Hvordan unngå overraskelser ved oppgradering av Oracle fra 10g til 11g!

Ingemar Jansson Haverstad

OUGN

14. april 2010

Version 0.85

Om foredragsholderen

Ingemar Jansson Haverstad
dbWatch AS

- Arbeidet med Oracle siden 1985...
  ... med Unix siden 1979

- Konsulent, instruktør og foredragsholder

ingemar@oraklet.no
www.oraklet.no/foredrag

ORACLE
Certified Professional

Agenda

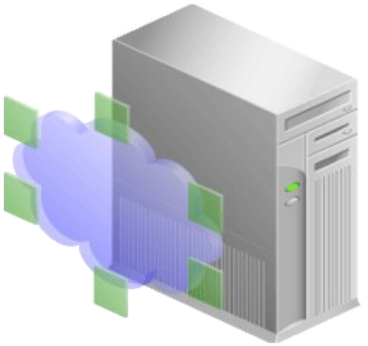
- Først litt bakgrunn.
- Testing ved hjelp av Swingbench.
- De små detaljene...
- Outlines og Baselines.

Presentasjonen baserer seg på «White Paper» fra *Oracle* av Maria Colgan, utgitt i november 2009:

«*Upgrading from Oracle Database 10g to 11g: What to expect from the Optimizer*»

Bakgrunn

- Datavarehus i Skatteetaten
  - Oppstart i 2003
- IBM/AIX Unix server
  - 16 CPU
  - 88 GB internminne

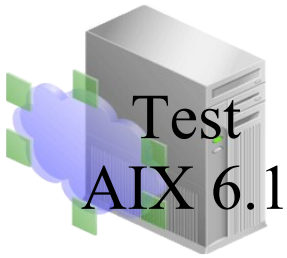


- Oracle10g R2 (10.2.0.4)
- Største tabellen 8.500.000.000 rader
- Temporær tabellplass 1-3 TB (dagsform)

Miljøer


Produksjon
AIX 5.3


DVH
Mart
OWB
Apps
Workflow


Test
AIX 6.1


DVH
Mart
OWB
Apps
Workflow


Utvikling
AIX 5.3


DVH
Mart
OWB
Apps
Workflow


Referanse
AIX 6.1


DVH
Mart
OWB
Apps
Workflow

Utfordringene...

- Oracle support status
- Statistikk på avveier...
- Parallellisering `/*+ Hinting, hinting */`
- Oracle Warehouse Builder 11.1
  - ny metadata arkitektur
- Oracle Warehouse Builder 11.2
  - ny klient design (GUI)
- Oracle Warehouse Builder (OWB)
- Oracle Data Integrator (ODI)

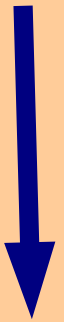
To oppgraderings veier

- *Database 10.2*
- *Warehouse Builder 10.2*

- *Database 11.1*
- **Beholde** *Warehouse Builder 10.2*

- *Database 11.2*
- *Warehouse Builder 11.2*

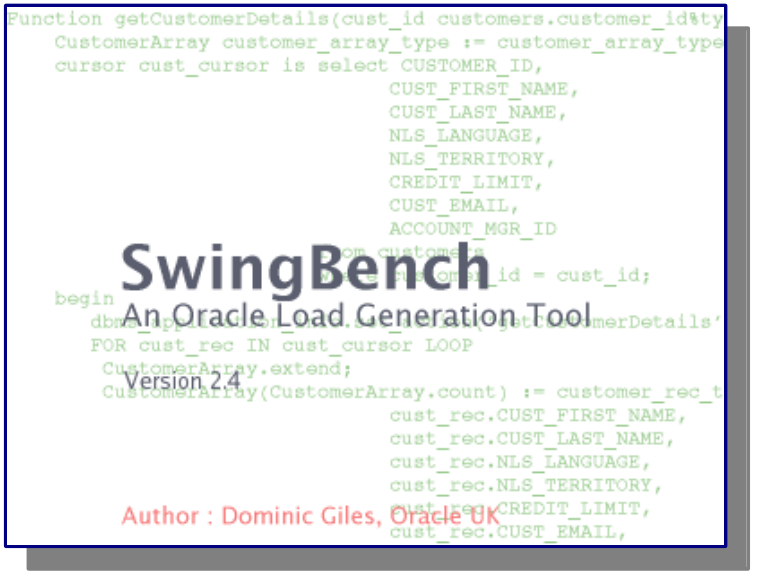
- *Database 10.2*
- *Warehouse Builder 10.2*



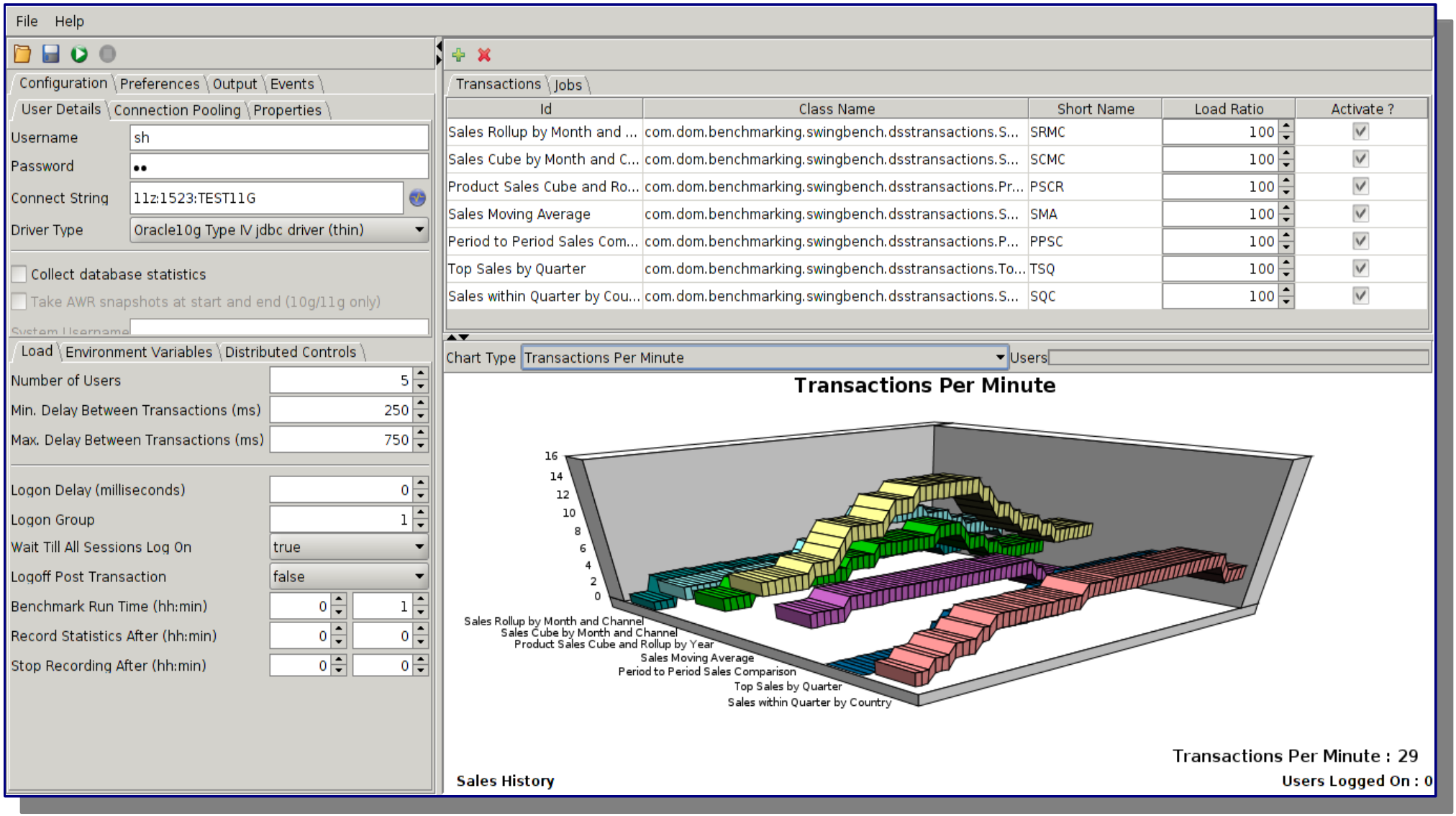
- *Database 11.2*
- *Warehouse Builder 11.2*

