

Redefining Tables Online Without Surprises

Nelson Calero

Oracle Open World – October 2017

Pythian



ORACLE
OPEN
WORLD

About me

- Database Consultant at Pythian since 2014
- Working with Oracle tools and Linux environments since 1996
- DBA Oracle (2001) & MySQL (2005)
- Co-founder and President of the Oracle user Group of Uruguay (2009)
- LAOUC Director of events (2013)

- Computer Engineer (1998)
- Oracle ACE (2014), Oracle ACE Director (2017)
- Oracle Certified Professional DBA 10g/11g/12c (2008)
- Amazon Solutions Architect – Associate since (2016)
- Oracle University Instructor (2011)
- Blogger and speaker: Oracle Open World, Collaborate, OTN Tour, Regional conferences



ORACLE®

Certified Professional

Oracle Database 12c
Administrator

ORACLE®

Certified Expert

Oracle Database 12c
Performance Management
and Tuning



Solutions Architect - Associate



<http://www.linkedin.com/in/ncalero>



@ncalerouy

Pythian overview

- 20 Years of data infrastructure management consulting
- 250+ Top brands
- 11800+ systems under management
- 400+ employees in 35 countries
- 10 Oracle ACEs, 4 ACED,
3 OakTable members, 2 OCM, 6 Microsoft MVPs,
1 Cloudera Champion of Big Data,
AWS Certified Solutions Architect – 2 Professional, 12 Associate
- Oracle, Microsoft, MySQL, Hadoop, Cassandra, MongoDB, and more
- Infrastructure, Cloud, DevOps, and application expertise



500+ Technical Experts Helping Peers Globally

ORACLE[®]
ACE PROGRAM



ORACLE[®]
ACE Director



ORACLE[®]
ACE



ORACLE[®]
ACE Associate

3 Membership Tiers

- Oracle ACE Director
- Oracle ACE
- Oracle ACE Associate

bit.ly/OracleACEProgram

Connect:

✉ oracle-ace_ww@oracle.com

f [Facebook.com/oracleaces](https://www.facebook.com/oracleaces)

t [@oracleace](https://twitter.com/oracleace)



Nominate yourself or someone you know: acenomination.oracle.com

Online table redefinition

- Motivation
- Online DDL review
- dbms_redefinition examples
 - Changes in different versions
 - Known restrictions
 - Problems found
- Real life usage to compress a VLDB

Motivation

- DDL operations have different optimizations to make it faster
 - DDL locks are held (shared or exclusive) while operation runs
 - Larger tables with some DDL that do not implement optimizations will keep locks longer
- DML automatically grab locks to keep consistency
 - Row Locks (TX) and Table Locks (TX) using different modes: RS, RX, S, SRX, X
 - Conflicting locks will delay running transactions
 - More visible with large tables and heavy concurrency
- Some example DDLs:
 - Change table columns: add/drop/modify attributes (name, datatype, length, compression, etc.)
 - Move a table to a different tablespace
 - Partition a non-partitioned table (and vice-versa)
 - Compress an uncompressed table (and vice-versa)
 - Reorganize data to change physical layout
 - rows per block (PCTFREE), Ordering (clustering factor)

Optimized DDL operations - example

- Adding a column to a table:
ALTER TABLE t1 **ADD C1 number**;
 - Metadata updated, no change made to existing rows
- Including now a default value:
ALTER TABLE t1 **ADD C1 number default 1**;
 - Before 12c: existing rows are updated adding the new value
 - Since 12c: metadata updated, no changes to existing rows
 - Restrictions with LOBs, IOTs, temporary, MVs, VPD, encryption, queue
- NOT NULL instead of NULL:
ALTER TABLE t1 **ADD C1 number default 1 not null** ;
 - Before 11g – existing rows are updated adding the new value
 - Since 11g – metadata updated, no changes to existing rows

https://docs.oracle.com/database/121/SQLRF/statements_3001.htm#CIHGHAJG

<http://www.oracle.com/technetwork/articles/database/ddl-optimization-in-odb12c-2331068.html>

