ORACLE®

ORACLE

12 Things about Oracle Database 12c

David Bayard
Oracle Enterprise Architect
Oracle North America Architecture Team

ORACLE DATABASE 12c

12 Things

1. It is Shipping
2. Hello EM Express
3. Multitenant Database Option
4. Automatic Data Optimization
5. PL/SQL from SQL
6. Improved Defaults
7. Increased Size Limits for VARCHAR2
8. Easy Top-N and Pagination Queries
9. Row Pattern Matching
10. Partitioning Improvements
11. Temporary UNDO
12. SQL Translation Framework

Much, Much more. Examples…

- Application Continuity & Transaction Guard
- Flex ASM
- Global Data Services
- Cross-Platform Backup and Restore
- Data Guard Far Sync
- Data Redaction
- Temporal Support (Transaction Time Temporal / Valid Time Temporal)
- Adaptive Execution Plans
- Enhanced Statistics
- Queryable Patch Inventory
- Oracle Intelligent Storage Protocol (OISP)

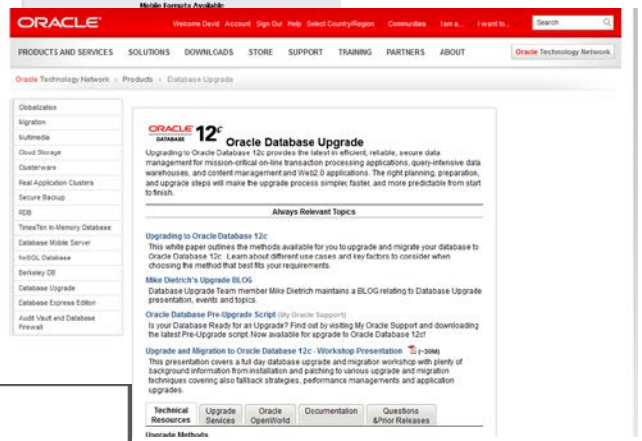
Copyright © 2012, Oracle and/or its affiliates. All rights reserved.
Insert Information Protection Policy Classification from Slide 13

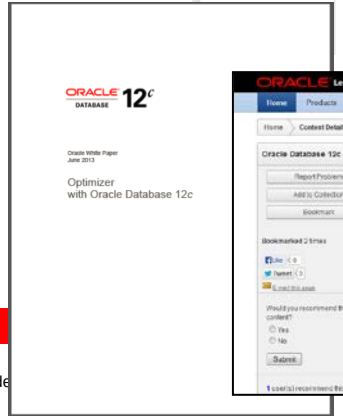
#1 It is Shipping

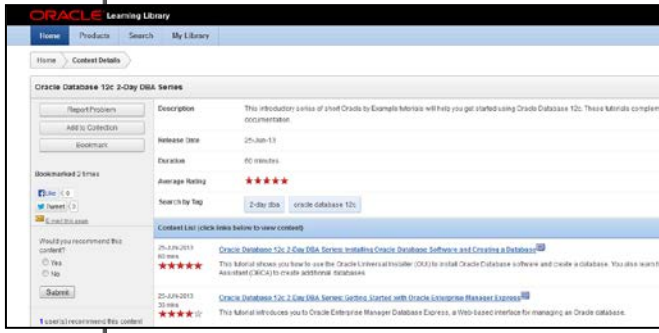
 Copyright © 2012, Oracle and/or its affiliates. All rights reserved. Insert Information Protection Policy Classification from Slide 13

Key 12c Resources

- Documentation Set
  - New Features Guide
  - License Guide
- Database Upgrade Microsite at OTN
  - Google: “Oracle Database Upgrade”
  - Upgrade and Migrate to Database 12c Workshop Presentation
- *“What to expect from the Optimizer with Oracle Database 12c”* White Paper
- Oracle Learning Library
  - 12c 2-Day DBA Series







 | Copyright © 2012, Oracle and/or its affiliates. All rights reserved. | Insert Information Protection Policy Classification from Slide

#2 Hello EM Express. Goodbye Database Control.

 | Copyright © 2012, Oracle and/or its affiliates. All rights reserved. | Insert Information Protection Policy Classification from Slide 13

ORACLE Enterprise Manager Database Express 12c

Help | SYSTEM | Log Out

ORCL (12.1.0.1.0) | Configuration | Storage | Security | Performance | host01.example.com

Database Home

Page Refreshed **2:54:04 AM GMT** Auto Refresh 1 Minute

Status

Up Time	19 hours, 49 minutes, 29 second
Type	Single instance (orcl)
Version	12.1.0.1.0 Enterprise Edition
Database Name	ORCL
Instance Name	orcl
Platform Name	Linux x86 64-bit
Host Name	host01.example.com
Oracle Home	/u01/app/oracle/product/12.1.0/c
Thread	1
Archiver	Stopped

Incidents - Last 24 Hours

Ins...	Time	Inci...	Pro...	Error
No Incidents				

Performance

Activity Class | Services

1 .8 .6 .4 .2 0

1:55 AM Apr 3 | 2:05 AM | 2:15 AM | 2:25 AM | 2:35 AM | 2:45 AM

Wait
User I/O
CPU

Resources

Host CPU

6% 4% 2%

Other

Active Sessions

100 80 60 40 20

Memory

1,600 MB 1,200 MB 800 MB 400 MB

Other SGA
Shared I/O Pool
Java Pool
Large Pool
Shared Pool
Buffer Cache
PGA

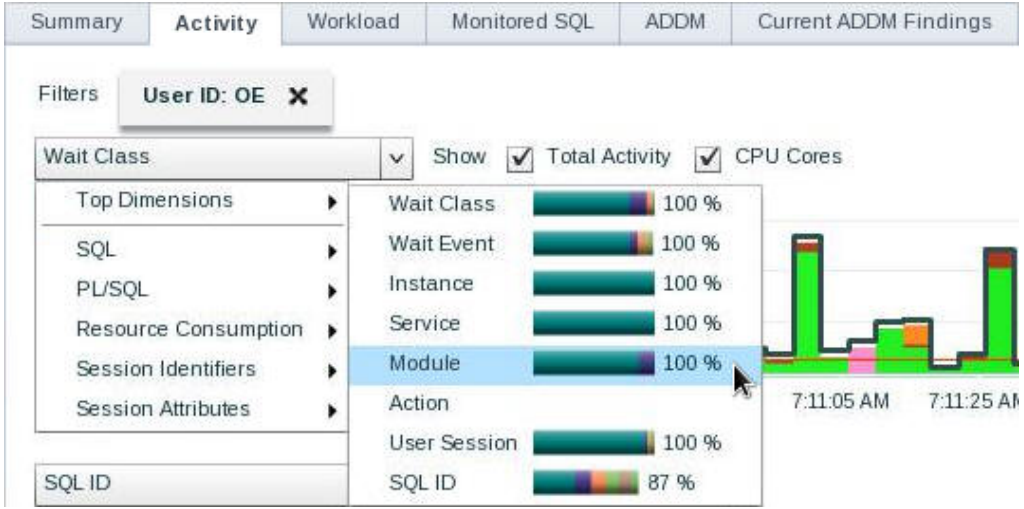
Data Storage

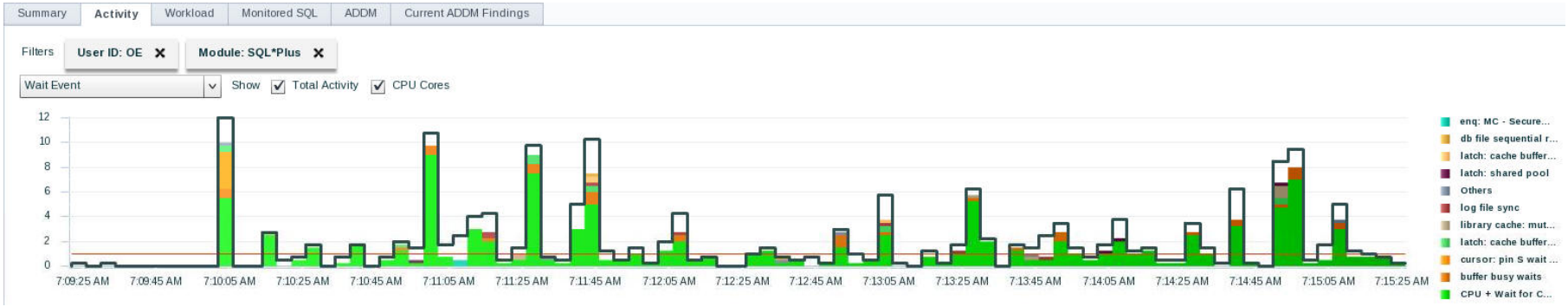
1.95 GB 1.56 GB 1.17 GB .78 GB .39 GB

User Data
Logs
Undo
Temporary
Sysaux
System

ORACLE

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.
Insert Information Protection Policy Classification from Slide 13





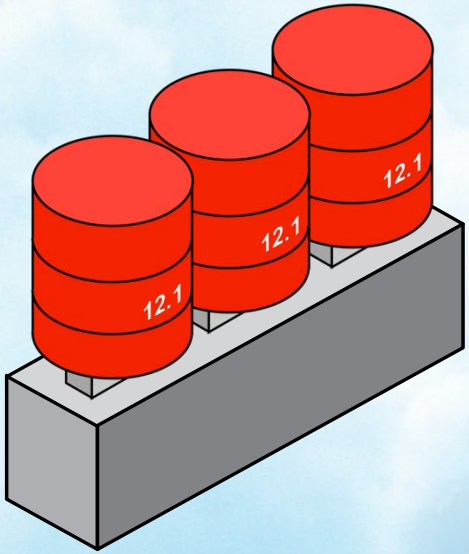
 | Copyright © 2012, Oracle and/or its affiliates. All rights reserved. | Insert Information Protection Policy Classification from Slide 13

#3 Oracle Multitenant Database Option (aka Pluggable Databases)

 Copyright © 2012, Oracle and/or its affiliates. All rights reserved. Insert Information Protection Policy Classification from Slide 13

New Multitenant Architecture

Ideal for consolidating databases on Clouds



- No application changes
- Isolation and multitenancy
- Fast provisioning and cloning
- Secure and highly available
- Lower IT costs
- Manage many as one
- Greater resource utilization
- Performant and scalable