Testmiljø, bruk Swingbench

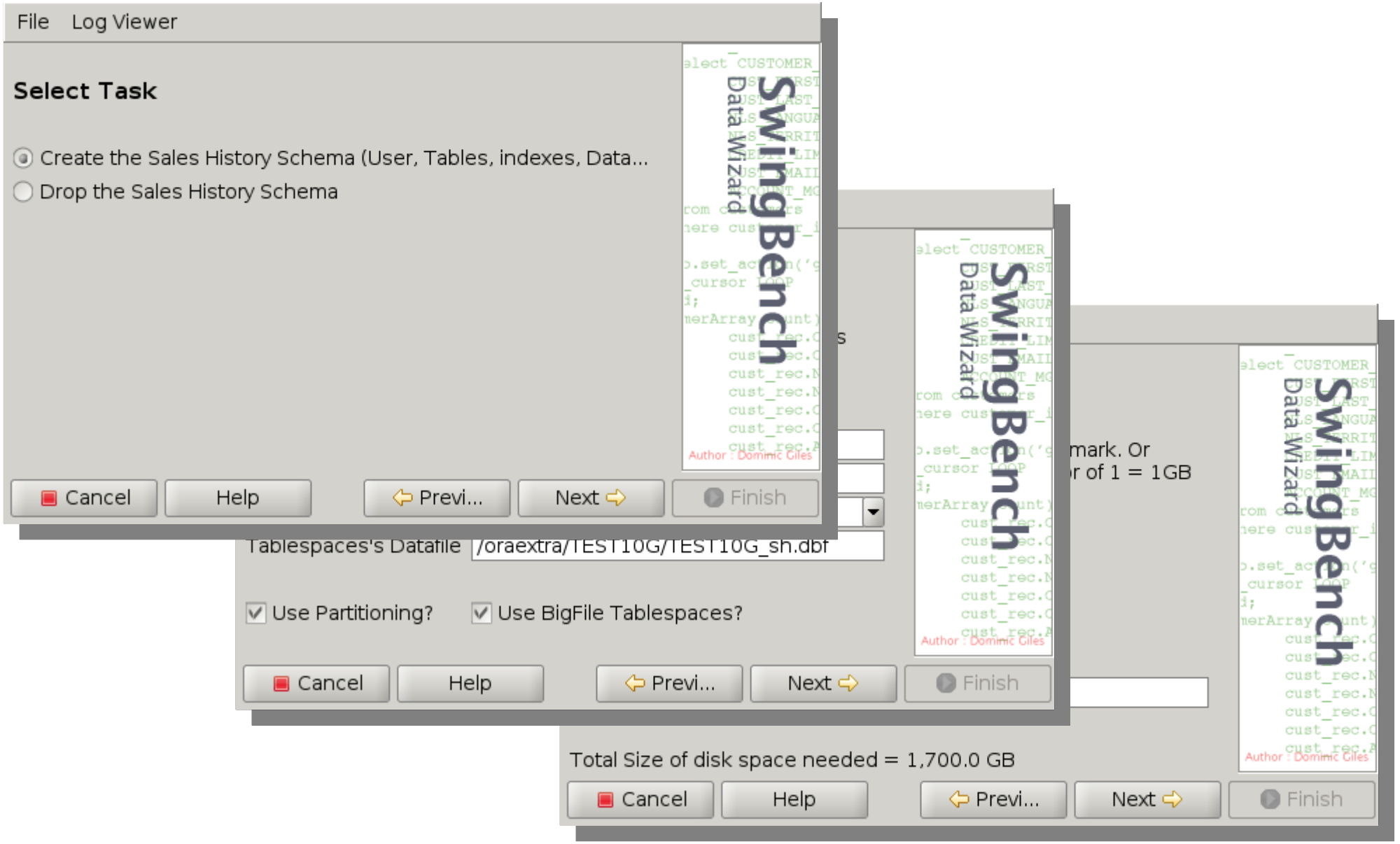
- Skrevet i Java av Dominic Giles, Oracle UK
- www.dominicgiles.com
- Støtte for OLTP – Order Entry (OE)
- Støtte for DSS/DW – Sales History (SH)



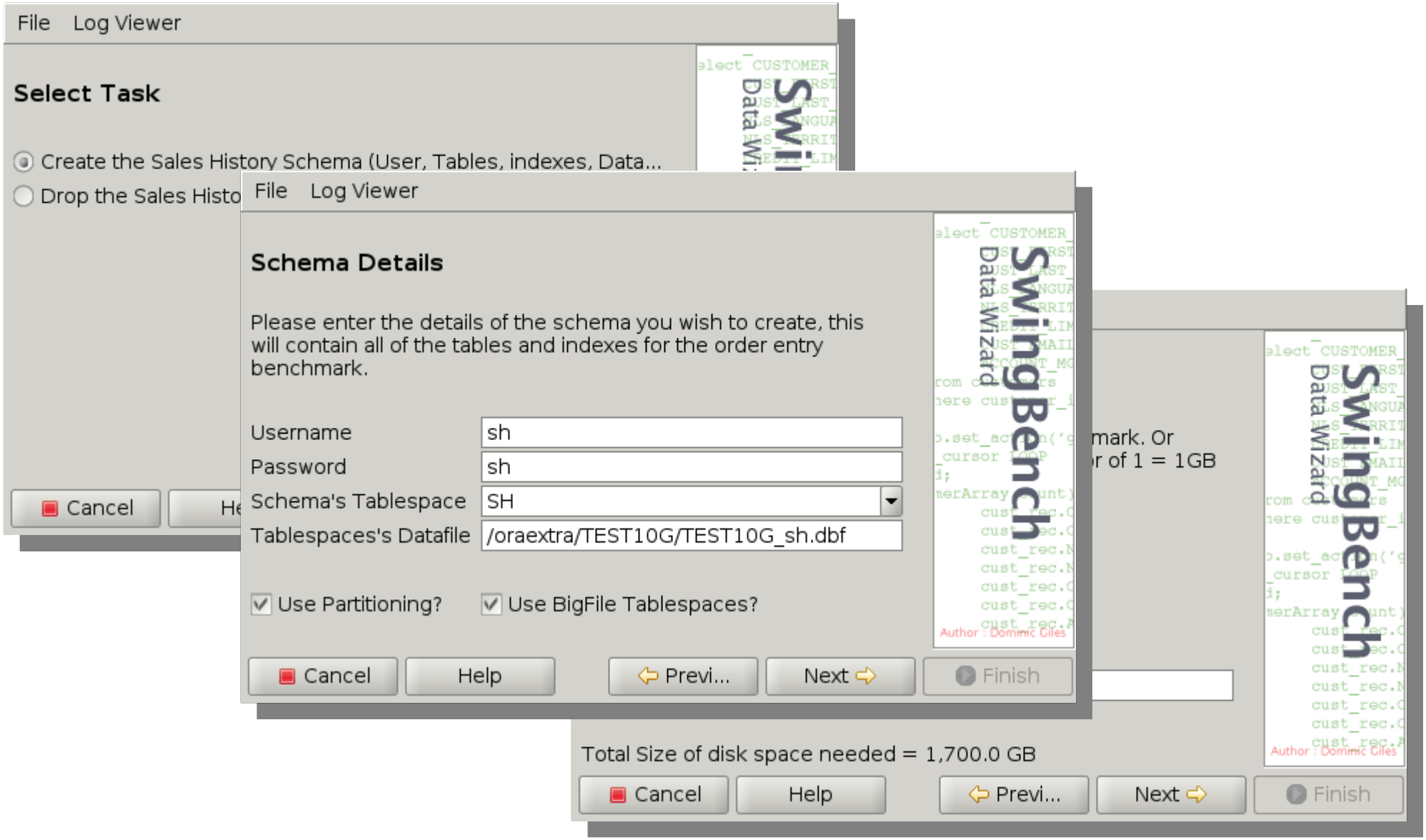
Swingbench


File
Help
Configuration
Preferences
Output
Events
User Details
Connection Pooling
Properties
Username
sh
Password
Connect String
11z:1523:TEST11G
Driver Type
Oracle10g Type IV jdbc driver (thin)
Collect database statistics
Take AWR snapshots at start and end (10g/11g only)
Load
Environment Variables
Distributed Controls
Number of Users
5
Min. Delay Between Transactions (ms)
250
Max. Delay Between Transactions (ms)
750
Logon Delay (milliseconds)
0
Logon Group
1
Wait Till All Sessions Log On
true
Logoff Post Transaction
false
Benchmark Run Time (hh:min)
Record Statistics After (hh:min)
Stop Recording After (hh:min)
Transactions
Jobs
Id
Class Name
Short Name
Load Ratio
Activate ?
Sales Rollup by Month and ...
com.dom.benchmarking.swingbench.dsstransactions.S...
SRMC
Sales Cube by Month and C...
SCMC
Product Sales Cube and Ro...
com.dom.benchmarking.swingbench.dsstransactions.Pr...
PSCR
Sales Moving Average
SMA
Period to Period Sales Com...
PPSC
Top Sales by Quarter
TSQ
Sales within Quarter by Cou...
SQC
Chart Type
Transactions Per Minute
Users
Transactions Per Minute
16
14
12
10
8
6
4
2
0
Sales Rollup by Month and Channel
Sales Cube by Month and Channel
Product Sales Cube and Rollup by Year
Sales Moving Average
Period to Period Sales Comparison
Top Sales by Quarter
Sales within Quarter by Country
Transactions Per Minute : 29
Sales History
Users Logged On : 0