Example

Terminal 1:

```
watch -d --interval=1 "sqlplus -s / as sysdba @locks | tee -a watch.log"
```

Terminal 2:

- Generate DML (ex: insert a record without commit)

Terminal 3:

- Play with DDL
- Example: alter table add / drop column, index, constraints

Today's focus

- We are looking to modify the structure of a table without blocking other users' transactions
 - Also known as Online DDL
 - Highlighted as HA features (as no application downtime is required)
- Several operations already implemented as online in Oracle if conditions are met (details to follow)
- Some only available on Enterprise Edition (EE)

Online operations in Oracle

- Only in Enterprise Edition
 - Online index rebuild – since 8i
 - Online index-organized table organization (alter table move)
 - Online table redefinition (dbms_redefinition) – since 9i
 - Online partition/sub-partition move – since 12.1
 - Online table move – since 12.2
 - Online SPLIT Partition and Subpartition – since 12.2
- Restrictions apply per functionality and version
 - Example: bitmaps, IOTs, Parallel DML, column datatypes
- Note the 'online' clause needs to be added to the SQL statement

```
alter index T1_IDX rebuild online;  
alter table T1 move online tablespace NEW_TS;
```

Online DDL – poor's man approach

- Create empty table with the changes needed
- Copy source rows to modified table
- Copy changes received since operation started
 - repeat as needed (controlling amount of data moved each time)
- lock table in exclusive mode
 - interchange old and new table (renaming both)
- Goal is to minimize the exclusive lock duration
- Dependent objects needs to be created manually
 - Constraints, indices, triggers – pre-creating them disabled adds run-time overhead
 - Names should be interchanged with original ones
- Several examples from open source databases
 - pt-online-schema-change (MySQL - Percona), osc_cli (MySQL – Facebook)
 - pg_repack (PostgreSQL)

Online operations in Oracle

- Obvious thing to remark: RTFM !!
 - Features change in versions
 - Restrictions for each operation based on structure
 - Patches to fix known bugs
- Example: ALTER TABLE <table> MOVE ...
 - On 12.2 the ONLINE option allows DML against the <table> while it runs
 - dbms_redefinition is needed before 12.2
 - Indexes are marked unusable because ROWID changes
 - ORA-1502 reported by queries using those indexes
 - LOB columns are not moved by default
 - we need to include it explicitly if we want to move them with the table

Example: moving a single partition

- In Oracle 12.1:

```
alter table test.T1 move partition P4 online update indexes;
```

- Bug 'Inconsistent Datatypes: Expected NUMBER (Doc ID 2028583.1)' on 12.1.0.2.170117

- needs one-off patch 20703000

- Physical attributes gets copied

- if we plan to change them, better to do it before moving:

```
alter table test.T1 modify default attributes STORAGE(INITIAL  
1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645);
```

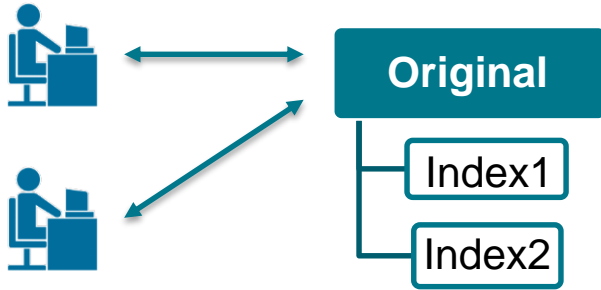
```
alter table test.T1 move partition P4 online update indexes;
```