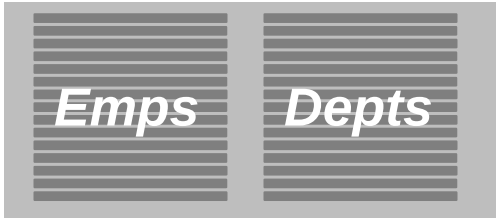
Copyright © 2012, Oracle and/or its affiliates. All rights reserved.
Insert Information Protection Policy Classification from Slide 13

Pluggable Databases

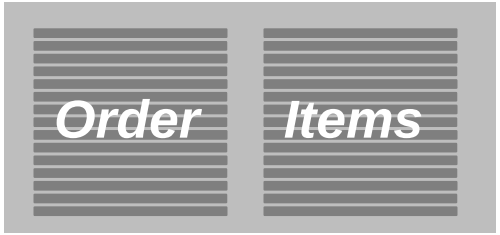
Old: Monolithic Data Dictionary Architecture

System Tablespace
Metadata for All Users

HR User Data



Sales User Data



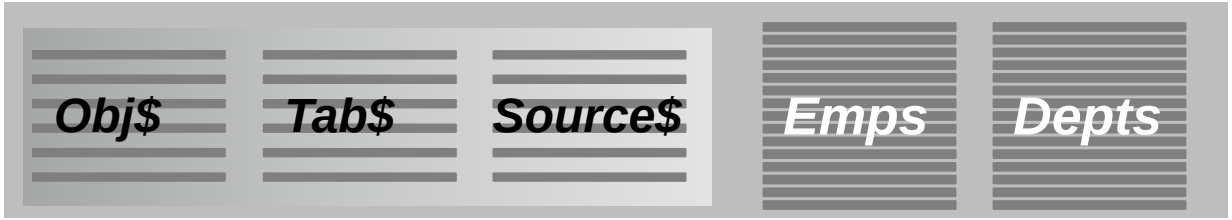
 Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

Pluggable Databases

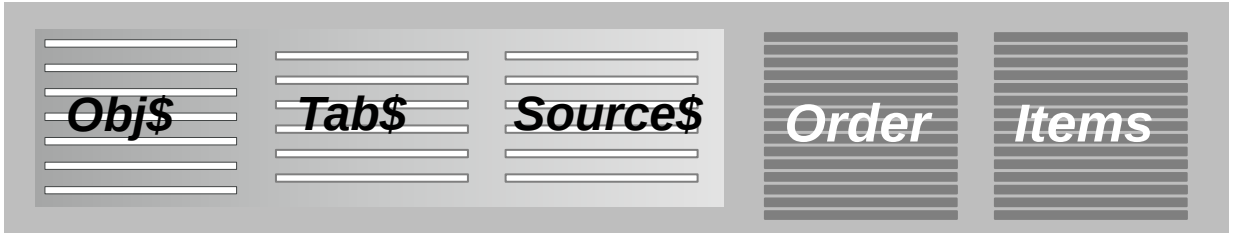
New: Modular Dictionary Architecture

Oracle Metadata Only

HR Metadata
& User Data



Sales Metadata
& User Data

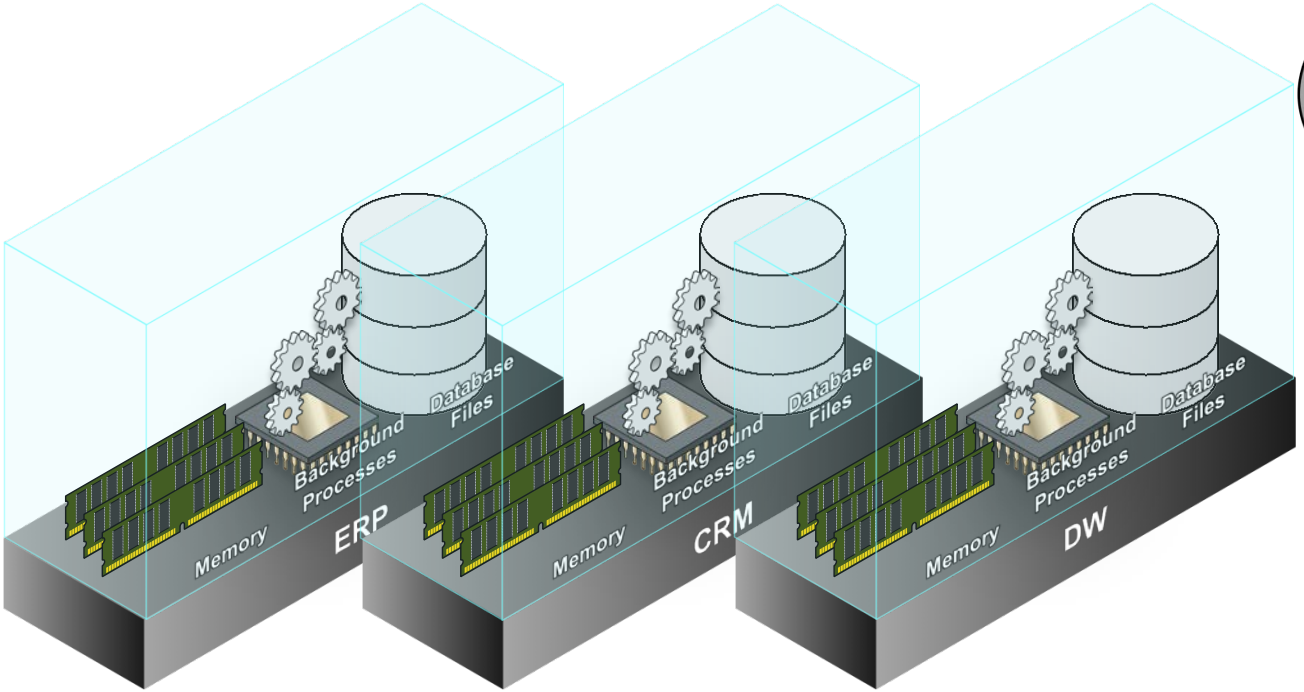


 | Copyright © 2012, Oracle and/or its affiliates. All rights reserved. | Insert Information Protection Policy Classification from Slide 13