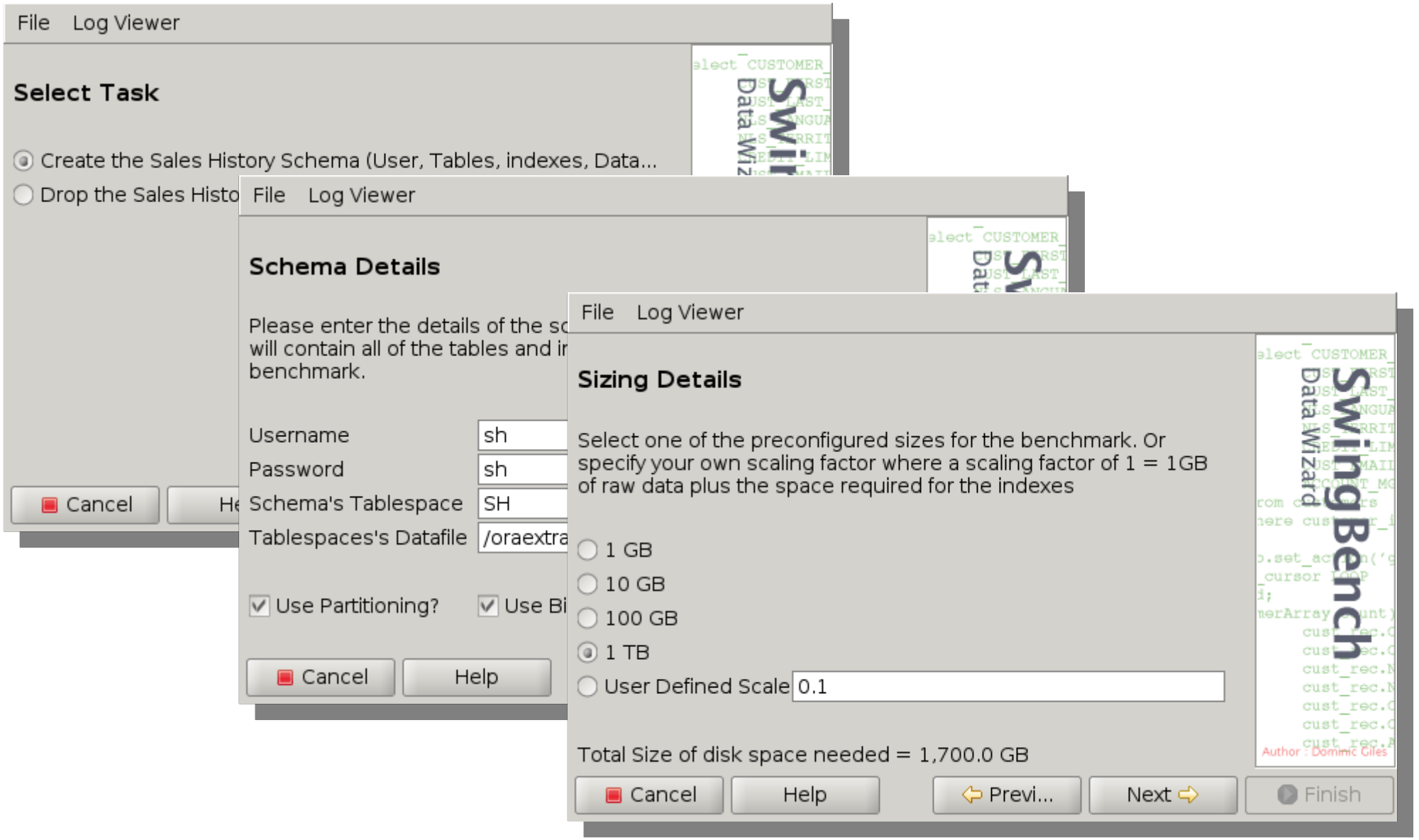
Swingbench datageneratorOpprett skjema



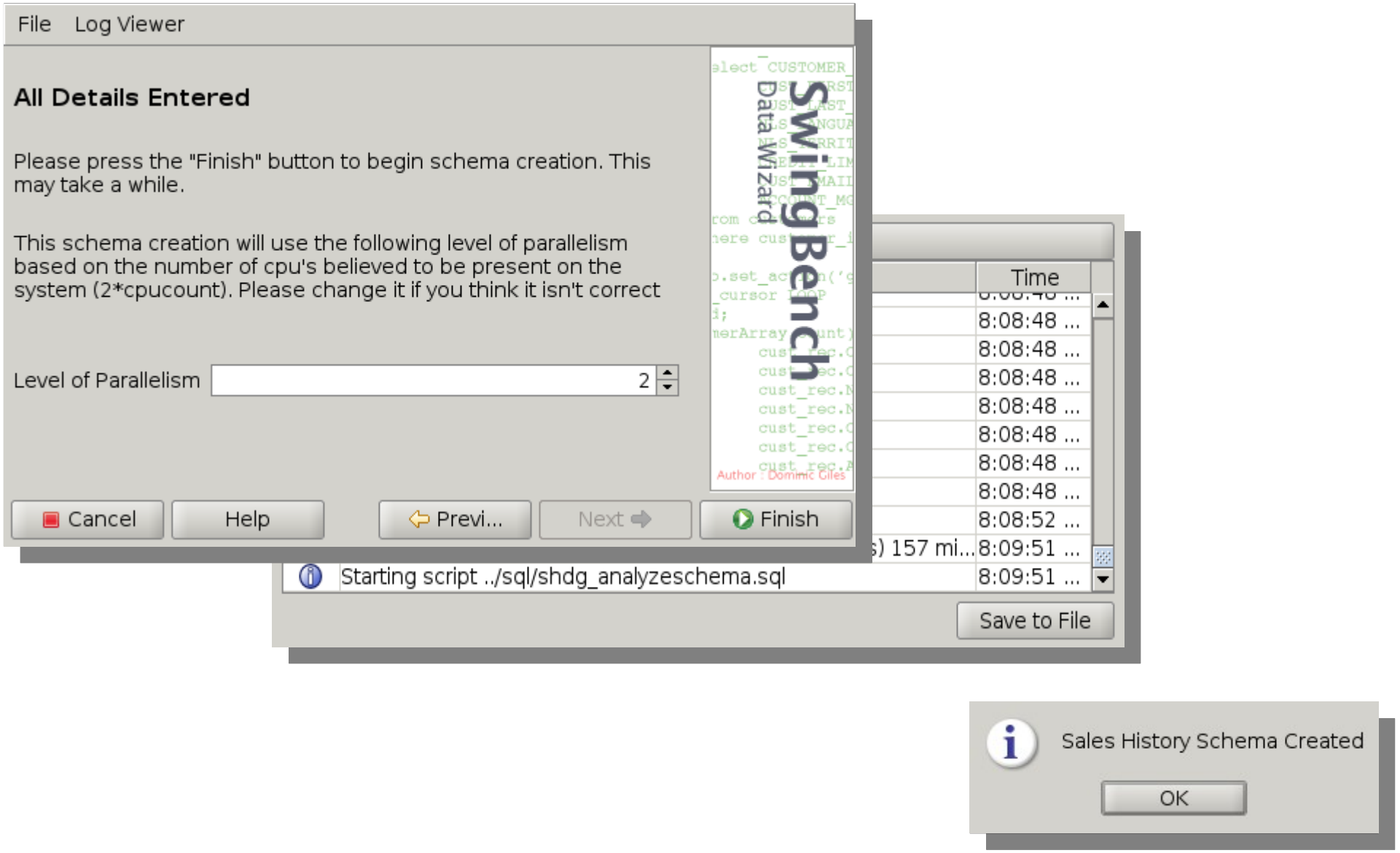
Swingbench datageneratorSkjema navn...