- Pre-12.1: only using dbms_redefinition

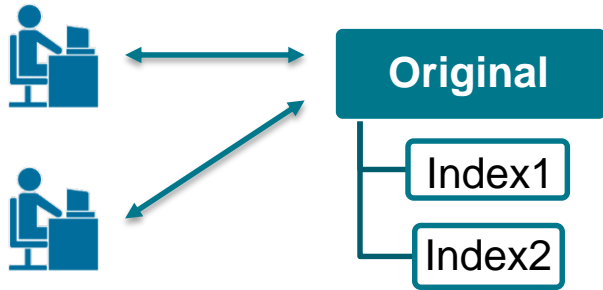
dbms_redefinition

- Available since version 9i to allow online DDL
- A snapshot log is automatically created saving changes while data is copied
- Resync as many times as needed (good to resume failures)
- Minimum time needed for the final step to rename tables
- Needs 2x original space – a temporal (interim) table is used
 - Don't forget increasing redolog activity, archivelog space and standby transport
- Simplified in 12c for some use cases
 - One procedure does all the work, but is less flexible

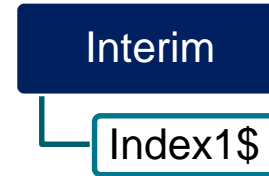
dbms_redefinition - workflow



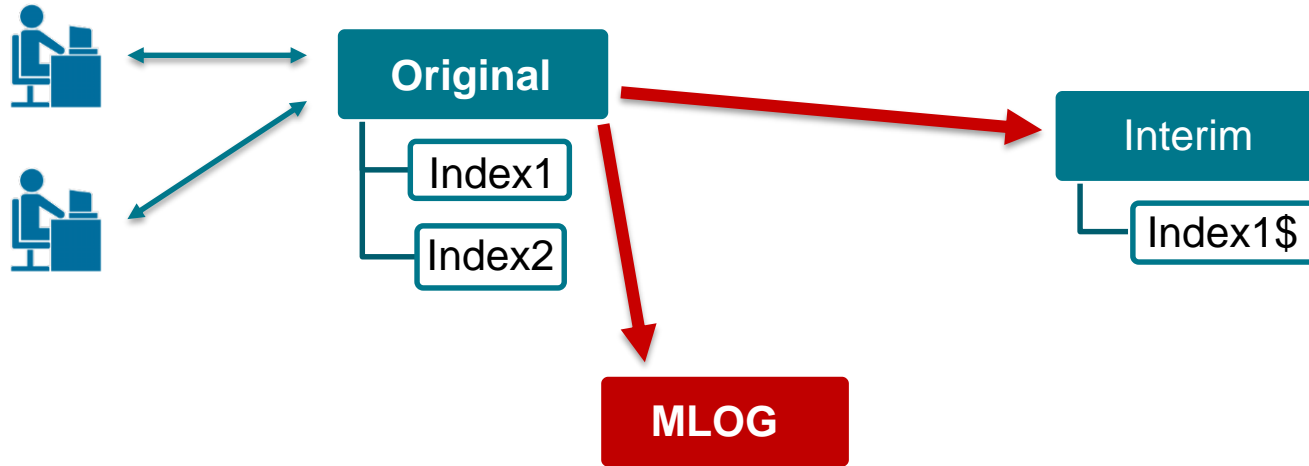
dbms_redefinition - workflow



1 create interim table

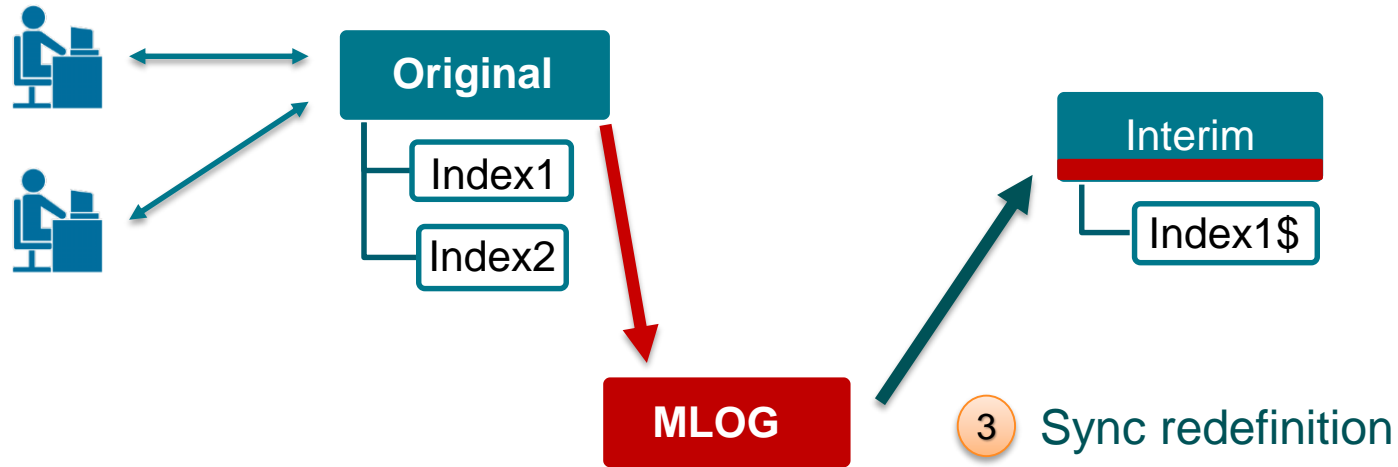


dbms_redefinition - workflow



2 Start redefinition

dbms_redefinition - workflow

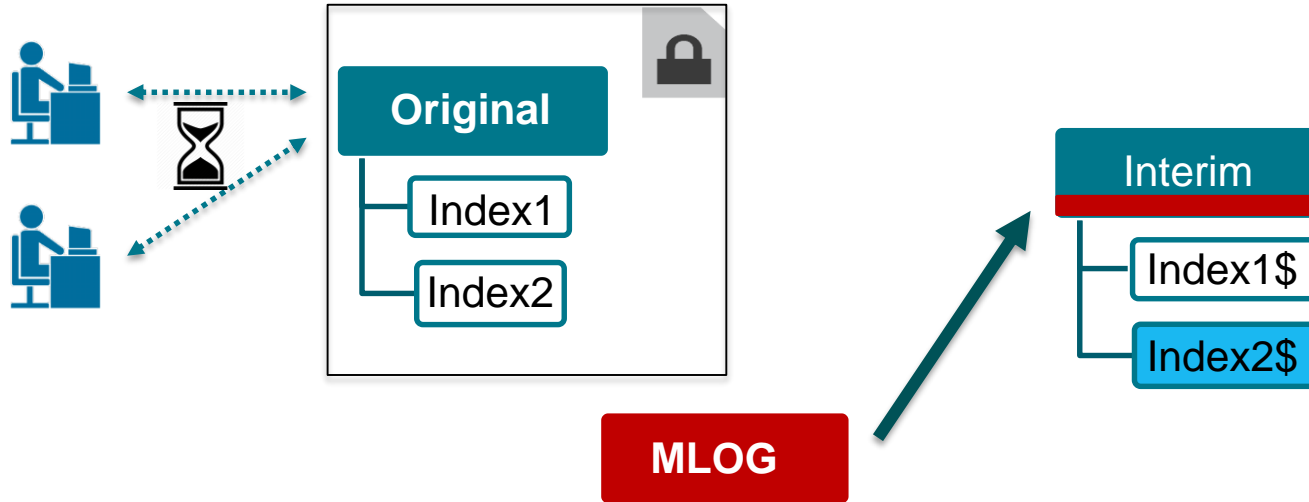


dbms_redefinition - workflow



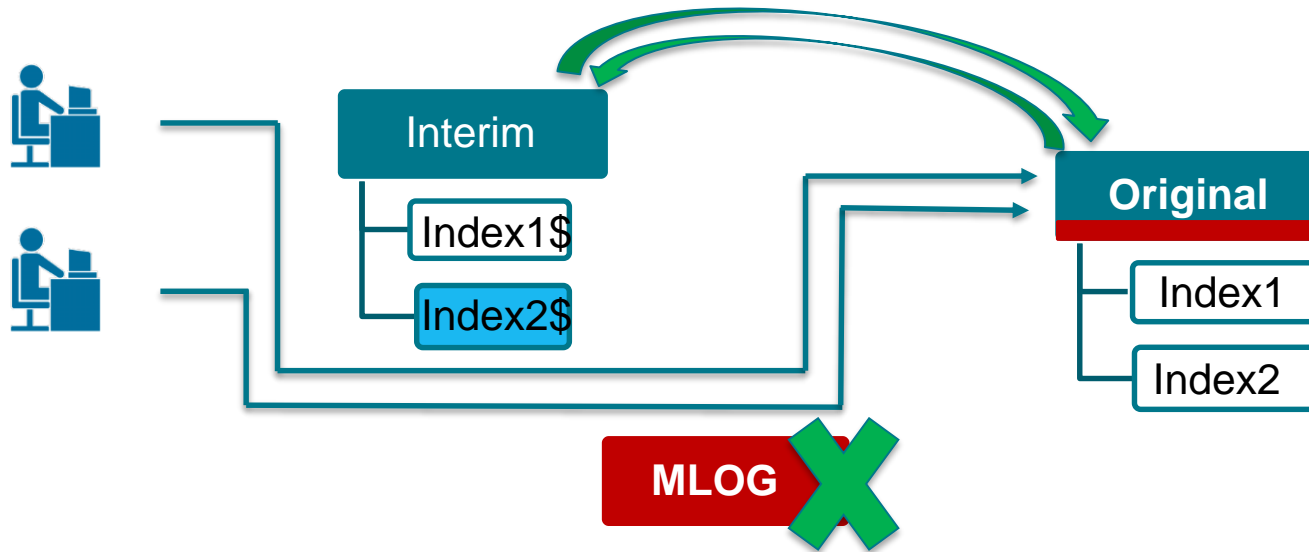
dbms_redefinition - workflow

5 Finish redefinition (1/2)