Oracle Database Architecture

Requires memory, processes and database files

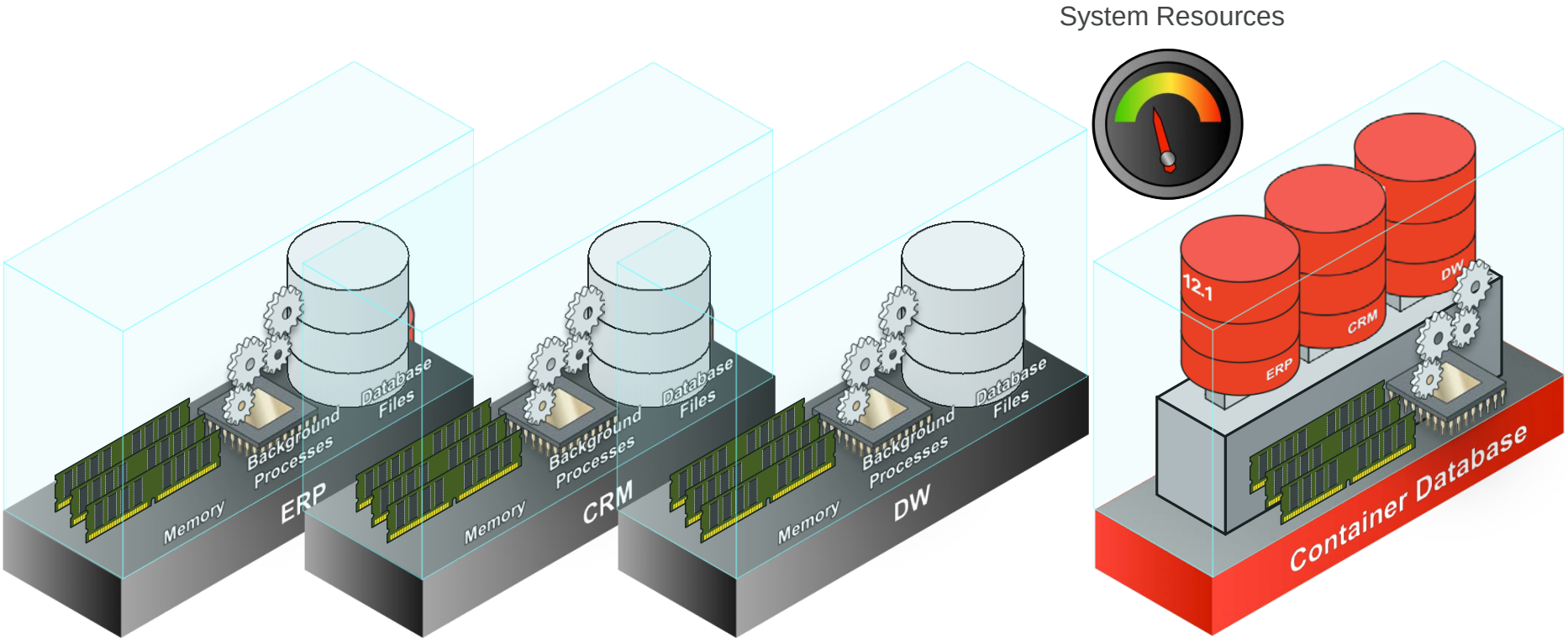
System Resources



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

New Multitenant Architecture

Memory and processes required at container level only

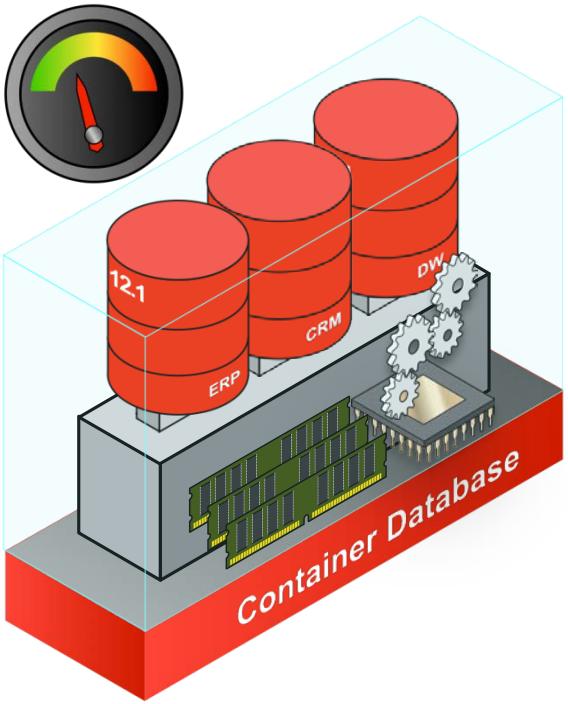


Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

New Multitenant Architecture

Memory and processes required at container level only

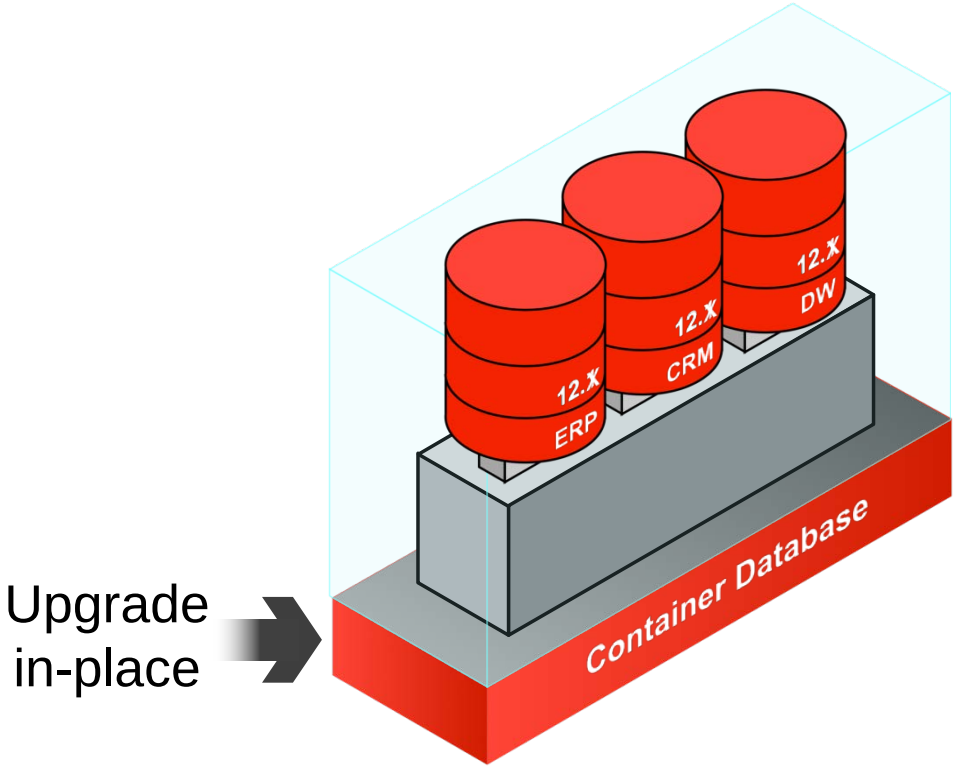
System Resources



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Simplified Patching

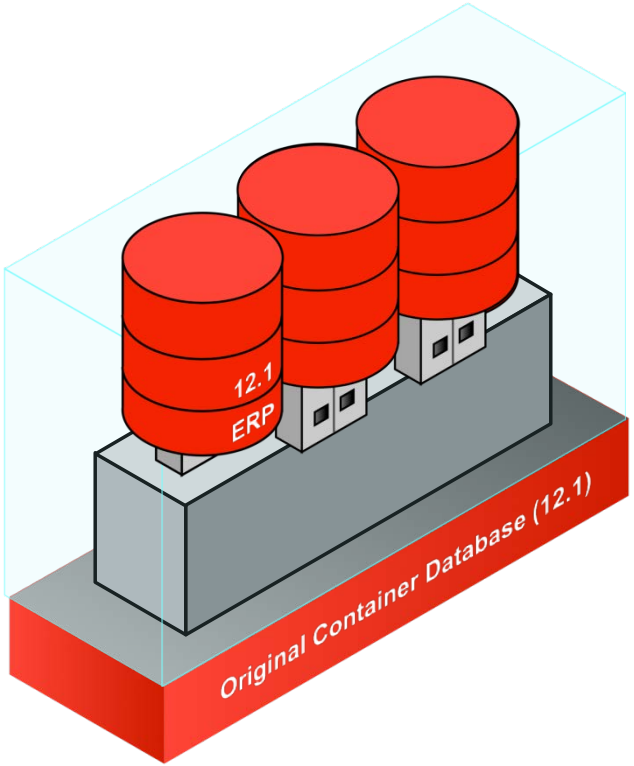
Apply changes once, all pluggable databases updated

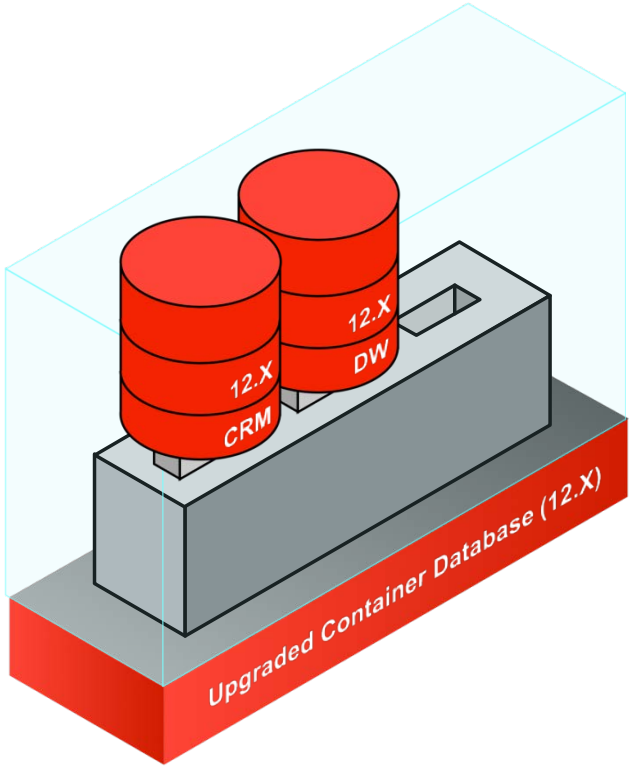


Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Simplified Upgrades

Flexible choice when patching & upgrading databases

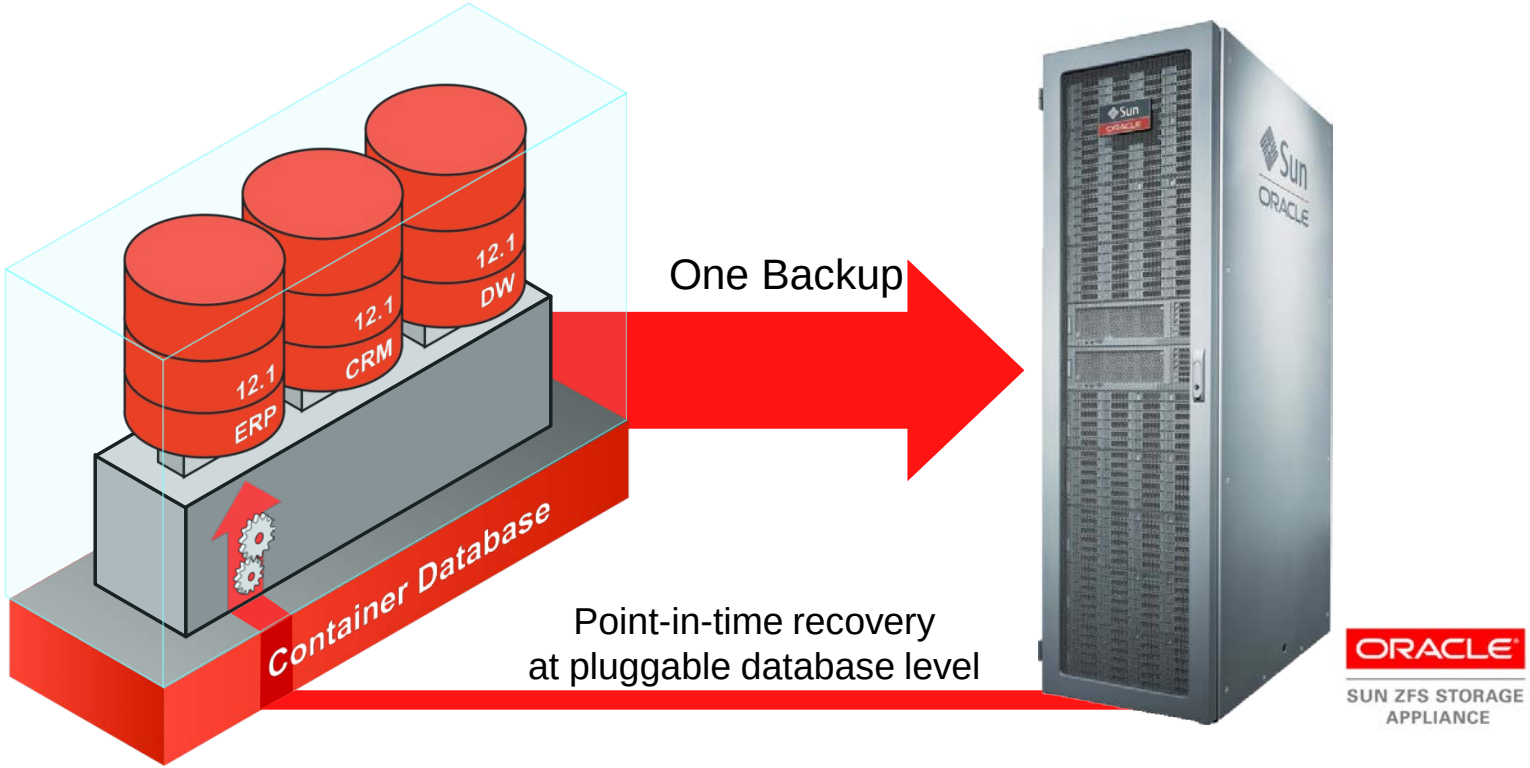




Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Manage Many Databases as One

