

Make IT

2022

Hey DBMS_COMPARISON! Are my Tables in sync?

Christian Gohmann

Make IT 2022

2nd of June 2022

trivadis

Part of Accenture

2 HALLO, GRÜEZI, HI!



CHRISTIAN GOHMANN

- Principal Consultant, Instructor and Tool Owner of db*BACKUP
- Working with Oracle since 2006
- Focused on High Availability Solutions, Migration Projects, Backup & Recovery and Cloud Technologies
- Oracle ACE





1994

FOUNDATION

> 22000

TRAINING PARTICIPANTS
PER YEAR



21000



ORDERS
PER YEAR



> 750

CUSTOMERS PER
YEAR



123 Mio.

TURNOVER

> 300

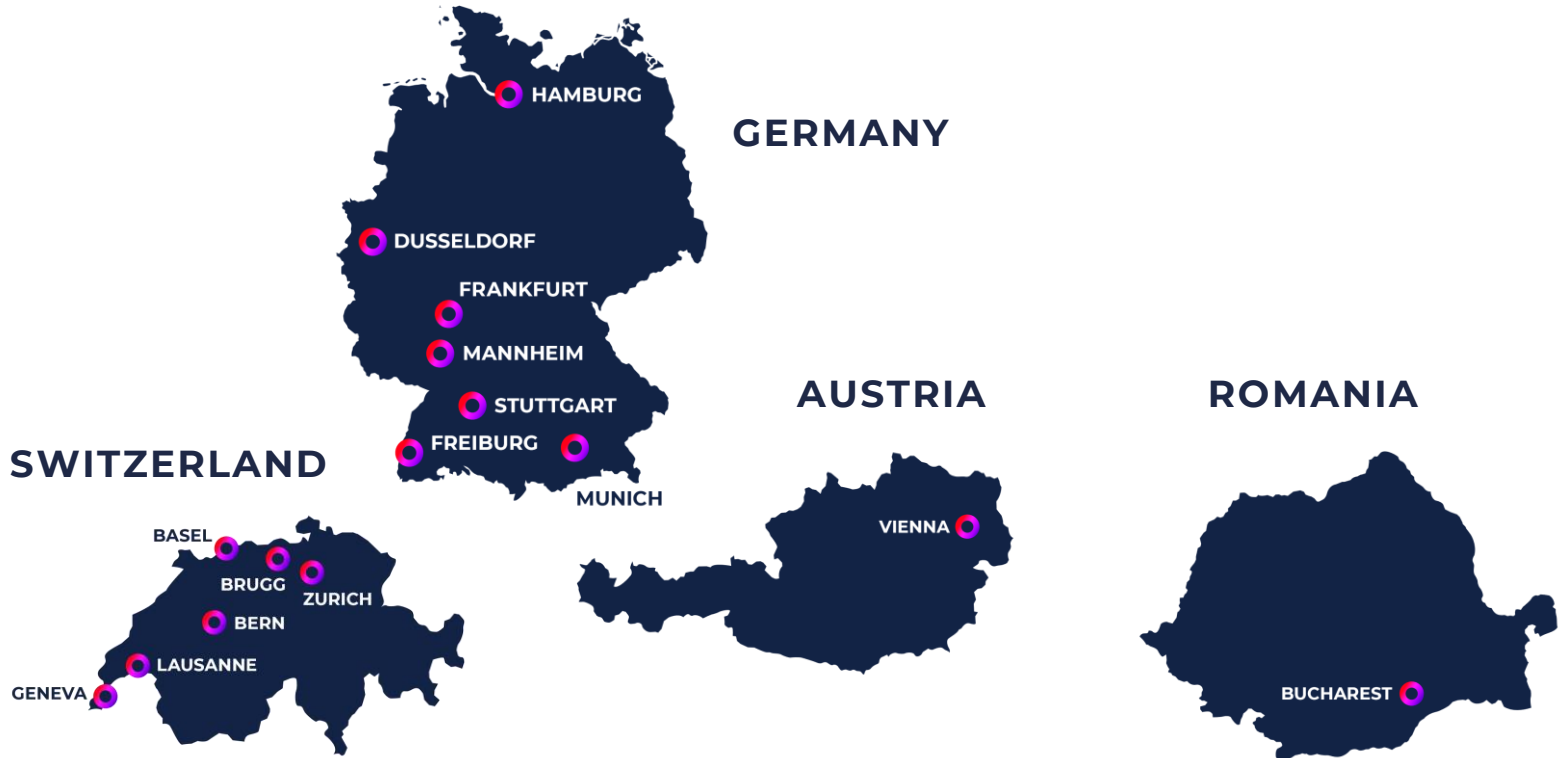
ACTIVE SLAs



trivadis

Part of Accenture

4 OUR WORKSPACES



5 AGENDA

1. Introduction

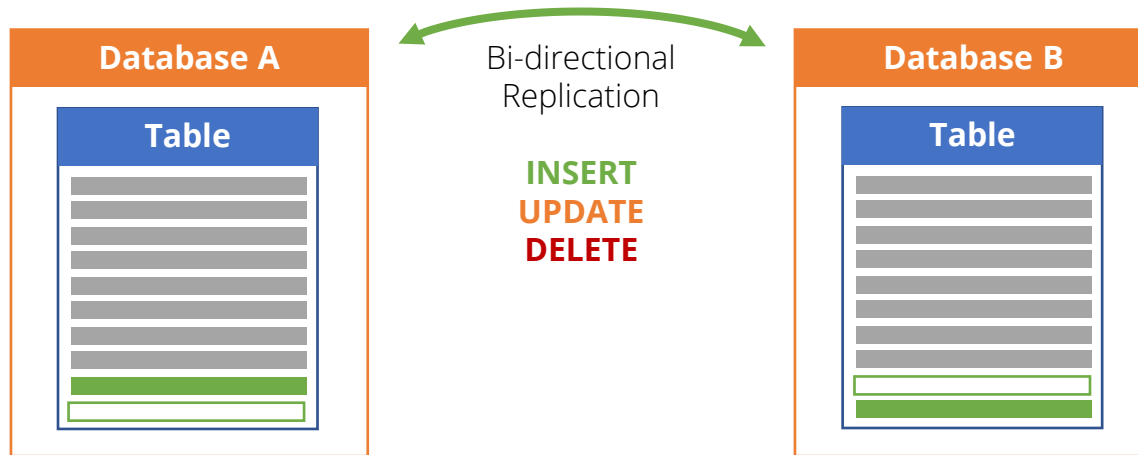
2. DBMS_COMPARISON

3. Execution

INTRODUCTION

7 MOTIVATION

- Data is replicated between different databases using **Logical Replication**
 - For example, Data Guard Logical Standby, GoldenGate, third-party products
 - **Uni- or bi-directional replication**
- If an error or collision occurs, the data **might get out-of-sync** (data deviation)



8 SOLUTION?

- Evaluate your Logical Replication configuration
 - Check error and collision handling
 - Improve monitoring
 - Compare the data between source and target regularly
- Identify data deviations and correct only the affected data
 - Avoid complete data refreshes to minimize the downtime

DBMS_COMPARISON

10 DBMS_COMPARISON

- **PL/SQL package**, introduced with Oracle 11g Release 1
- **Scans the contents of two tables** for differences (this is called a comparison)
 - Either in the same database or in two different databases
 - A database link is used to connect between source and target database
- The scan can **include the affected rows** or a general state of the synchronization
 - Getting the affected rows require more time
- **Differences can be converged** to get a synchronized state
 - Deactivate your Logical Replication before doing this
 - Only rows with differences are synchronized using INSERT, UPDATE, DELETE statements



A table which is part of a replication is also known as a **shared database object**.