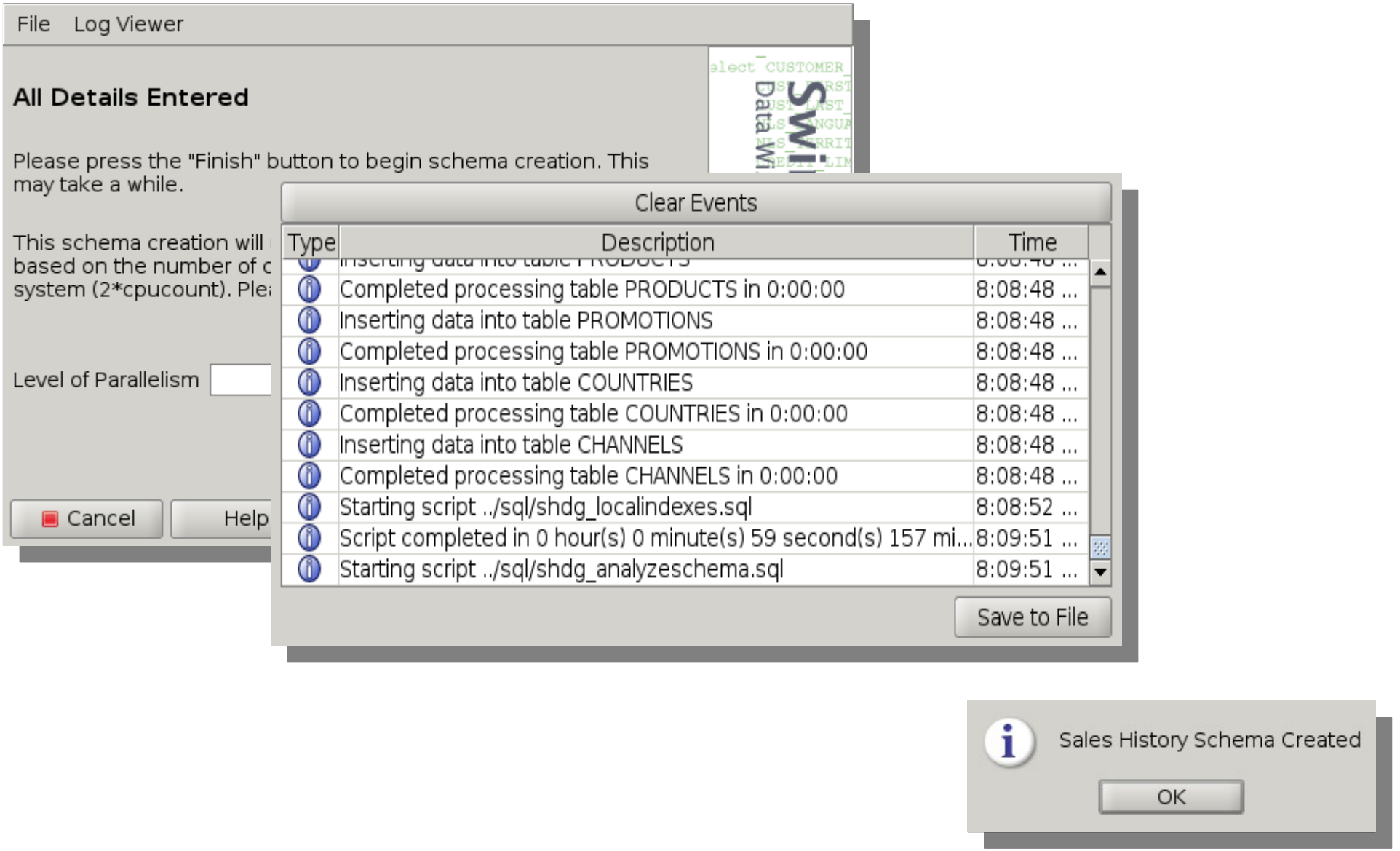
Swingbench datageneratorStørrelse



Swingbench datageneratorParalellisering



Swingbench datagenerator Oppsummering



10g: miljøvariabler i Unix/Linux bash

```
oracle@11z:~ $ cat .test10g

export ORACLE_BASE=/oracle
export ORACLE_HOME=${ORACLE_BASE}/product/10.2.0.4/db
export ORACLE_SID=TEST10G

export PATH=...:${ORACLE_HOME}/bin:${ORACLE_HOME}/OPatch
export CDPATH=.:${ORACLE_HOME}:${ORACLE_BASE}/admin/${ORACLE_SID}

PS1='\u@\h:\W \[\033[31m\][${ORACLE_SID}]\[\033[00m\] $ '

unset LANG

oracle@11z:~ $ . .test10g

oracle@11z:~ [TEST10G] $ cd udump
/oracle/admin/TEST10G/udump

oracle@11z:udump [TEST10G] $
```

11g: miljøvariabler i Unix/Linux bash

```
oracle@11z:~ $ cat .test11g

export ORACLE_BASE=/oracle
export ORACLE_DIAG=${ORACLE_BASE}/diag
export ORACLE_HOME=${ORACLE_BASE}/product/11.2.0.1/db
export ORACLE_SID=TEST11G
export ORACLE_UNQNAME=${ORACLE_SID}

export PATH=...:${ORACLE_HOME}/bin
export CDPATH=.:${ORACLE_HOME}:${ORACLE_BASE}/admin/${ORACLE_SID}:\
$ORACLE_DIAG/rdbms/${ORACLE_DIAG}/rdbms/$(echo $ORACLE_DB | \
   tr '[:upper:]' '[:lower:]')/$(echo $ORACLE_SID | \
   tr '[:lower:]' '[:upper:]')

PS1='\u@\h:\W \[\033[31m\][${ORACLE_SID}]\[\033[00m\] $ '


oracle@11z:~ $ . .test11g

oracle@11z:~ [TEST11G] $ cd trace
/oracle/diag/rdbms/test11g/TEST11G/trace

oracle@11z:trace [TEST11G] $
```

10g: SQL-prompt

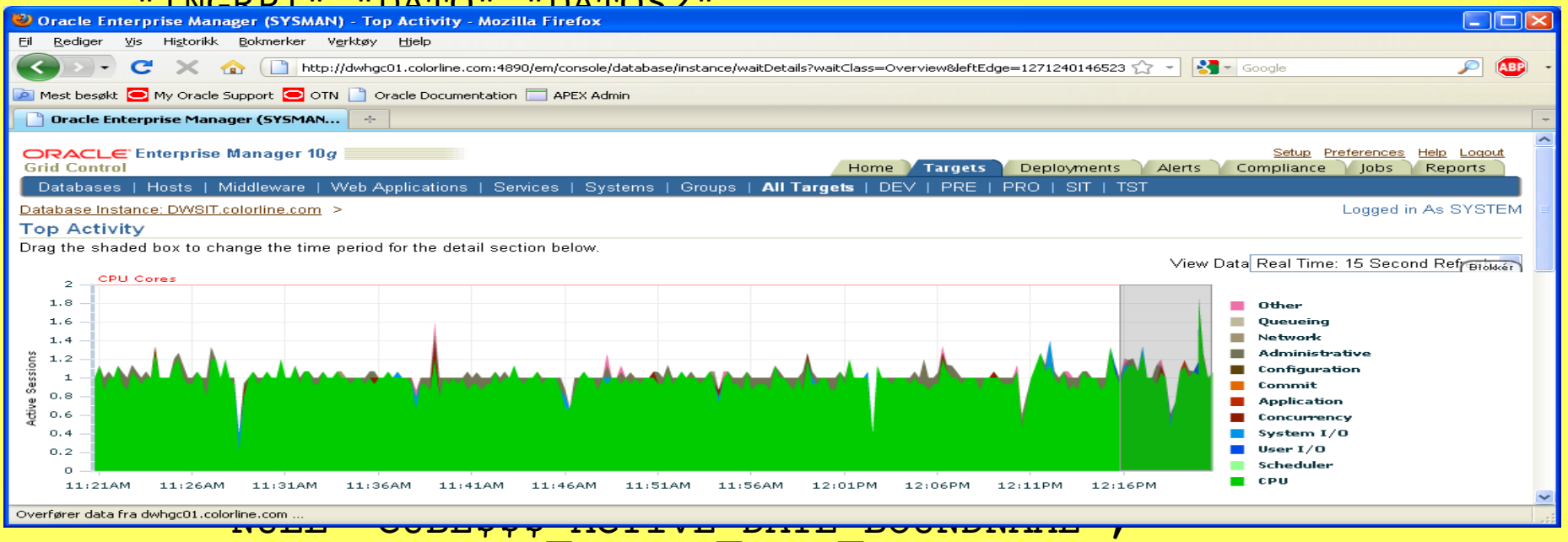
```
oracle@11z:trace [TEST10G] $ cd sqlplus/admin
/oracle/product/10.2.0.4/db/sqlplus/admin

oracle@11z:trace [TEST10G] $ vi glogin.sql
...
--- Oppdatert av Ingemar, 9. april 2010
SET PAGES 99
SET LINES 135

COLUMN file_name FORMAT a65

SET sqlprompt '10gR2 _USER"@"_CONNECT_IDENTIFIER SQL> '

oracle@11z:admin [TEST10G] $ sqlplus / as sysdba

SQL*Plus: Release 10.2.0.4.0 - Fri Apr 9 10:10:51 2010

Connected to:
Oracle Database 10g Enterprise Edition Release 10.2.0.4.0 – 64bit
With the Partitioning, OLAP and Data Mining

10gR2 SYS@TEST10G SQL>
```

11g: SQL-prompt

```
oracle@11z:trace [TEST11G] $ cd sqlplus/admin
/oracle/product/10.2.0.4/db/sqlplus/admin

oracle@11z:trace [TEST11G] $ vi glogin.sql
...
--- Oppdatert av Ingemar, 9. april 2010
SET PAGES 99
SET LINES 135

COLUMN file_name FORMAT a65

SET sqlprompt '11gR2 _USER"@"_CONNECT_IDENTIFIER SQL> '

oracle@11z:admin [TEST11G] $ sqlplus / as sysdba

SQL*Plus: Release 11.2.0.1.0 Fri Apr 9 10:16:44 2010

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit
With the Partitioning, OLAP and Data Mining options

11gR2 SYS@TEST11G SQL>
```

OWB 11g: Innledende testing

```
MERGE
/*+ APPEND PARALLEL("POS_FACT_BUDOMSETNING") */
INTO "POS_FACT_BUDOMSETNING" USING
(
  SELECT
      "INGRP1"."BUD_SALG" "BUD_SALG",
      "INGRP1"."DATO" "DATO$2"
```




```
      ...
```

- OWB 11g legger ut SQL-kode som kan være ulik OWB 10g

OWB 10g: ??!

```
MERGE
/*+ APPEND PARALLEL("POS_FACT_BUDOMSETNING") */
INTO "POS_FACT_BUDOMSETNING" USING
(
  SELECT
  /*+ ORDERED NO_MERGE("INGRP1") NO_MERGE */
      "INGRP1"."BUD_SALG" "BUD_SALG",
       "INGRP1"."DATO" "DATO$2",
       "INGRP1"."BUD_VAREKOST" "BUD_VAREKOST$2",
       "DIM_TID"."DAG_ID" "DAG_ID",
       "DIM_DRIFTSSTED"."DRIFTSSTED_ID" "DRIFTSSTED_ID",
       "DIM_ARTIKKEL"."HOVEDGRUPPE_ID" "HOVEDGRUPPE_ID"
   FROM
     (SELECT
      /*+ NO_MERGE */
          "AGG_BUDSJETT$2"."BUD_OMSETNING$2" "BUD_SALG",
           NVL("KEY_LOOKUP_TID"."DA_ACTUAL_DATE", NULL ) "DATO",
           "AGG_BUDSJETT$2"."BUD_VAREKOST$3" "BUD_VAREKOST",
           NULL "CUBE$$$_ACTIVE_DATE_BOUNDNAME",
           ...
```

- Er dette hintet generert av OWB 10g?