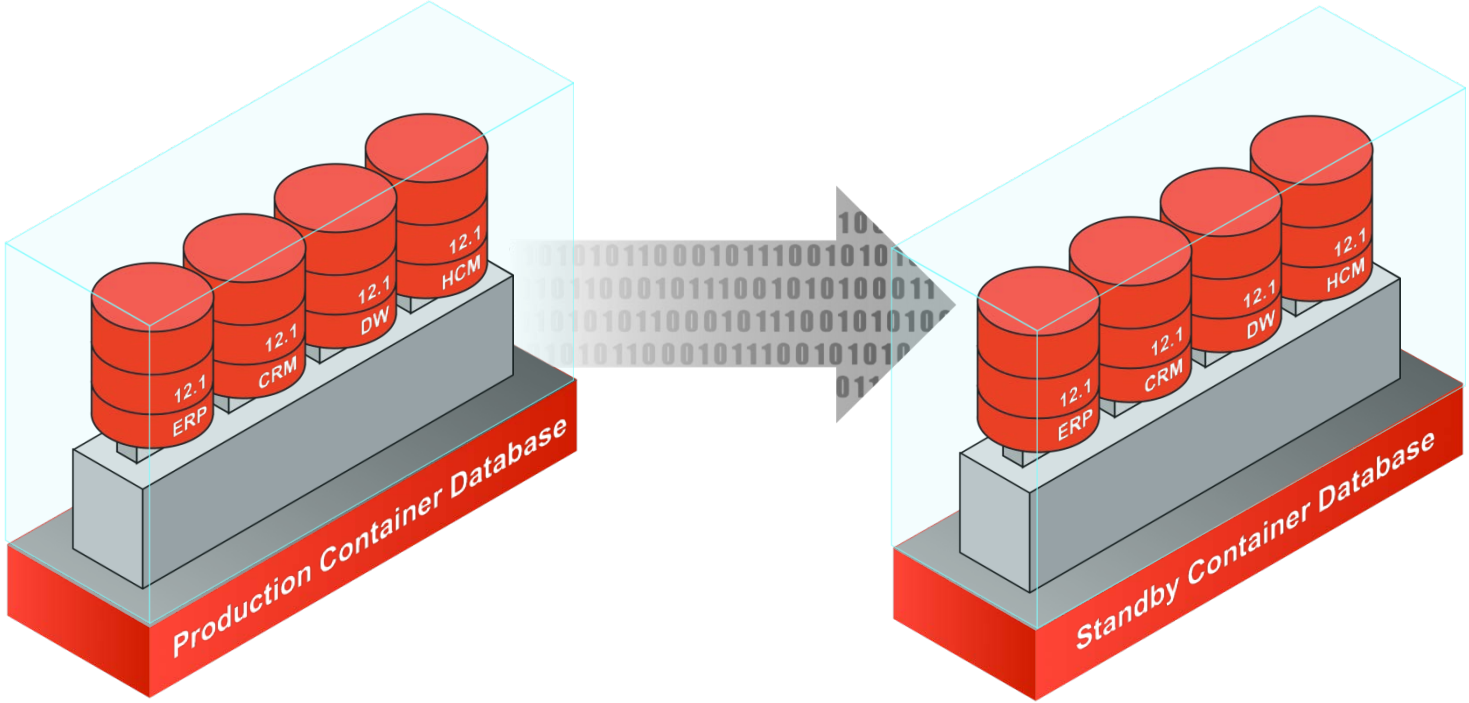
Backup databases as one; recover at pluggable database Level



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Manage Many Databases as One

One standby database covers all pluggable databases



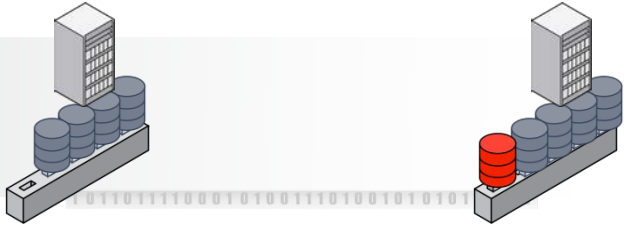
Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Managing Database Service Level Tiers

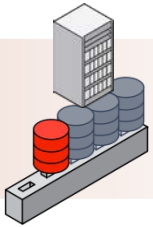
Change tiers as databases become more mission critical

GOLD RAC, Data Guard, Daily Incrementals

SILVER Data Guard, Daily Incrementals



BRONZE Weekly Full Backups



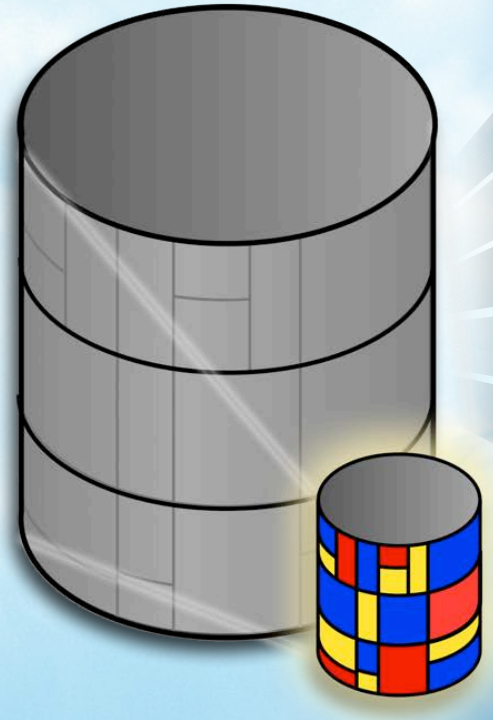
Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

#4 Automatic Data Optimization

 Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

Automatic Data Optimization

Optimize data storage based on usage



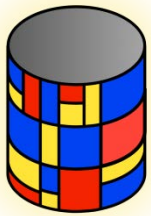
- Heat Map
- Smart Compression
- Automated Tiering
- Network Compression
- In Database Archiving

ORACLE

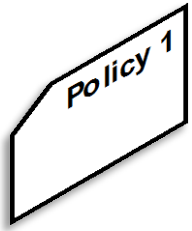
 Copyright © 2012, Oracle and/or its affiliates. All rights reserved. Insert Information Protection Policy Classification from Slide 13

Automatic Data Optimization

Simplifying the life cycle of data



- An in memory heat map tracks access to segments and blocks
  - Data is periodically written to disk
  - Information is accessible by views or stored procedures

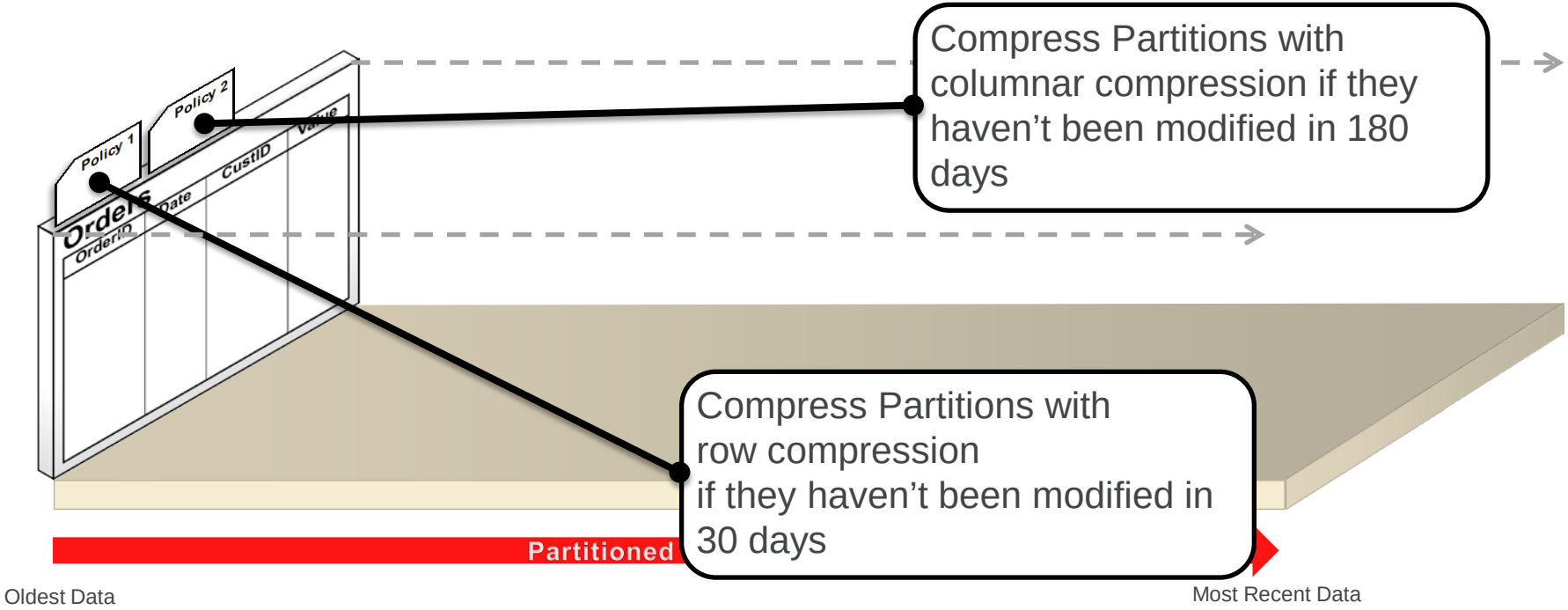


- Uses can attach policies to tables to compress or tier data based on access to data
  - Tables or Partitions can be moved between compression levels whilst data is still being accessed