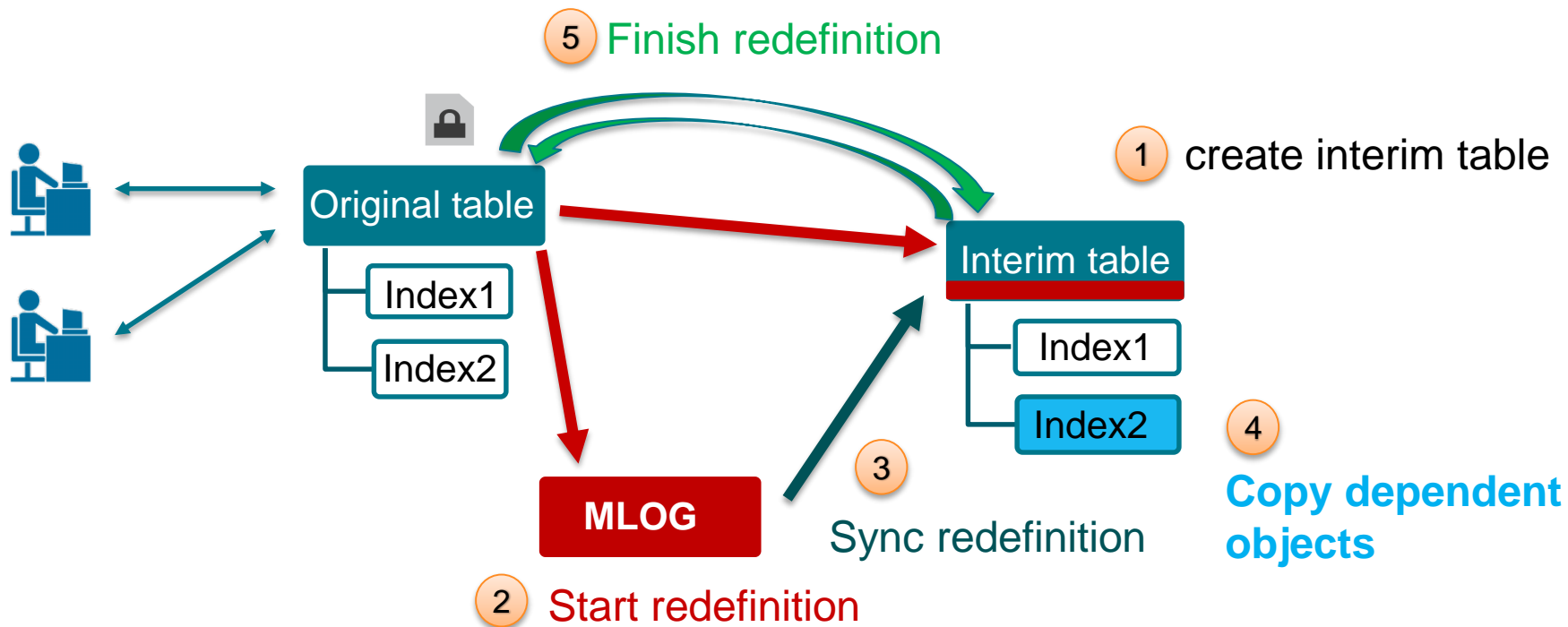


dbms_redefinition - workflow

5 Finish redefinition (2/2)



dbms_redefinition – complete workflow



dbms_redefinition – 12c simple usage

```
DBMS_REDEFINITION.REDEF_TABLE (
  uname          IN  VARCHAR2,
  tname          IN  VARCHAR2,
  table_compression_type  IN  VARCHAR2 := NULL,
  table_part_tablespace  IN  VARCHAR2 := NULL,
  index_key_compression_type  IN  VARCHAR2 := NULL,
  index_tablespace      IN  VARCHAR2 := NULL,
  lob_compression_type  IN  VARCHAR2 := NULL,
  lob_tablespace        IN  VARCHAR2 := NULL,
  lob_store_as         IN  VARCHAR2 := NULL)
```

We can combine all parameters in a single execution to:

- compress table and all LOBs
- change tablespace for table (or all partitions), all indexes and all LOBs
- change LOB to store as SECUREFILE or BASICFILE

dbms_redefinition – 12c example

Moving an entire table changing table, indexes and LOBs attributes:

```
begin
  DBMS_REDEFINITION.REDEF_TABLE (
    uname           => 'TEST',
    tname           => 'T1',
