOB;
```

- Specify one or more stored outlines to be migrated:

```
DBMS_SPM.MIGRATE_STORED_OUTLINE (
outln_list IN DBMS_SPM.NAME_LIST,
fixed IN VARCHAR2 := 'NO')
RETURN CLOB;
```

Quiz

When the `OPTIMIZER_USE_SQL_PLAN_BASELINES` parameter is set to `TRUE`, the optimizer:

a. Does not develop an execution plan; it uses an accepted plan in the baselines
b. Compares the plan that it develops with accepted plans in the baselines
c. Compares the plan that it develops with enabled plans in the baselines
d. Does not develop an execution plan; it uses enabled plans in the baselines
e. Develops plans and adds them to the baselines as verified

Summary

In this lesson, you should have learned how to:

- Manage SQL performance through changes
- Set up SQL Plan Management
- Set up various SQL Plan Management scenarios
- Load hinted plans into SQL Plan Management
- Migrate Stored Outlines to SQL Plan Baselines

Practice 12: Overview
Using SQL Plan Management

This practice covers the use of SQL Plan Management.