 | Copyright © 2012, Oracle and/or its affiliates. All rights reserved. |

Automatic Data Optimization

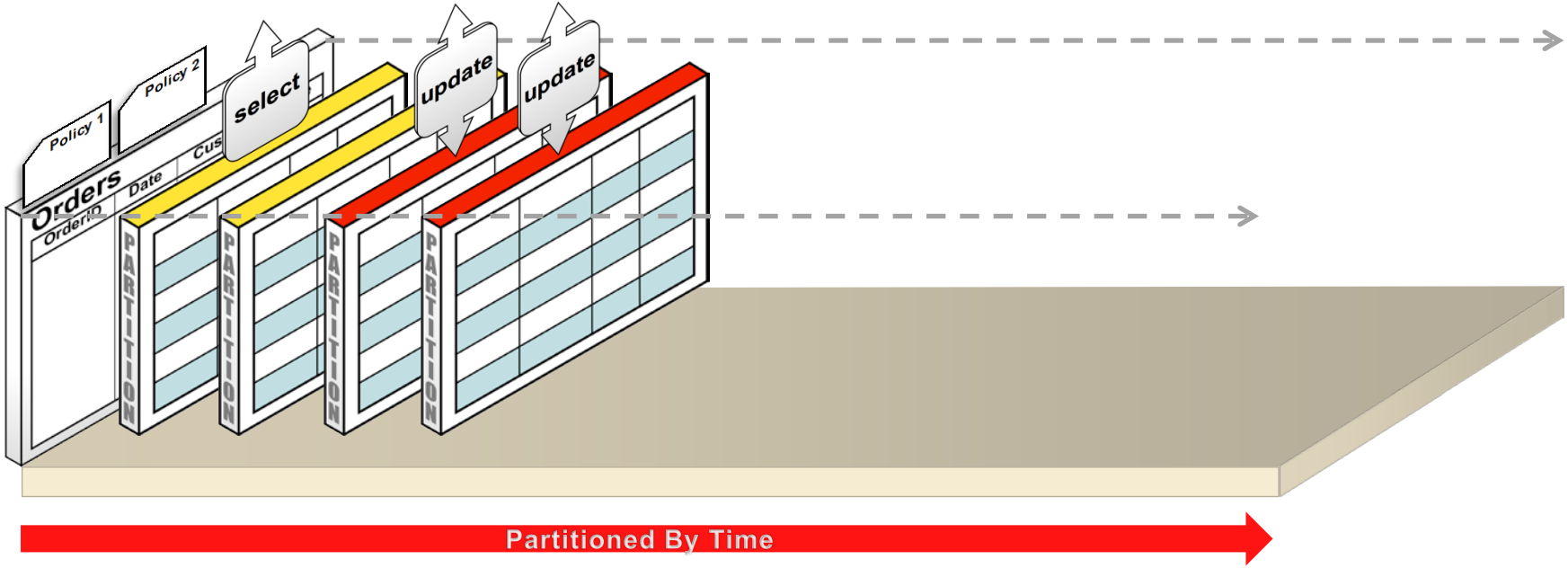
Add compression and tiering policies to tables



 | Copyright © 2012, Oracle and/or its affiliates. All rights reserved. | Insert Information Protection Policy Classification from Slide 13

Automatic Data Optimization

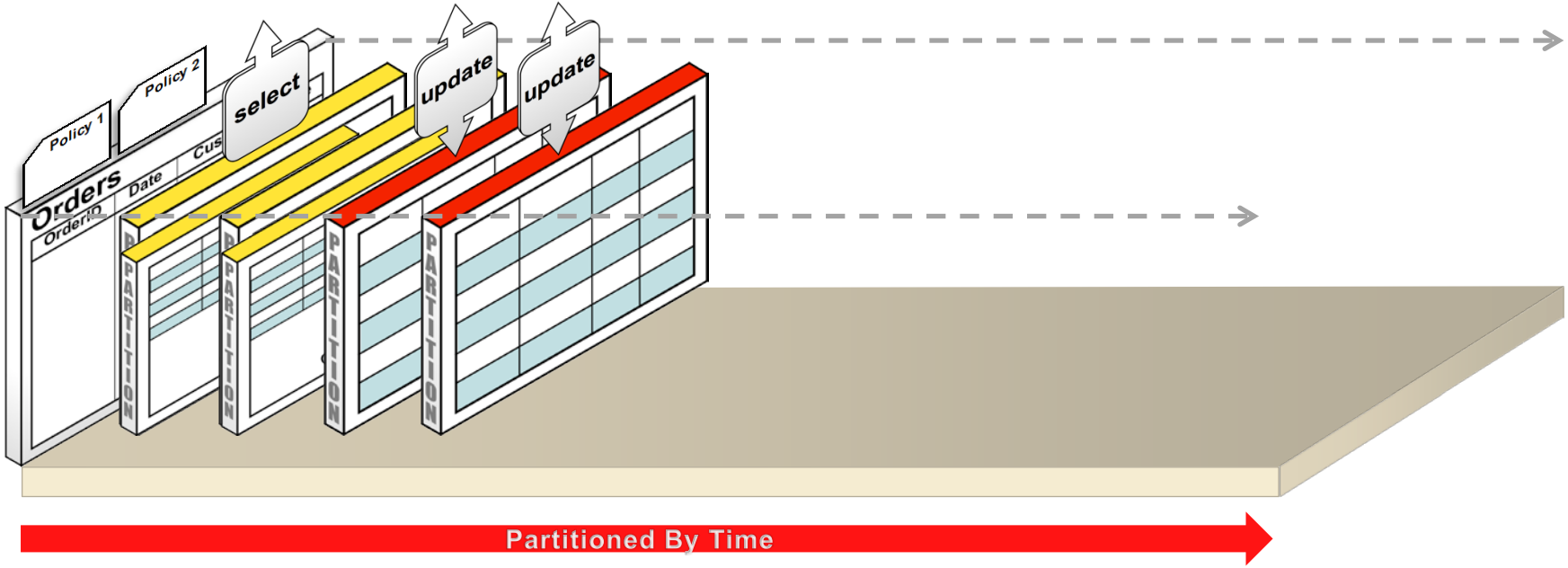
A heat map tracks the activity of segments and blocks



 | Copyright © 2012, Oracle and/or its affiliates. All rights reserved. | Insert Information Protection Policy Classification from Slide 13