10g: Optimaliseringsparametere

```
10gR2 SYS@TEST10G SQL> SHOW PARAMETERS optimizer

NAME                                 TYPE        VALUE
------------------------------------ ----------- ----------
optimizer_dynamic_sampling           integer     2
optimizer_features_enable            string      10.2.0.4
optimizer_index_caching              integer     0
optimizer_index_cost_adj             integer     100
optimizer_mode                       string      ALL_ROWS
optimizer_secure_view_merging        boolean     TRUE
```

11g: Optimaliseringsparametere

```
11gR2 SYS@TEST11G SQL> SHOW PARAMETERS optimizer

NAME                                 TYPE        VALUE
------------------------------------ ----------- ----------
optimizer_capture_sql_plan_baselines boolean     FALSE
optimizer_dynamic_sampling           integer     2
optimizer_features_enable            string      11.2.0.1
optimizer_index_caching              integer     0
optimizer_index_cost_adj             integer     100
optimizer_mode                       string      ALL_ROWS
optimizer_secure_view_merging        boolean     TRUE
optimizer_use_invisible_indexes      boolean     FALSE
optimizer_use_pending_statistics     boolean     FALSE
optimizer_use_sql_plan_baselines     boolean     TRUE
```

- Ikke dokumentert parameter

```
_optimizer_ignore_hints              boolean     FALSE
```

Nye muligheter i 11g

- `OPTIMIZER_USE_INVISIBLE_INDEXES = FALSE`
  - DBA kan vurdere om indekser skal brukes i produksjon.
- `OPTIMIZER_USE_PENDING_STATISTICS = FALSE`
  - Drøye med å «publisere» statistikk.
- `OPTIMIZER_CAPTURE_SQL_PLAN_BASELINES = FALSE`
  - SQL Plan Management (SPM), hente inn informasjon om SQLer som kjøres for videre evaluering.
- `OPTIMIZER_USE_SQL_PLAN_BASELINES = TRUE`
  - Gir oss mulighet til å kun benytte verifiserte SQLer.

Statistikk i 11g

- Bedre kontroll over hvert individuelt objekt.
- Nye prosedyrer i DBMS_STATS pakken:
  - `DBMS_STATS.SET_TABLE_PREFS`
  - `DBMS_STATS.SET_SCHEMA_PREFS`
  - `DBMS_STATS.SET_DATABASE_PREFS`
  - `DBMS_STATS.SET_GLOBAL_PREFS`
- Parametere du har kontroll over:
  - `AUTOSTATS_TARGET`
  - `CASCADE`
  - `DEGREE`
  - `ESTIMATE_PERCENT`
  - `METHOD_OPT`
  - `NO_INVALIDATE`
  - `GRANULARITY`
  - `PUBLISH`
  - `INCREMENTAL`
  - `STALE_PERCENT`

Ny statistikk algoritme i 11g

- Problem i *Oracle10g* ved innhenting av statistikk for data som var som var ujevnt fordelt.
- Algoritmen for `AUTO_SAMPLE_SIZE` er skrevet om fullstendig.
- Ny algoritme er «hash» basert.
- Brukes slik: `AUTO_SAMPLE_SIZE` er satt til `DEFAULT` i `DBMS_STATS.GATHER.*STATS` prosedyrene.

DBMS_STATS - Histogram

- Slå av histogram i SALES tabellen, skjema SH:

```
11gR2 SYS@TEST11G SQL> BEGIN
                         DBMS_STATS.SET_TABLE_PREFS
                          ('SH',
                           'SALES',
                           'METHOD_OPT',
                           'FOR ALL COLUMNS SIZE 1');
                       END;
/

PL/SQL-prosedyren ble fullført.
```

DBMS_STATS – Kopiere statistikk

- Kopiere statistikk fra en partisjon `SALES_Q1_2010` til en ny partisjon `SALES_Q2_2010`:

```
11gR2 SYS@TEST11G SQL> BEGIN
                         DBMS_STATS.COPY_TABLE_STATS
                          ('SH',
                           'SALES',
                           'SALES_Q1_2010',
                           'SALES_Q2_2010',
                           FORCE => TRUE);
                       END;
/

PL/SQL-prosedyren ble fullført.
```

DBMS_STATS – Gruppere statistikk

- Statistikk kan utvides til å gjelde kolonner som kan grupperes.
- Her er fylke koblet mot land:

```
11gR2 SYS@TEST11G SQL> VARIABLE err VARCHAR2(255);
11gR2 SYS@TEST11G SQL> BEGIN
                         :err := DBMS_STATS.CREATE_EXTENDED_STATS
                          ('SH',
                           'CUSTOMERS',
                           '(COUNTRY_ID, CUST_STATE_PROVINCE)');
                       END;
/

PL/SQL-prosedyren ble fullført.
```

DBMS_STATS – Statistikk på partisjoner

- I *Oracle11g* er det mulig å legge til statistikk fra nye partisjoner uten å trekke med seg de eksisterende partisjonene.
- Introdusert «Synopsis» for å lagre informasjon om hver partisjon.

```
11gR2 SYS@TEST11G SQL> BEGIN
                         DBMS_STATS.SET_TABLE_PREFS
                           ('SH',
                            'SALES',
                            'INCREMENTAL',
                            'TRUE');
                       END;
/
PL/SQL-prosedyren ble fullført.

11gR2 SYS@TEST11G SQL> BEGIN
                         DBMS_STATS.GATHER_TABLE_STATS
                         ('SH',
                          'SALES');
                       END;
/
PL/SQL-prosedyren ble fullført.
```

DBMS_STATS – Låse statistikk

- I *Oracle10g* var det kun mulig å låse statistikk for tabeller eller skjemaer.

- I *Oracle11g* finnes det nye prosedyrer hvor dette også gjøres på partisjonsnivå.

```
11gR2 SYS@TEST11G SQL> BEGIN
                         DBMS_STATS.LOCK_PARTITION_STATS
                           ('SH',
                            'SALES',
                            'SALES_Q4_2009');
                       END;
/

PL/SQL-prosedyren ble fullført.
```

DBMS_STATS – Statistikk på vent...

- Det er nå mulig å vente med å «publisere» statistikk.

```
11gR2 SYS@TEST11G SQL> BEGIN
                         DBMS_STATS.SET_TABLE_PREFS
                           ('SH',
                            'SALES',
                            'PUBLISH',
                            'FALSE');
                       END;
/

PL/SQL-prosedyren ble fullført.

11gR2 SYS@TEST11G SQL> BEGIN
                         DBMS_STATS.GATHER_TABLE_STATS
                         ('SH',
                          'SALES');
                       END;

/

PL/SQL-prosedyren ble fullført.
```

DBMS_STATS – Publisere statistikk

- Sett parameteren `OPTIMIZER_USE_PENDING_STATISTICS` for å teste hvilken effekt statistikken har innen den blir publisert.

```
11gR2 SYS@TEST11G SQL> ALTER SESSION SET
                         optimizer_use_pending_statistics = TRUE;

Endret system.

Testing, testing, testing...
```

- Publiser statistikken når resultatet er lovende...

```
11gR2 SYS@TEST11G SQL> BEGIN
                         DBMS_STATS.PUBLISH_PENDING_STATS
                         ('SH',
                          'SALES');
                       END;
/

PL/SQL-prosedyren ble fullført.
```

10g: Group-by

- Først «JOIN», deretter «GROUP BY»

```
10gR2 SH@TEST10G SQL> EXPLAIN PLAN FOR
                      SELECT p.prod_id, sum(s.quantity_sold)
                      FROM Products p, Sales s
                      WHERE p.prod_id = s.prod_id
                      GROUP BY p.prod_id;

10gR2 SH@TEST10G SQL> SELECT * FROM TABLE(dbms_xplan.display());

-------------------------------------------------------------------------
| Id  | Operation           | Name     | Rows  | Bytes | Cost (%CPU)|
-------------------------------------------------------------------------
|   0 | SELECT STATEMENT    |          |    72 |   792 | 14323   (9) |
|   1 |  HASH GROUP BY      |          |    72 |   792 | 14323   (9) |
|*  2 |   HASH JOIN         |          |  9999K|   104M| 13478   (3) |
|   3 |    TABLE ACCESS FULL| PRODUCTS |    72 |   288 |     3   (0) |
|   4 |    TABLE ACCESS FULL| SALES    |  9999K|    66M| 13394   (2) |
-------------------------------------------------------------------------
```

11g: Group-by placement

- Utfører «GROUP BY» før «JOIN»

```
11gR2 SH@TEST11G SQL> EXPLAIN PLAN FOR
                      SELECT p.prod_id, sum(s.quantity_sold)
                      FROM Products p, Sales s
                      WHERE p.prod_id = s.prod_id
                      GROUP BY p.prod_id;

11gR2 SH@TEST10G SQL> SELECT * FROM TABLE(dbms_xplan.display());

-----------------------------------------------------------------------------
| Id  | Operation                     | Name      | Rows | Bytes | Cost (%CPU)|
-----------------------------------------------------------------------------
|   0 | SELECT STATEMENT              |           |   72 |  1512 | 15342   (8)|
|   1 |  SORT GROUP BY NOSORT         |           |   72 |  1512 | 15342   (8)|
|   2 |   MERGE JOIN                  |           |   92 |  1932 | 15342   (8)|
|   3 |    SORT JOIN                  |           |  135 |  2295 | 15323   (8)|
|   4 |     VIEW                      | VW_GBC_5  |  135 |  2295 | 15323   (8)|
|   5 |      HASH GROUP BY            |           |  135 |  945  | 15323   (8)|
|   6 |       PARTITION RANGE ALL     |           | 9999K|   66M | 14364   (2)|
|   7 |        TABLE ACCESS FULL      | SALES     | 9999K|   66M | 14364   (2)|
|*  8 |    SORT JOIN                  |           |   72 |   288 |    19   (6)|
|   9 |     TABLE ACCESS FULL         | PRODUCTS  |   72 |   288 |    18   (0)|
-----------------------------------------------------------------------------
```