    table_compression_type => 'ROW STORE COMPRESS ADVANCED',
    table_part_tablespace => 'TEST_COMP',
    index_key_compression_type => 'COMPRESS ADVANCED LOW',
    index_tablespace   => 'TEST_IDX_COMP',
    lob_compression_type   => 'COMPRESS HIGH',
    lob_tablespace      => 'TEST_COMP',
    lob_store_as        => 'SECUREFILE');
end;
```

NOTE: Table and index owner (TEST) needs quota privilege on new tablespace TEST_COMP.

dbms_redefinition – 12c example



Demo – example1.sql

Compressing a partitioned table, doing some validations and discovering some issues:

- hidden columns created when using ROWID
- NOVALIDATE constraints after redefinition (BUG 4396234)
- see concurrency effect

NOTE: All examples used in this presentation are in dropbox

dbms_redefinition - restrictions

- Same schema
- LONG and LONG RAW columns cannot be redefined
 - cannot be moved manually either (alter table .. move)
 - LONG columns must be converted to CLOBS
 - LONG RAW columns must be converted to BLOBS
 - needs application testing
- NOTE: BFILE columns can be redefined online starting on 12.2
- Known bugs with one-off patch available (11.2.0.4 example):
 - 18889295 - ORA-600 [4511] kdbchk: row locked by non-existent transaction on OLTP compressed table
 - 21887221 - Primary Key Constraint index changes after redefinition when a primary key index is a subset of another normal index being present on the same table

dbms_redefinition – long approach

- 1) Manually create the interim table with the changes we want
- 2) Validates if table can be redefined online
DBMS_REDEFINITION.**can_redef_table** (owner, seg, flag, part);
- 3) Start Redefinition
DBMS_REDEFINITION.**start_redef_table** (owner, seg, interim, part);
- 4) Apply changes captured while previous step ran
DBMS_REDEFINITION.**sync_interim_table** (owner, seg, interim);
- 5) Copy dependent objects
DBMS_REDEFINITION.**copy_table_dependents** (owner, seg, interim, errors);
- 6) Finish – exchange table names (p_seg and p_int_tab)
DBMS_REDEFINITION.**finish_redef_table** (owner, seg, interim, part);

dbms_redefinition – short or long usage?