Automatic Data Optimization

Policies are automatically applied to tables



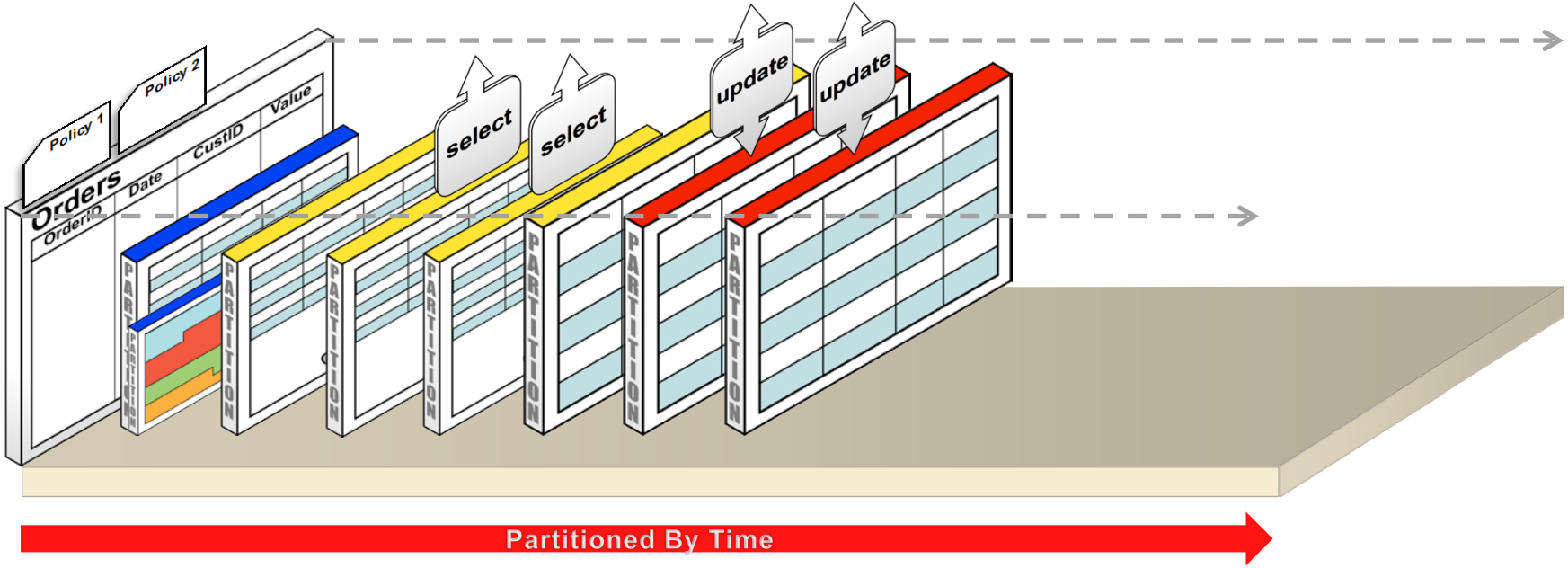
Oldest Data

Most Recent Data

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

Automatic Data Optimization

Policies are automatically applied to tables



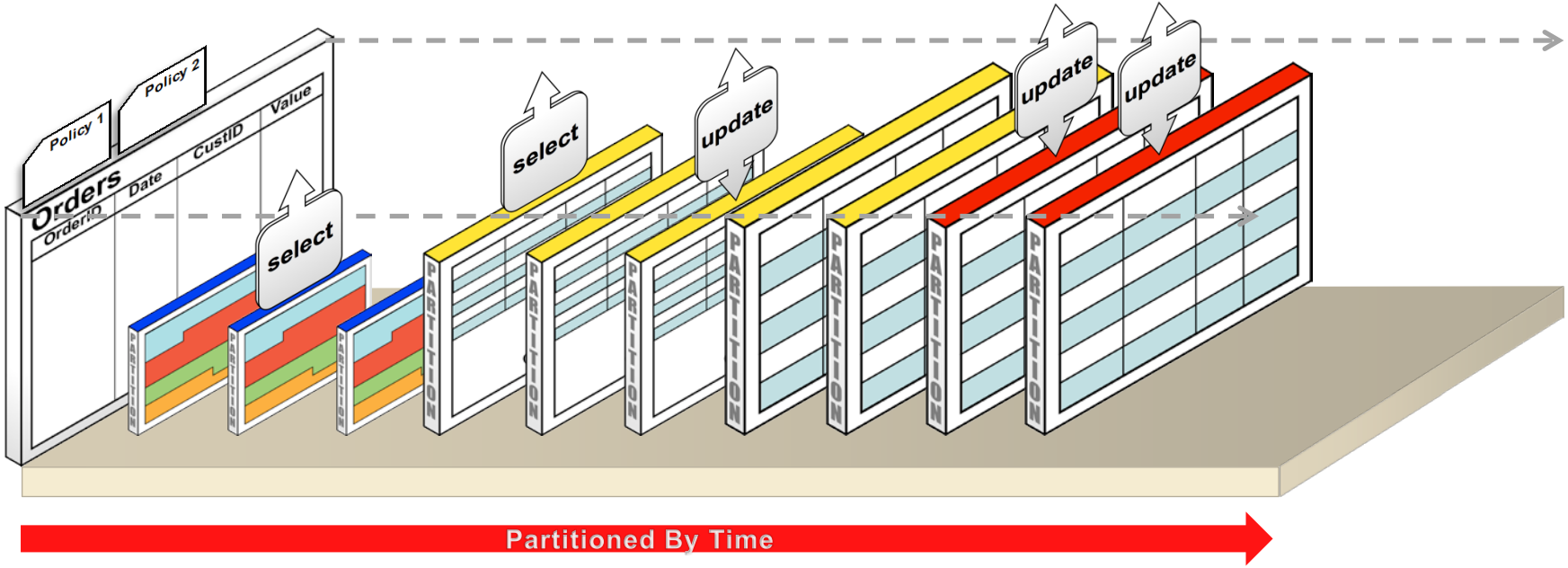
Oldest Data

Most Recent Data

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

Automatic Data Optimization

Policies are automatically applied to tables



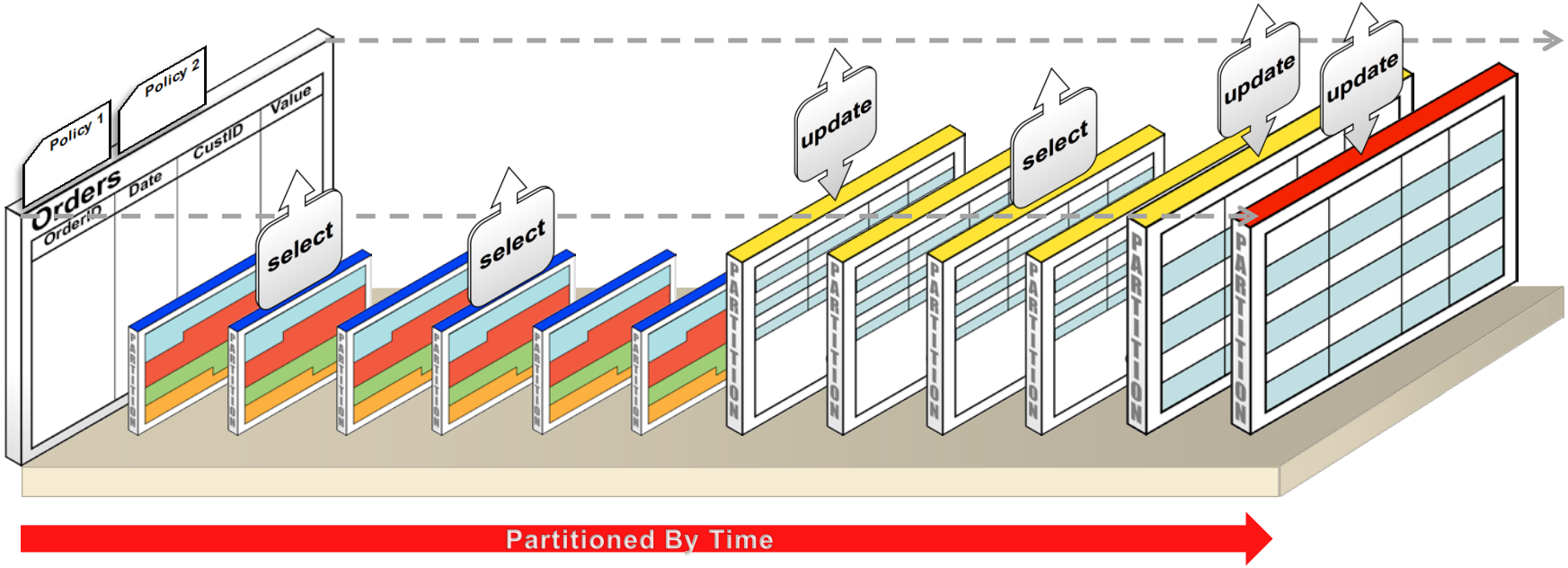
Oldest Data

Most Recent Data

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

Automatic Data Optimization

Reduce storage footprint, read compressed data faster



Oldest Data

Most Recent Data

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.
Confidential – Oracle Restricted
Use Information Lifecycle Policy Classification from Slide 13

#5 Even better PL/SQL from SQL

 Copyright © 2012, Oracle and/or its affiliates. All rights reserved. Insert Information Protection Policy Classification from Slide 13

PL/SQL from SQL

```
c##tkyte%CDB1> create table t ( x varchar2(5) );
Table created.

c##tkyte%CDB1> insert into t values ( 'a' );
c##tkyte%CDB1> insert into t values ( '1' );
c##tkyte%CDB1> insert into t values ( null );
```


Copyright © 2012, Oracle and/or its affiliates. All rights reserved.
Insert Information Protection Policy Classification from Slide 13

PL/SQL from SQL

```
c##tkyte%CDB1> create or replace
  2  function is_number_ool(x in varchar2)
  3  return varchar2
  4  is
  5      Plsql_Num_Error exception;
  6      pragma exception_init(Plsql_Num_Error, -06502);
  7  begin
  8      if (To_Number(x) is NOT null) then
  9        return 'Y';
 10      else
 11        return '';
 12      end if;
 13  exception
 14      when Plsql_Num_Error then
 15        return 'N';
 16  end Is_Number_ool;
 17  /
Function created.
```


Copyright © 2012, Oracle and/or its affiliates. All rights reserved.
Insert Information Protection Policy Classification from Slide 13

PL/SQL from SQL

```
c##tkyte%CDB1> select rownum, x,
  2          is_number_ool(x) is_num
  3    from t;

    ROWNUM X     IS_NUM
---------- ----- ----------
         1 a     N
         2 1     Y
         3
```


Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

PL/SQL from SQL

```
c##tkyte%CDB1> with
  2    function Is_Number
  3    (x in varchar2) return varchar2 is
  4      Plsql_Num_Error exception;
  5      pragma exception_init(Plsql_Num_Error, -06502);
  6    begin
  7      if (To_Number(x) is NOT null) then
  8        return 'Y';
  9      else
 10        return '';
 11      end if;
 12    exception
 13      when Plsql_Num_Error then
 14        return 'N';
 15    end Is_Number;
 16  select rownum, x, is_number(x) is_num from t
 17  /
```


Copyright © 2012, Oracle and/or its affiliates. All rights reserved.
Insert Information Protection Policy Classification from Slide 13

PL/SQL from SQL

```
select is_number_ool( to_char(object_id) ),
       is_number_ool( owner )
  from stage

call     count       cpu    elapsed       rows
------- ------  -------- ---------- ----------
Parse        1      0.00       0.00          0
Execute      1      0.00       0.00          0
Fetch      875      0.93       1.34      87310
------- ------  -------- ---------- ----------
total      877      0.93       1.34      87310
```


Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

PL/SQL from SQL

```
with
  function Is_Number ... end Is_Number;
select is_number( to_char(object_id) ) …,

call     count       cpu    elapsed       rows
------- ------  -------- ---------- ----------
Parse        1      0.00       0.00          0
Execute      1      0.00       0.00          0
Fetch      875      0.29       0.55      87310
------- ------  -------- ---------- ----------
total      877      0.29       0.55      87310
```


Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

#6 Improved Defaults

 Copyright © 2012, Oracle and/or its affiliates. All rights reserved. Insert Information Protection Policy Classification from Slide 13

Improved Defaults

- Default to a sequence
- Default when null inserted
- Identity Type
- Metadata-only Defaults for NULL columns

 Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

Improved Defaults - sequences

```
c##tkyte%CDB1> create sequence s;
Sequence created.

c##tkyte%CDB1> create table t
  2  ( x int default s.nextval primary key,
  3    y varchar2(30)
  4  )
  5  /
Table created.
```


Copyright © 2012, Oracle and/or its affiliates. All rights reserved.
Insert Information Protection Policy Classification from Slide 13

Improved Defaults - sequences

```
c##tkyte%CDB1> insert into t(y) values ('hello world');
1 row created.


c##tkyte%CDB1> select * from t;

         X Y
---------- ------------------------------
         1 hello world
```


Copyright © 2012, Oracle and/or its affiliates. All rights reserved.
Insert Information Protection Policy Classification from Slide 13

Improved Defaults – when null

```
c##tkyte%CDB1> create table t
  2  ( x number default s.nextval primary key,
  3    y number,
  4    z number default on null 42
  5  );

Table created.
```


Copyright © 2012, Oracle and/or its affiliates. All rights reserved.
Insert Information Protection Policy Classification from Slide 13

Improved Defaults – when null

```
c##tkyte%CDB1> insert into t (y, z) values ( 55, NULL );
c##tkyte%CDB1> insert into t (y,z) values ( 100, 200 );
c##tkyte%CDB1> insert into t (x,y,z) values (-1,-2,-3);

c##tkyte%CDB1> select * from t;

         X          Y          Z
---------- ---------- ----------
         2         55         42
         3        100        200
        -1         -2         -3
```


Copyright © 2012, Oracle and/or its affiliates. All rights reserved.
Insert Information Protection Policy Classification from Slide 13

Improved Defaults – identities

```
c##tkyte%CDB1> create table t
  2  ( x number generated as identity,
  3    y number
  4  )
  5  /
Table created.
```


Copyright © 2012, Oracle and/or its affiliates. All rights reserved.
Insert Information Protection Policy Classification from Slide 13

Improved Defaults – identities

```
c##tkyte%CDB1> insert into t (x,y) values (1,100);
insert into t (x,y) values (1,100)
            *
ERROR at line 1:
ORA-32795: cannot insert into a generated always identity column

c##tkyte%CDB1> insert into t (y) values (200);
1 row created.

c##tkyte%CDB1> select * from t;
         X          Y
---------- ----------
         1        200
```


Copyright © 2012, Oracle and/or its affiliates. All rights reserved.
Insert Information Protection Policy Classification from Slide 13