10g: Extended join predicate push down

- View og Table «JOIN» - HASH JOIN eller SORT MERGE JOIN

```
10gR2 SH@TEST10G SQL> EXPLAIN PLAN FOR
                      SELECT p.prod_id, v1.row_count
                      FROM products p,
                          (SELECT s.prod_id, count(*) row_count
                           FROM sales s
                           WHERE s.quantity_sold BETWEEN 1 AND 47
                           GROUP BY s.prod_id) v1
                      WHERE p.supplier_id = 12
                      AND   p.prod_id = v1.prod_id(+);

10gR2 SH@TEST10G SQL> SELECT * FROM TABLE(dbms_xplan.display());

-------------------------------------------------------------------------
| Id  | Operation             | Name     | Rows  | Bytes | Cost (%CPU)|
-------------------------------------------------------------------------
|   0 | SELECT STATEMENT      |          |     1 |    33 | 13844   (5)|
|*  1 |  HASH JOIN OUTER      |          |     1 |    33 | 13844   (5)|
|*  2 |   TABLE ACCESS FULL   | PRODUCTS |     1 |     7 |     3   (0)|
|   3 |   VIEW                |          |   135 |  3510 | 13841   (5)|
|   4 |    HASH GROUP BY      |          |   135 |   945 | 13841   (5)|
|*  5 |     TABLE ACCESS FULL | SALES    |  4795K|    32M| 13454   (3)|
-------------------------------------------------------------------------
```

11g: Extended join predicate push down

- I 11g utvidet til å inkludere «GROUP BY», «DISTINCT» og «SEMI-JOINS».

```
11gR2 SH@TEST11G SQL> SELECT * FROM TABLE(dbms_xplan.display());

-------------------------------------------------------------------------------------
| Id  | Operation                          | Name           | Rows  | Bytes | Cost (%CPU) |
-------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT                   |                |     1 |    20 |  8552   (1) |
|   1 |  NESTED LOOPS OUTER                |                |     1 |    20 |  8552   (1) |
|*  2 |   TABLE ACCESS FULL                | PRODUCTS       |     1 |     7 |    18   (0) |
|   3 |   VIEW PUSHED PREDICATE            |                |     1 |    13 |             |
|*  4 |    FILTER                          |                |       |       |             |
|   5 |     SORT AGGREGATE                 |                |     1 |     7 |             |
|   6 |      PX COORDINATOR                |                |       |       |             |
|   7 |       PX SEND QC (RANDOM)          | :TQ10000       |     1 |     7 |             |
|   8 |        SORT AGGREGATE              |                |     1 |     7 |             |
|   9 |  PX PARTITION RANGE ALL            |                | 34486 |  235K |  8533   (1) |
|* 10 |   TABLE ACCESS BY LOCAL INDEX ROWID| SALES          | 34486 |  235K |  8533   (1) |
|  11 |    BITMAP CONVERSION TO ROWIDS     |                |       |       |             |
|* 12 |     BITMAP INDEX SINGLE VALUE      | SALES_PROD_BIX |       |       |             |
-------------------------------------------------------------------------------------
```

«Outlines» og «Baselines»

- «Outlines» støttes ikke i *Oracle11g*.
- «SQL Plan Management» (SPM) er det nye...
  - Beholder historikken til eksekveringsplanene.
- Nye planer blir evaluert mot eksisterende plan.
- «SQL Plan baseline» består av aksepterte eksekveringsplaner for en SQL-setning.

10g: Outlines

```
10gR2 SYS@TEST10G SQL> GRANT CREATE ANY OUTLINE TO sh;
GRANT-kommandoen var vellykket.

10gR2 SYS@TEST10G SQL> ALTER SYSTEM SET
                         create_stored_outlines = OLDPLAN;
```

```
10gR2 SYS@TEST10G SQL> ALTER SYSTEM SET
                         create_stored_outlines = FALSE;

10gR2 SYS@TEST10G SQL> CONNECT sh/sh
Tilkoblet.

10gR2 SH@TEST10G SQL> SELECT name, sql_text
                      FROM user_outlines
                      WHERE category = 'OLDPLAN';

NAME                            SQL_TEXT
------------------------------ --------------------------------
SYS_OUTLINE_10041209010870651  SELECT SUM(amount_sold),
                                      t.calendar_year,
                                      t.calendar_quarter_number,

152 rader valgt.
```

10g: Eksportere outlines

```
oracle@11z:dpdump [TEST10G] $ exp outln \
                              file=sh_outline.dmp \
                              owner=OUTLN \
                              rows=Y \
                              statistics=NONE

Export: Release 10.2.0.4.0 - Production on Ma Apr 12 09:27:38 2010

Tilkoblet: Oracle Database 10g Enterprise Edition 10.2.0.4.0 - 64bit

Eksport utført i WE8ISO8859P1-tegnsett og AL16UTF16 NCHAR-tegnsett

...
. . eksporterer tabell                  OL$        152 rader eksportert
. . eksporterer tabell             OL$HINTS       2577 rader eksportert
. . eksporterer tabell             OL$NODES        321 rader eksportert
...
Eksporten er avsluttet uten advarsler.
```

10g: Kopiere statistikk

```
10gR2 SYS@TEST10G SQL> BEGIN
                         DBMS_STATS.CREATE_STAT_TABLE
                           ('SH',
                            'SH_STATS_TAB');
                       END;
/

PL/SQL-prosedyren ble fullført.

10gR2 SYS@TEST10G SQL> BEGIN
                         DBMS_STATS.EXPORT_SCHEMA_STATS
                           ('SH',
                            'SH_STATS_TAB');
                       END ;
/

PL/SQL-prosedyren ble fullført.
```

- Statistikk til å benytte frem til oppgraderingen er stabil.
- Statistikk til å arbeide videre med i test systemer.

10g: Eksportere statistikk til 11g

```
oracle@11z:dpdump [TEST10G] $ exp sh \
                              file=sh_statistikk.dmp \
                              Tables='SH_STATS_TAB' \
                              rows=Y \
                              statistics=NONE

Export: Release 10.2.0.4.0 - Production on Ti Apr 13 08:23:54 2010

Tilkoblet: Oracle Database 10g Enterprise Edition 10.2.0.4.0 - 64bit

Eksport utført i WE8ISO8859P1-tegnsett og AL16UTF16 NCHAR-tegnsett

. . eksporterer tabell     SH_STATS_TAB      41075 rader eksportert

Eksporten er avsluttet uten advarsler.
```

11g: Importere outlines fra 10g

```
oracle@11z:dpdump [TEST11G] $ imp outln \
                              file=sh_outline.dmp  \
                              full=Y \
                              ignore=Y

Import: Release 11.2.0.1.0 - Production on Ma Apr 12 10:27:01 2010

Koblet til: Oracle Database 11g Enterprise Edition 11.2.0.1.0 64bit

Eksportfil opprettet av EXPORT:V10.02.01 via vanlig tilgangsvei
import utført i tegnsettet WE8ISO8859P1 og tegnsettet AL16UTF16 NCHAR

. importerer objektene til OUTLN til OUTLN
. . importerer tabell                    "OL$"        152 rader importert
. . importerer tabell               "OL$HINTS"       2577 rader importert
. . importerer tabell               "OL$NODES"        321 rader importert

Importen er avsluttet uten advarsler.
```

11g: Importere statistikk fra 10g

```
oracle@11z:dpdump [TEST11G] $ imp sh \
                              file=sh_statistikk.dmp  \
                              full=Y \
                              ignore=Y

...

Import: Release 11.2.0.1.0 - Production on Tue Apr 13 11:51:29 2010

Koblet til: Oracle Database 11g Enterprise Edition 11.2.0.1.0 64bit

Eksportfil opprettet av EXPORT:V10.02.01 via vanlig tilgangsvei
import utført i tegnsettet WE8ISO8859P1 og tegnsettet AL16UTF16 NCHAR

. importerer objektene til SH til SH
. . importerer tabell     "SH_STATS_TAB"      41075 rader importert

Importen er avsluttet uten advarsler.
```

11g: Kopiere statistikk fra 10g

```
11gR2 SYS@TEST11G SQL> BEGIN
                         DBMS_STATS.DELETE_SCHEMA_STATS
                           ('SH');
                       END;
/
PL/SQL-prosedyren ble fullført.

11gR2 SYS@TEST11G SQL> BEGIN
                         DBMS_STATS.UPGRADE_STAT_TABLE
                           ('SH',
                            'SH_STATS_TAB');
                       END ;
/
PL/SQL-prosedyren ble fullført.

11gR2 SYS@TEST11G SQL> BEGIN
                         DBMS_STATS.IMPORT_SCHEMA_STATS
                           ('SH',
                            'SH_STATS_TAB');
                       END ;
/
PL/SQL-prosedyren ble fullført.
```

11g: Migrere outlines til baselines

```
11gR2 SYS@TEST11G SQL> VARIABLE report CLOB;
11gR2 SYS@TEST11G SQL> BEGIN
                         :report := DBMS_SPM.MIGRATE_STORED_OUTLINE
                         (attribute_name => 'ALL');
                       END;
/

PL/SQL-prosedyren ble fullført.
```

- Kan også hente eksekveringsplaner fra:
  - SQL Tuning Set (STS).
  - Cursor Cache ved å sette
    - `OPTIMIZER_FEATURES_ENABLE = 10.2.0.4`
  - SQL baselines fra en tabell.