Old approach (with more steps) needed in 12c for:

- structural changes
- moving only some partitions/lobs/indexes (not all)
 - Not recommended for structural changes

start_redef_table offers some extra features:

- Reorder rows using order by clause (*orderby_cols* parameter)
- Copy only some columns, or create new combining functions to existing ones (*col_mapping* parameter)

NOTE: data is copied without generating UNDO (APPEND hint is used in the INSERT statement)

dbms_redefinition – COPY_TABLE_DEPENDENTS

- Clone indexes, privileges, triggers, constraints and statistics
- Triggers and referential integrity constraints are created disabled, enabled automatically after redefinition completes
- All cloned objects are renamed to the original names when redefinition completes – interim table is the original table at that point, having internal names for cloned objects

History:

- Added in 10g
- Since 11.1 dependent objects are not invalidated (PL/SQL, views, synonyms etc.)

dbms_redefinition – new in 12.2

- `V$ONLINE_REDEF`

Progress reported by `START_REDEF_TABLE` and `COPY_TABLE_DEPENDENTS`

- Restarting redefinition after failures

`DBA_REDEFINITION_STATUS` view includes new columns to identify the object and error, allowing to resolve error and resume failed operation

Example: tablespace full when executing `SYNC`

- Rolling back redefinition after it was completed

New parameter `ENABLE_ROLLBACK` in `START_REDEF_TABLE`

New procedures `ABORT_ROLLBACK` and `ROLLBACK`

After completing a redefinition with rollback enabled, `MLOG` is not dropped. We can `SYNC` after that to keep old source table (now interim) updated, until we decide to rollback or abort it

- `BFILE` datatype support
- `SYNC_INTERIM_TABLE` can incremental refresh dependent MVs on the interim table

dbms_redefinition – troubleshooting

- Errors are recorded in *dba_redefinition_errors*
 - From redefinition step (first) and copy dependent objects procedure
 - Nothing from compact 12c REDEF_TABLE – it works or aborts
- Manual procedure allows to fix errors and resume
- To cancel a non completed redef

```
DBMS_REDEFINITION.ABORT_REDEF_TABLE
```
- If in doubt: SQL trace of the session running redefinition

dbms_redefinition – more examples

Demo

- example2.sql : 11g approach to move a single partition
 - Using DBMS_METADATA.GET_DDL to generate interim table
 - See concurrency effect and data movement
- example3.sql – redefining a table having hidden columns

dbms_redefinition – troubleshooting examples

Demo

- example4.sql: solving errors when copying constraints
 - Interim table have NOT NULL columns and we try to copy constraints using *copy_table_dependents*
 - Side effect after resolving it with final table structure
- example5.sql: solving error when copying index
 - Source table has a bitmap index, using *copy_table_dependents* gives error
 - Using *register_dependent_object* to redefine table

dbms_redefinition – Summary

- Redefine using PK if available
 - Avoid hidden column added when using ROWID (known issue as per MOS 459762.1)
- When creating Interim table, columns should be *nullable* to avoid errors when copying dependent
 - Use `dbms_metadata.get_ddl` to automate it
- NOT NULL constraints are left NOVALIDATE after clone
 - Bug:4396234 - we need to modify them after redefinition completes
- To exclude a particular objects from COPY_TABLE_DEPENDENTS:
 - Use REGISTER_DEPENDENT_OBJECT before
 - object must be manually pre-created, and will be renamed at the end
- Optimizer stats can be copied – check if that makes sense for you
 - Clustering factor change - data is defragmented, orderby clause can be used, pctfree can change, etc.