Improved Defaults – identities

```
c##tkyte%CDB1> create table t
  2  ( x number generated by default
  3             as identity
  4             (start with 42
  5              increment by 1000),
  6    y number
  7  )
  8  /

Table created.
```


Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

Improved Defaults – identities

```
c##tkyte%CDB1> insert into t (x,y) values (1,100);
1 row created.

c##tkyte%CDB1> insert into t (y) values (200);
1 row created.

c##tkyte%CDB1> select * from t;

         X          Y
---------- ----------
         1        100
        42        200
```


Copyright © 2012, Oracle and/or its affiliates. All rights reserved.
Insert Information Protection Policy Classification from Slide 13

Improved Defaults – metadata only defaults

```
c##tkyte%CDB1> create table t
  2  as
  3  select *
  4    from stage;
Table created.

c##tkyte%CDB1> exec show_space('T')
…
Full Blocks        .....................          1,437
Total Blocks.............................          1,536
Total Bytes..............................     12,582,912
Total MBytes.............................             12
…

PL/SQL procedure successfully completed.
```


Copyright © 2012, Oracle and/or its affiliates. All rights reserved.
Insert Information Protection Policy Classification from Slide 13

Improved Defaults – metadata only defaults

```
c##tkyte%CDB1> set timing on
c##tkyte%CDB1> alter table t add (data char(2000) default 'x');
Table altered.

Elapsed: 00:00:00.07


ops$tkyte%ORA11GR2> set timing on
ops$tkyte%ORA11GR2> alter table t add (data char(2000) default 'x');
Table altered.

Elapsed: 00:00:28.59
```


Copyright © 2012, Oracle and/or its affiliates. All rights reserved.
Insert Information Protection Policy Classification from Slide 13

Improved Defaults – metadata only defaults

```
c##tkyte%CDB1> exec show_space('T')
…
Full Blocks        .....................          1,437
Total Blocks.............................          1,536
Total Bytes..............................     12,582,912
Total MBytes.............................             12
…
PL/SQL procedure successfully completed.


ops$tkyte%ORA11GR2> exec show_space('T')
…
Total MBytes.............................              9   <<<= before

Total MBytes.............................            192   <<<= after
…
PL/SQL procedure successfully completed.
```


Copyright © 2012, Oracle and/or its affiliates. All rights reserved.
Insert Information Protection Policy Classification from Slide 13

#7 Increased Size Limit for VARCHAR2, NVARCHAR2, and RAW Data Types

 | Copyright © 2012, Oracle and/or its affiliates. All rights reserved. | Insert Information Protection Policy Classification from Slide 13

32k Strings

- Varchar2, NVarchar2 and Raw datatypes may be upto 32k in size, like in PL/SQL
- Compatible = 12.0.0.0 or higher
- Max_SQL_String_Size init.ora set to EXTENDED (default is not this)
- Not supported in clustered and index organized tables
- Will be stored out of line (LOB) but work just like long strings to your program

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

32k Strings

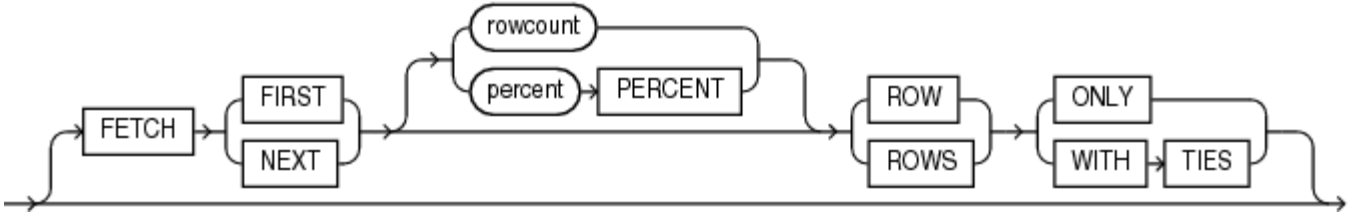
```
c##tkyte%CDB1> create table t ( x varchar(32767) );
Table created.

c##tkyte%CDB1> insert into t values ( rpad('*',32000,'*') );
1 row created.

c##tkyte%CDB1> select length(x) from t;

 LENGTH(X)
----------
     32000
```


Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

#8 Easy Top-N and pagination queries

 Copyright © 2012, Oracle and/or its affiliates. All rights reserved. Insert Information Protection Policy Classification from Slide 13

Row Limiting Clause



Copyright © 2012, Oracle and/or its affiliates. All rights reserved.
Insert Information Protection Policy Classification from Slide 13

Row Limiting Clause

```
c##tkyte%CDB1> create table t
  2  as
  3  select * from stage;
Table created.

c##tkyte%CDB1> create index t_idx on t(owner,object_name);
Index created.

c##tkyte%CDB1> set autotrace on explain
```


Copyright © 2012, Oracle and/or its affiliates. All rights reserved.
Insert Information Protection Policy Classification from Slide 13

Row Limiting Clause

```
c##tkyte%CDB1> select /*+ first_rows(5) */ owner, object_name, object_id
  2    from t
  3   order by owner, object_name
  4   FETCH FIRST 5 ROWS ONLY;
…
-----------------------------------------------------------------------------------------
| Id  | Operation                      | Name   | Rows   | Bytes | Cost (%CPU)| Time     |
-----------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT               |        |      5 |  1450 |     7   (0)| 00:00:01 |
|*  1 |  VIEW                          |        |      5 |  1450 |     7   (0)| 00:00:01 |
|*  2 |   WINDOW NOSORT STOPKEY        |        |      5 |   180 |     7   (0)| 00:00:01 |
|   3 |    TABLE ACCESS BY INDEX ROWID| T      | 87310 |  3069K|     7   (0)| 00:00:01 |
|   4 |     INDEX FULL SCAN            | T_IDX  |      5 |       |     3   (0)| 00:00:01 |
-----------------------------------------------------------------------------------------

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

   1 - filter("from$_subquery$_003"."rowlimit_$$_rownumber"<=5)
   2 - filter(ROW_NUMBER() OVER ( ORDER BY "OWNER","OBJECT_NAME")<=5)
```


Copyright © 2012, Oracle and/or its affiliates. All rights reserved.
Insert Information Protection Policy Classification from Slide 13

Row Limiting Clause

```
c##tkyte%CDB1> select /*+ first_rows(5) */ owner, object_name, object_id
  2    from t
  3   order by owner, object_name
  4  OFFSET 5 ROWS FETCH NEXT 5 ROWS ONLY;
…
-----------------------------------------------------------------------------------------
| Id  | Operation                      | Name   | Rows   | Bytes | Cost (%CPU)| Time     |
-----------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT               |        |      5 |  1450 |     7   (0)| 00:00:01 |
|*  1 |  VIEW                          |        |      5 |  1450 |     7   (0)| 00:00:01 |
|*  2 |   WINDOW NOSORT STOPKEY        |        |      5 |   180 |     7   (0)| 00:00:01 |
|   3 |    TABLE ACCESS BY INDEX ROWID| T      | 87310 |  3069K|     7   (0)| 00:00:01 |
|   4 |     INDEX FULL SCAN            | T_IDX  |      5 |       |     3   (0)| 00:00:01 |
-----------------------------------------------------------------------------------------
Predicate Information (identified by operation id):
---------------------------------------------------

   1 - filter("from$_subquery$_003"."rowlimit_$$_rownumber"<=CASE  WHEN (5>=0)
              THEN 5 ELSE 0 END +5 AND "from$_subquery$_003"."rowlimit_$$_rownumber">5)
   2 - filter(ROW_NUMBER() OVER ( ORDER BY "OWNER","OBJECT_NAME")<=CASE  WHEN
              (5>=0) THEN 5 ELSE 0 END +5)
```


Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

Row Limiting Clause

```
c##tkyte%CDB1> select owner, object_name, object_id
  2    from t
  3   order by owner, object_name
  4   FETCH NEXT 0.01 PERCENT ROWS ONLY;
…
9 rows selected.
-------------------------------------------------------------------------------------
| Id  | Operation           | Name | Rows  | Bytes |TempSpc| Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT    |      | 87310 |    25M|       |  1230   (1)| 00:00:01 |
|*  1 |  VIEW               |      | 87310 |    25M|       |  1230   (1)| 00:00:01 |
|   2 |   WINDOW SORT       |      | 87310 |  3069K|  4120K|  1230   (1)| 00:00:01 |
|   3 |    TABLE ACCESS FULL| T    | 87310 |  3069K|       |   401   (1)| 00:00:01 |
-------------------------------------------------------------------------------------

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

   1 - filter("from$_subquery$_003"."rowlimit_$$_rownumber"<=CEIL("from$_sub
              query$_003"."rowlimit_$$_total"*0.01/100))
```


Copyright © 2012, Oracle and/or its affiliates. All rights reserved.
Insert Information Protection Policy Classification from Slide 13

#9 Row Pattern Matching

 Copyright © 2012, Oracle and/or its affiliates. All rights reserved. Insert Information Protection Policy Classification from Slide 13

Row Pattern Matching

- New pattern matching clause in SQL called *“match_recognize”*
- Match_recognize clause enables users to:
  - Logically partition and order the data used in match_recognize clause
  - define patterns using regular expression syntax over pattern variables
  - the regular expression is matched against a sequence of rows
  - each pattern variable is defined using conditions on individual rows and aggregates

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

Row Pattern Matching

```
c##tkyte%CDB1> create table stocks
  2  ( symbol   varchar2(10),
  3    tstamp   date,
  4    price    number,
  5    primary key (symbol,tstamp)
  6  )
  7  organization index
  8  /
Table created.
```


Copyright © 2012, Oracle and/or its affiliates. All rights reserved.
Insert Information Protection Policy Classification from Slide 13

Row Pattern Matching

```
c##tkyte%CDB1> declare
  2      l_data sys.odciNumberList :=
  3             sys.odciNumberList
  4             ( 35, 34, 33, 34, 35, 36,
  5             37, 36, 35, 34, 35, 36, 37 );
  6      l_cnt  number := l_data.count;
  7  begin
  8      for i in 1 .. l_cnt
  9      loop
 10          insert into stocks
 11          ( symbol, tstamp, price )
 12          values
 13          ('ORCL', sysdate-l_cnt+i, l_data(i) );
 14      end loop;
 15      commit;
 16  end;
 17  /

PL/SQL procedure successfully completed.
```