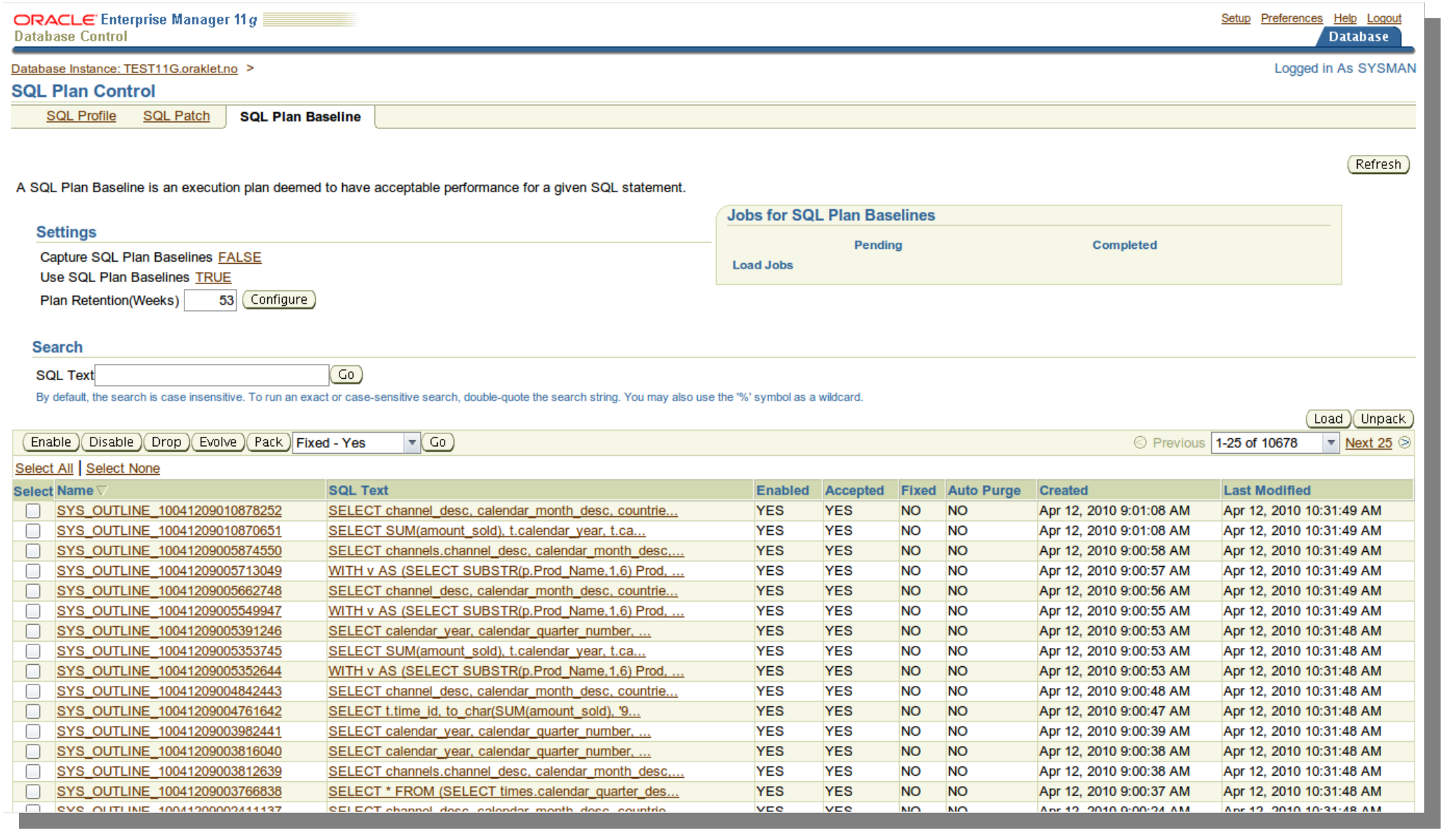
11g: Verifisere baselines

```
11gR2 SH@TEST11G SQL> EXPLAIN PLAN FOR
                      SELECT SUM(amount_sold),
                             t.calendar_year,
                             t.calendar_quarter_number,
                             c.country_name
                      FROM sales s,
                          times t,
                          countries c,
                          customers cu
                      WHERE s.time_id         = t.time_id
                        AND t.calendar_year   = '2007'
                        AND cu.country_id     = c.country_id
                        AND s.cust_id         = cu.cust_id
                        AND c.country_iso_code = 'NZ'
                      GROUP BY t.calendar_year,
                               t.calendar_quarter_number,
                               c.country_name;

Forklart.
```

11g: Verifisere baselines i SQL

```
11gR2 SH@TEST11G SQL> SELECT * FROM TABLE(dbms_xplan.display());

-----------------------------------------------------------------------------
| Id  | Operation                               | Name            | Rows        |
-----------------------------------------------------------------------------
|   0 | SELECT STATEMENT                        |                 | 30          |
|   1 |  PX COORDINATOR                         |                 |             |
...
|* 12 |  HASH JOIN                              |                 |   667K      |
|  13 |   TABLE ACCESS BY LOCAL INDEX ROWID     | SALES           |  1825       |
|  14 |    NESTED LOOPS                         |                 |   666K      |
|  15 |     BUFFER SORT                         |                 |             |
|  16 |      PX RECEIVE                         |                 |             |
|  17 |       PX SEND ROUND-ROBIN               | :TQ10001        |             |
|* 18 |        TABLE ACCESS FULL                | TIMES           |   365       |
|  19 |     PARTITION RANGE ITERATOR            |                 |             |
|  20 |      BITMAP CONVERSION TO ROWIDS        |                 |             |
|* 21 |       BITMAP INDEX SINGLE VALUE         | SALES_TIME_BIX  |             |
|  22 |   BUFFER SORT                           |                 |             |
|  23 |    PX RECEIVE                           |                 |  1999K      |
|  24 |     PX SEND BROADCAST                   | :TQ10002        |  1999K      |
|  25 |      TABLE ACCESS FULL                  | CUSTOMERS       |  1999K      |
-----------------------------------------------------------------------------

Note
-----
   - SQL plan baseline "SYS_OUTLINE_10041209010870651" used for this statement
```

11g: Verifisere baselines i OEM



10g: Extended join predicate push down

- View og Table «JOIN» - HASH JOIN eller SORT MERGE JOIN

```
10gR2 SH@TEST10G SQL> EXPLAIN PLAN FOR
                      SELECT p.prod_id, v1.row_count
                      FROM products p,
                           (SELECT s.prod_id, count(*) row_count
                            FROM sales s
                            WHERE s.quantity_sold BETWEEN 1 AND 47
                            GROUP BY s.prod_id) v1
                      WHERE p.supplier_id = 12
                      AND   p.prod_id = v1.prod_id(+);

10gR2 SH@TEST10G SQL> SELECT * FROM TABLE(dbms_xplan.display());

-------------------------------------------------------------------------
| Id  | Operation             | Name     | Rows  | Bytes | Cost (%CPU)|
-------------------------------------------------------------------------
|   0 | SELECT STATEMENT      |          |     1 |    33 | 13844   (5)|
|*  1 |  HASH JOIN OUTER      |          |     1 |    33 | 13844   (5)|
|*  2 |   TABLE ACCESS FULL   | PRODUCTS |     1 |     7 |     3   (0)|
|   3 |   VIEW                |          |   135 |  3510 | 13841   (5)|
|   4 |    HASH GROUP BY      |          |   135 |   945 | 13841   (5)|
|*  5 |     TABLE ACCESS FULL | SALES    |  4795K|    32M| 13454   (3)|
-------------------------------------------------------------------------
```

11g: Extended join predicate push down

- I 11g utvidet til å inkludere «GROUP BY», «DISTINCT» og «SEMI-JOINS».

```
11gR2 SH@TEST11G SQL> SELECT * FROM TABLE(dbms_xplan.display());

-------------------------------------------------------------------------------------
| Id  | Operation                          | Name           | Rows  | Bytes | Cost (%CPU) |
-------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT                   |                |     1 |    20 |  8552   (1) |
|   1 |  NESTED LOOPS OUTER                |                |     1 |    20 |  8552   (1) |
|*  2 |   TABLE ACCESS FULL                | PRODUCTS       |     1 |     7 |    18   (0) |
|   3 |   VIEW PUSHED PREDICATE            |                |     1 |    13 |             |
|*  4 |    FILTER                          |                |       |       |             |
|   5 |     SORT AGGREGATE                 |                |     1 |     7 |             |
|   6 |      PX COORDINATOR                |                |       |       |             |
|   7 |       PX SEND QC (RANDOM)          | :TQ10000       |     1 |     7 |             |
|   8 |        SORT AGGREGATE              |                |     1 |     7 |             |
|   9 |  PX PARTITION RANGE ALL            |                | 34486 |  235K |  8533   (1) |
|* 10 |   TABLE ACCESS BY LOCAL INDEX ROWID| SALES          | 34486 |  235K |  8533   (1) |
|  11 |    BITMAP CONVERSION TO ROWIDS     |                |       |       |             |
|* 12 |     BITMAP INDEX SINGLE VALUE      | SALES_PROD_BIX |       |       |             |
-------------------------------------------------------------------------------------
```