Real life approach to compress VLDB

Procedure outline

- By object or tablespace approach?
- Identify objects to exclude (matching known restrictions)
 - both for dbms_redefinition and compression
- Plan space needs – 2x for each table (and index)
- Automate whenever possible – dropping objects too?
- Able to split indexes from table maintenance?
 - Needs application usage planning, as it could affect SQL using them

Facts

- 30Tb database, 350 tablespaces, 2000 datafiles
- Combination of all datatypes and structures
 - IOTs, partitioned, basicfile LOBs, LONG
- More than 24hs needed to redefine 1Tb tables
- Different space savings found – floor around 2x

THANK YOU

Questions?

✉ calero@pythian.com

🐦 [@ncalerouy](https://twitter.com/ncalerouy)

🌐 <http://www.linkedin.com/in/ncalero>

References - documentation

- DBMS_REDEFINITION - 12.2 Database PL/SQL Packages and Types Reference
https://docs.oracle.com/database/122/ARPLS/DBMS_REDEFINITION.htm#ARPLS042
- Automatic Locks in DML Operations - 12.2 Database SQL Language Reference
<https://docs.oracle.com/database/122/SQLRF/Automatic-Locks-in-DML-Operations.htm#SQLRF55502>
- Managing Tables - 12.2 Database Administrator's Guide
<https://docs.oracle.com/database/122/ADMIN/managing-tables.htm>
- All examples in this presentation are in dropbox, <http://bit.ly/2w71qxb>



dbms_redefinition – common issues

1) ORA-12093: invalid interim table "OWNER"."TABLE"

- Intermediate table named in the “redef_table” is not created yet

2) ORA-12089: cannot online redefine table "OWNER"."TABLE" with no primary key

- A table with no PK was tried to be redefined without using the ROWID option

3) ORA-23539: table "OWNER"."NAME" currently being redefined

- Table redefinition finished with errors and a new “redef_table” was run

If data was already copied and failure was during copy dependents steps, we don't need to drop the table and start again - we can keep the already copied data, resolve errors, sync and repeat copy dependants.

dbms_redefinition – common issues (2)

4) ORA-00997: illegal use of LONG datatype

- Columns LONG, LONG RAW and BFILE cannot be redefined.

https://docs.oracle.com/cd/B28359_01/server.111/b28310/tables007.htm#ADMIN11674

5) ORA-23541: tables do not match tables used while defining the redefinition

- When COPY_TABLE_DEPENDENTS is used for a single partition, it tries to copy partitioned table objects when target is not partitioned and only an exchange partition will be executed

Solution:

- copying dependents should not be used when moving single partitions

dbms_redefinition – unexpected issues

6) ORA-25965: fact table must be included in the from clause

- This is because index DDL generated by redefinition contains reference to source table without being changed to the new name

SQL reported in DBA_REDEFINITION_ERRORS.DDL_TXT:

```
CREATE BITMAP INDEX "TEST"."TMP$$_IDX_JOIN0"  
                ON "TEST"."T3_RDEF" ("T97912"."CODE")  
FROM "TEST"."T2" "T97912", "TEST"."T3" "T97911"  
WHERE "T97911"."ID"="T97912"."BUCK_ID"
```

Solution:

- manually create index with the correct name before starting
- exclude index from redefinition using REGISTER_DEPENDENT_OBJECT

dbms_redefinition – unexpected issues (2)

7) ORA-14024: number of partitions of LOCAL index must equal that of the underlying table

- Interim table was created using dbms_metadata, but it got table and index creation with less partitions than the existing one (not updated with new partitions added)

Solution:

- Manually adjust interim table creation SQL adding missing partitions from existing index

8) ORA-01442: column to be modified to NOT NULL is already NOT NULL

- Intermediate table was created with not null columns

Solution:

- Remove not null constraint from interim table and retry

desc will show table not having not null constraints, but the check constraint is there and enabled

- 1) drop check constraints on interim tables
- 2) resync, retry copy dependent and finish redefinition
- 3) enable validate check constraints

Caused by a known bug:4396234 ET10.2OREDEF: NULLABLE COL OF *_TAB_COLUMNS TABLE NOT UPDATED AFTER ONLINE REDEF