Copyright © 2012, Oracle and/or its affiliates. All rights reserved.
Insert Information Protection Policy Classification from Slide 13

Row Pattern Matching

```
c##tkyte%CDB1> select symbol, tstamp, price,
  2         rpad('*',price,'*') hist
  3    from stocks
  4   order by symbol, tstamp;

SYMBOL     TSTAMP         PRICE HIST
---------- --------- ---------- ----------------------------------------
ORCL       01-SEP-12         35 ***********************************
ORCL       02-SEP-12         34 **********************************
ORCL       03-SEP-12         33 *********************************
ORCL       04-SEP-12         34 **********************************
ORCL       05-SEP-12         35 ***********************************
ORCL       06-SEP-12         36 ************************************
ORCL       07-SEP-12         37 *************************************
ORCL       08-SEP-12         36 ************************************
ORCL       09-SEP-12         35 ***********************************
ORCL       10-SEP-12         34 **********************************
ORCL       11-SEP-12         35 ***********************************
ORCL       12-SEP-12         36 ************************************
ORCL       13-SEP-12         37 *************************************
13 rows selected.
```


Copyright © 2012, Oracle and/or its affiliates. All rights reserved.
Insert Information Protection Policy Classification from Slide 13

Row Pattern Matching

```
c##tkyte%CDB1> SELECT *
  2  FROM stocks MATCH_RECOGNIZE
  3  ( PARTITION BY symbol
  4    ORDER BY tstamp
  5    MEASURES
  6       STRT.tstamp AS start_tstamp,
  7       LAST(DOWN.tstamp) AS bottom_tstamp,
  8       LAST(UP.tstamp) AS end_tstamp
  9    ONE ROW PER MATCH
 10    AFTER MATCH SKIP TO LAST UP
 11    PATTERN (STRT DOWN+ UP+)
 12    DEFINE
 13      DOWN AS DOWN.price < PREV(DOWN.price),
 14      UP AS UP.price > PREV(UP.price)
 15  ) MR
 16   ORDER BY MR.symbol, MR.start_tstamp;

SYMBOL     START_TST BOTTOM_TS END_TSTAM
---------- --------- --------- ---------
ORCL       01-SEP-12 03-SEP-12 07-SEP-12
ORCL       07-SEP-12 10-SEP-12 13-SEP-12
```

```
SYMBOL     TSTAMP         PRICE HIST
---------- --------- ---------- ----------------------------------------
ORCL       01-SEP-12         35 ************************************
ORCL       02-SEP-12         34 ***********************************
ORCL       03-SEP-12         33 **********************************
ORCL       04-SEP-12         34 ***********************************
ORCL       05-SEP-12         35 ************************************
ORCL       06-SEP-12         36 *************************************
ORCL       07-SEP-12         37 **************************************
ORCL       08-SEP-12         36 *************************************
ORCL       09-SEP-12         35 ************************************
ORCL       10-SEP-12         34 ***********************************
ORCL       11-SEP-12         35 ************************************
ORCL       12-SEP-12         36 *************************************
ORCL       13-SEP-12         37 **************************************
13 rows selected.
```


Copyright © 2012, Oracle and/or its affiliates. All rights reserved.
Insert Information Protection Policy Classification from Slide 13

#10 Partitioning Improvements

 Copyright © 2012, Oracle and/or its affiliates. All rights reserved. Insert Information Protection Policy Classification from Slide 13

Partitioning Improvements

- Asynchronous Global Index Maintenance for DROP and TRUNCATE partition
- Cascade Functionality for TRUCATE and EXCHANGE partition
- Multiple partition operations in a single DDL
- Online move of a partition (without DBMS_REDEFINITION)
- Interval + Reference partitioning

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.
Insert Information Protection Policy Classification from Slide 13

#11 Temporary UNDO

 Copyright © 2012, Oracle and/or its affiliates. All rights reserved. Insert Information Protection Policy Classification from Slide 13

Temporary UNDO

- UNDO for temporary tables can now be managed in TEMP
- Reduce the amount of UNDO in the UNDO tablespace
  - Better for retention periods for “real” data
- Reduce the size of the redo generated
- Allows for DML on temporary tables in Active Data Guard

- ALTER SYSTEM/SESSION SET TEMP_UNDO_ENABLED=true|false

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.
Insert Information Protection Policy Classification from Slide 13

Temporary UNDO

```
c##tkyte%CDB1> create global temporary table gtt
  2  on commit delete rows
  3  as
  4  select * from stage where 1=0;

Table created.
```


Copyright © 2012, Oracle and/or its affiliates. All rights reserved.
Insert Information Protection Policy Classification from Slide 13

Temporary UNDO

```
c##tkyte%CDB1> alter session set temp_undo_enabled = false;
Session altered.

c##tkyte%CDB1> insert into gtt
  2  select *
  3    from stage;
87310 rows created.

Statistics
----------------------------------------------------------
…
     566304  redo size
…

c##tkyte%CDB1> update gtt
  2     set object_name = lower(object_name);
87310 rows updated.

Statistics
----------------------------------------------------------
…
    8243680  redo size
…
```


Copyright © 2012, Oracle and/or its affiliates. All rights reserved.
Insert Information Protection Policy Classification from Slide 13

Temporary UNDO

```
c##tkyte%CDB1> alter session set temp_undo_enabled = true;
Session altered.

c##tkyte%CDB1> insert into gtt
  2  select *
  3    from stage;
87310 rows created.

Statistics
----------------------------------------------------------
…
        280  redo size
…

c##tkyte%CDB1> update gtt
  2     set object_name = lower(object_name);
87310 rows updated.

Statistics
----------------------------------------------------------
…
          0  redo size
…
```


Copyright © 2012, Oracle and/or its affiliates. All rights reserved.
Insert Information Protection Policy Classification from Slide 13

#12 SQL Translation Framework

 Copyright © 2012, Oracle and/or its affiliates. All rights reserved. Insert Information Protection Policy Classification from Slide 13

http://kerryosborne.oracle-guy.com/2013/07/sql-translation-framework/

Kerry Osborne's Oracle Blog

Just another Oracle blog

HOME ABOUT DISCLAIMER ORACLE WHITEPAPERS / PRESENTATIONS

« 12c – New SQL_ID Calculation

12c – parallel_degree_level (control for auto DOP) »

SQL Translation Framework

July 13, 2013, 5:15 pm

My favorite new Oracle Database 12c feature is the SQL Translation Framework. The feature grew out of SQL Developer's ability to translate SQL from non-Oracle RDBMS's. For example, there is a pre-built Sybase ASE translation package that is designed to translate the Sybase dialect of SQL into Oracle SQL dialect. So that's what the feature is designed for. But the developers decided to move it to the database and to allow us to write our own translations which opens up a whole world of possibilities.

The first thought that occurred to me when I saw this feature listed in the 12c New Features doc, was that I might be able to use it to fix badly written SQL behind the scenes. I've written and talked quite a bit about using hint based mechanisms (Outlines, SQL Profiles, Baselines, and SQL Patches) to alter execution plans without having to change the code. Those technique work great most of the time, but there are cases where hints alone can't fix the problem. In some cases it is necessary to change the SQL statement text to get the desired results. And the SQL Translation Framework gives us the tool kit we need to do just that. And by the way, although I do have a tendency to use Oracle features for purposes for which they were not originally intended, in this case, I think the developers knew full well that the features could be used to address performance issues by re-writing SQL. As proof, here is a snippet from the 12c Release 1 Migration guide.

> In addition to translating non-Oracle SQL statements, the SQL Translation Framework can also be used to substitute an Oracle SQL statement with another Oracle statement to address a semantic or a performance issue. In this way, you can address an application issue without patching the client application.

So let's dive in. There are two main components to the framework. The first is a pl/sql package to programmatically translate code (also called the Translator in the docs). The second is a set of maps for individual SQL statements (this is called a SQL Translation Profile). There are a couple of requirements to use this feature.

 Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

```
SQLTranslationProfile
---------------------------------------------------------------------------------------------------------------------------------------------------------------
<SQLTranslationProfile ForeignSQLSyntax="TRUE" TranslateNewSQL="TRUE" RaiseTranslationError="FALSE" LogTranslationError="FALSE" TraceTranslation="FALSE"
  <SQLTranslations>
    <SQLTranslation Enabled="TRUE">
      <SQLText>select 1 from dual</SQLText>
      <TranslatedText>select 2 from dual</TranslatedText>
    </SQLTranslation>
    <SQLTranslation Enabled="TRUE">
      <SQLText>select 1 from hr.countries</SQLText>
      <TranslatedText>select count(*) from hr.countries</TranslatedText>
    </SQLTranslation>
    <SQLTranslation Enabled="TRUE">
      <SQLText>select count(*) from countries</SQLText>
      <TranslatedText>select count(*) from jobs</TranslatedText>
    </SQLTranslation>
    <SQLTranslation Enabled="TRUE">
      <SQLText>select count(*) from hr.countries</SQLText>
      <TranslatedText>select count(*) from hr.jobs</TranslatedText>
    </SQLTranslation>
  </SQLTranslations>
  <ErrorTranslations/>
</SQLTranslationProfile>

SYS@LAB1211> select 1 from dual;

         2
----------
         2
```


Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

12 Things

1. It is Shipping
2. Hello EM Express
3. Multitenant Database Option
4. Automatic Data Optimization
5. PL/SQL from SQL
6. Improved Defaults
7. Increased Size Limits for VARCHAR2
8. Easy Top-N and Pagination Queries
9. Row Pattern Matching
10. Partitioning Improvements
11. Temporary UNDO
12. SQL Translation Framework

Much, Much more. Examples…

- Application Continuity & Transaction Guard
- Flex ASM
- Global Data Services
- Cross-Platform Backup and Restore
- Data Guard Far Sync
- Data Redaction
- Temporal Support (Transaction Time Temporal / Valid Time Temporal)
- Adaptive Execution Plans
- Enhanced Statistics
- Queryable Patch Inventory
- Oracle Intelligent Storage Protocol (OISP)

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.
Insert Information Protection Policy Classification from Slide 13

The preceding is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle’s products remains at the sole discretion of Oracle.

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

Hardware and Software
Engineered to Work Together

ORACLE