11g: Hints og SPM

- Hva hvis en SQL må masseres på innsiden?
- Kjør den først for å legge den i bufferen.

```
11gR2 SH@TEST11G SQL> VARIABLE ctgy VARCHAR2(50);
11gR2 SH@TEST11G SQL> BEGIN
                        :ctgy := 'Photo';
                      END;
/

PL/SQL-prosedyren ble fullført.

11gR2 SH@TEST11G SQL> SELECT prod_name,
                             SUM(amount_sold)name,
                             SUM(amount_sold)
                      FROM   Sales s, Products p
                      WHERE  s.prod_id=p.prod_id
                        AND  prod_category = :ctgy
                      GROUP BY prod_name;


...
10 rader valgt.
```

11g: Finn SQL-ID

- Vi trenger `SQL_ID` fra `V$SQL`.

```
11gR2 SYS@TEST11G SQL> SELECT sql_id, sql_fulltext
                       FROM v$sql
                       WHERE sql_text LIKE '%SELECT prod_name, S%';

SQL_ID         SQL_FULLTEXT
-----------    ----------------------------------------------
3mba7k4am3k47 SELECT prod_name, SUM(amount_sold)
              FROM   Sales s, Products p
              WHERE s.prod_id
```

11g: SPM - opprett SQL plan baseline

- Hent SQLen fra bufferen

```
11gR2 SYS@TEST11G SQL> VARIABLE cnt NUMBER;

11gR2 SYS@TEST11G SQL> EXECUTE :cnt :=
                         DBMS_SPM.LOAD_PLANS_FROM_CURSOR_CACHE
                         (sql_id => '3mba7k4am3k47');

PL/SQL-prosedyren ble fullført.
```

11g: SPM – finn SQL-adresse

```
11gR2 SYS@TEST11G SQL> SELECT sql_handle,
                              sql_text,
                              plan_name,
                              enabled
                       FROM dba_sql_plan_baselines
                       WHERE sql_text LIKE '%SELECT prod_name, S%';

SQL_HANDLE                     SQL_TEXT
PLAN_NAME                 ENA
------------------------------ --------------------------------
------------------------------ ---
SYS_SQL_bf5c9b08f72bde3e SELECT prod_name, SUM(amount_sold)
                         FROM   Sales s, Products p
                         WHERE s.prod_id

SQL_PLAN_byr4v13vkrrjyc2717242 YES
```

11g: SPM – gjør SQLen utilgjengelig

```
11gR2 SYS@TEST11G SQL> VARIABLE cnt NUMBER;

11gR2 SYS@TEST11G SQL> BEGIN
                       :cnt := DBMS_SPM.ALTER_SQL_PLAN_BASELINE
                        (sql_handle => 'SYS_SQL_bf5c9b08f72bde3e',
                        plan_name => 'SQL_PLAN_byr4v13vkrrjyc2717242',
                        attribute_name => 'ENABLED',
                        attribute_value => 'NO');
                       END;
/

PL/SQL-prosedyren ble fullført.
```

11g: SPM – Sjekk igjen

```
11gR2 SYS@TEST11G SQL> SELECT sql_handle,
                             sql_text,
                             plan_name,
                             enabled
                      FROM dba_sql_plan_baselines
                      WHERE sql_text LIKE '%SELECT prod_name, S%';

SQL_HANDLE                     SQL_TEXT
PLAN_NAME            ENA
------------------------------ --------------------------------
------------------------------ ---
SYS_SQL_bf5c9b08f72bde3e SELECT prod_name, SUM(amount_sold)
                         FROM   Sales s, Products p
                         WHERE s.prod_id

SQL_PLAN_byr4v13vkrrjyc2717242 NO
```

11g: SPM – Legg inn SQL med hints

```
11gR2 SH@TEST11G SQL> VARIABLE ctgy VARCHAR2(50);
11gR2 SH@TEST11G SQL> BEGIN
                        :ctgy := 'Photo';
                      END;
/

PL/SQL-prosedyren ble fullført.

11gR2 SH@TEST11G SQL> SELECT /*+ INDEX(p) */ prod_name,
                             SUM(amount_sold)name,
                             SUM(amount_sold)
                      FROM   Sales s, Products p
                      WHERE s.prod_id=p.prod_id
                        AND    prod_category = :ctgy
                     GROUP BY prod_name;


...
10 rader valgt.
```

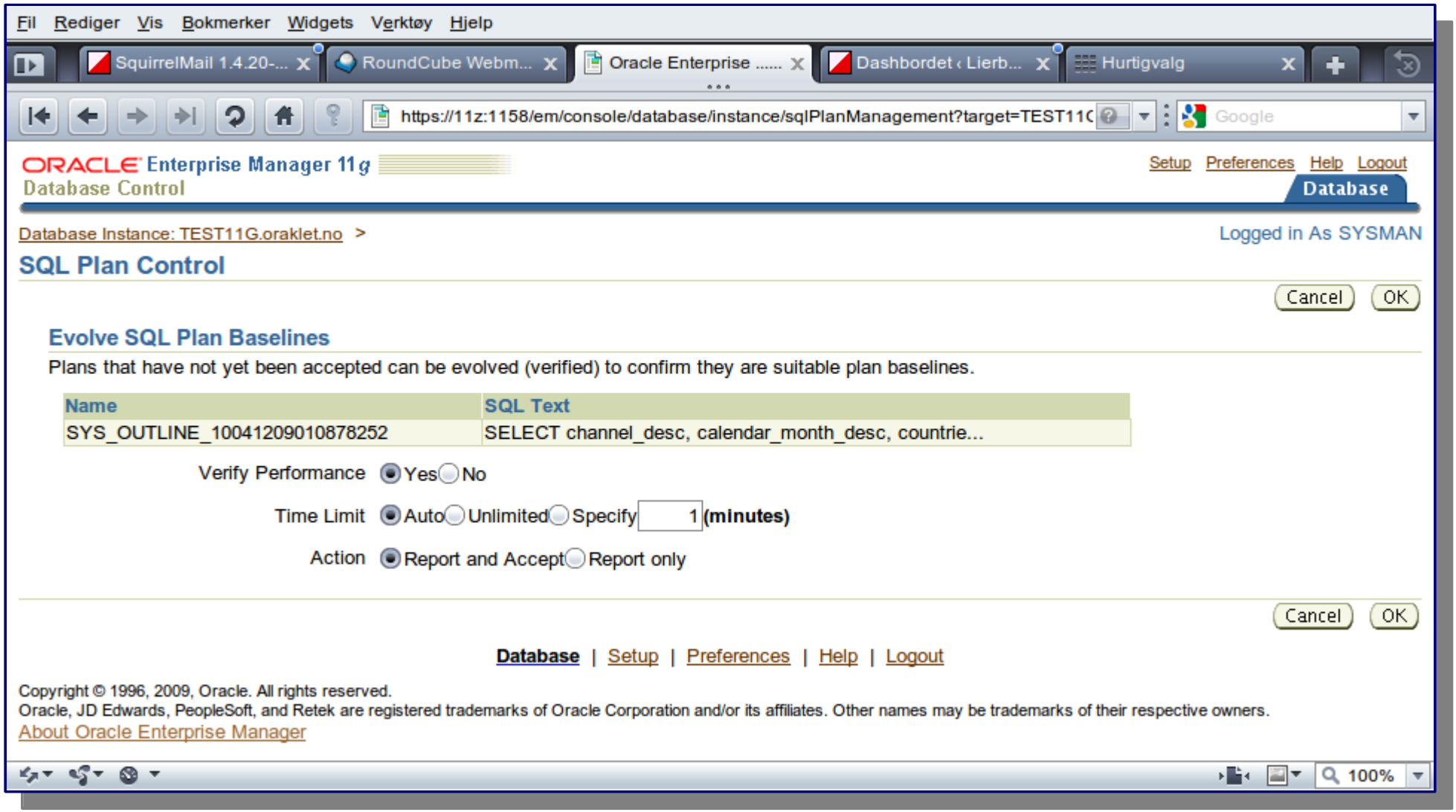
11g: SPM – Finn hinted SQL

```
11gR2 SYS@TEST11G SQL> SELECT sql_id,
                              plan_hash_value,
                              sql_fulltext
                       FROM   V$SQL
                       WHERE  sql_text LIKE
                         '%SELECT /*+ INDEX(p) */ prod_na%';

SQL_ID         PLAN_HASH_VALUE SQL_FULLTEXT
-------------- --------------- ----------------------------
ghf87ujpcfq22     1329740961   SELECT /*+ INDEX(p) */
                               prod_name, SUM(amount_sold)
                               FROM   Sales s, Products p
```

11g: SPM – ny plan på plass...

```
11gR2 SYS@TEST11G SQL> BEGIN
                         :cnt := DBMS_SPM.LOAD_PLANS_FROM_CURSOR_CACHE
                             (sql_id => 'ghf87ujpcfq22',
                             plan_hash_value => 1329740961,
                             sql_handle => 'SYS_SQL_bf5c9b08f72bde3e');
                           END;
/

PL/SQL-prosedyren ble fullført.
```

11g: SPM i Oracle Enterprise Manager



Konklusjon –

+++ Mange muligheter å håndtere optimalisering på.
++ Bra dokumentasjon om oppgradering til *Oracle11g*.
++ Fordel med støtte fra *Oracle Enterprise Manager*.

--- Vanskelig når flere miljøer må oppgraderes.
-- Tar tid å sjekke alle situasjoner som kan oppstå.
-- *Hva med Oracle Warehouse Builder?*