11 SUPPORTED OBJECT TYPES

- Tables
- Single-Table Views
- Materialized Views
- Synonyms for above object types



A comparison of different object type is possible, e.g., table with materialized view.

12 (UN)SUPPORTED DATA TYPES

- Not all column data types are supported for a comparison

Supported

- BINARY_DOUBLE
- BINARY_FLOAT
- CHAR
- DATE
- FLOAT
- INTERVAL DAY TO SECOND
- INTERVAL YEAR TO MONTH
- NCHAR
- NUMBER
- NVARCHAR2
- RAW
- TIMESTAMP
- TIMESTAMP WITH LOCAL TIME ZONE
- TIMESTAMP WITH TIME ZONE
- VARCHAR2

Unsupported

- BFILE
- BLOB
- CLOB
- LONG
- LONG RAW
- NCLOB
- Oracle-supplied types (including any types, XML types, spatial types, and media types)
- ROWID
- UROWID
- User-defined types (including object types, REFs, varrays, and nested tables)



Exclude columns with unsupported data types to allow a partial comparison.

13 GENERAL REQUIREMENTS

- Character set must be the same between source and target
- Table shape (columns, data type) must be equal
 - Parameter `column_list` be used if the shape is different
- For the data type `TIMESTAMP WITH LOCAL TIME ZONE (TSLTZ)`, the time zone version must be equal



The remote database can be also 10g Release 1 and higher.

14 INDEX REQUIREMENTS 1/2

- To compare rows between source and target table, each row has to be uniquely identified
- For all scan modes, the database objects must have one of the following types of indexes
 - Single column index on NUMBER, TIMESTAMP, INTERVAL or DATE data type
 - Composite index with only columns of the same data type family
- One of the following index types can be used to achieve this
 - Primary key index
 - Unique index with one or more NOT NULL columns



The columns of the index have to be part of the comparison.

15 INDEX REQUIREMENTS 2/2

- Use parameters INDEX_SCHEMA_NAME and INDEX_NAME to select a specific index
 - Uniqueness is not checked
- If no index can be found or is not suitable, ORA-23676 is raised

ORA-23676: no eligible index on local table "SOURCE_USER"."EMPLOYEES"

16 SCAN MODES

- Tables can be scanned completely or partially
- Four different scan modes can be used
 - Parameter `scan_mode` in `DBMS_COMPARISON.CREATE_COMPARISON`

FULL (CMP_SCAN_MODE_FULL) (default)

- Full database objects comparison

RANDOM (CMP_SCAN_MODE_RANDOM)

- A random portion of data is compared

CYCLIC (CMP_SCAN_MODE_CYCLIC)

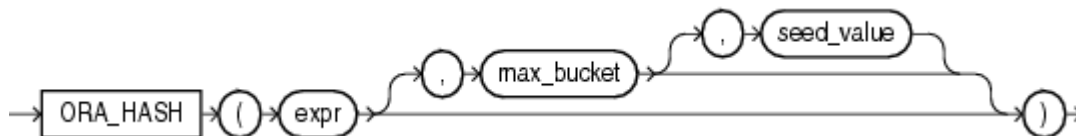
- A different portion (starting after the last run) of the object is compared

CUSTOM (CMP_SCAN_MODE_CUSTOM)

- The user can define the range of data to compare

17 ORA_HASH

- DBMS_COMPARISON uses the **ORA_HASH function** to generate a hash for each row
 - **32-bit hash algorithm** with $2^{32} - 1$ different buckets



- This function is the reason for the data type restrictions
- All compared columns are checked with **nested ORA_HASH calls**

```
ora_hash(NVL(to_char(s."EMPLOYEE_ID"), 'ORA$STREAMS$NV'), 4294967295,  
ora_hash(NVL((s."LAST_NAME"), 'ORA$STREAMS$NV'), 4294967295, 0))
```

- **Hash collisions can happen**, but with the additional check of the primary key, they are very unlikely

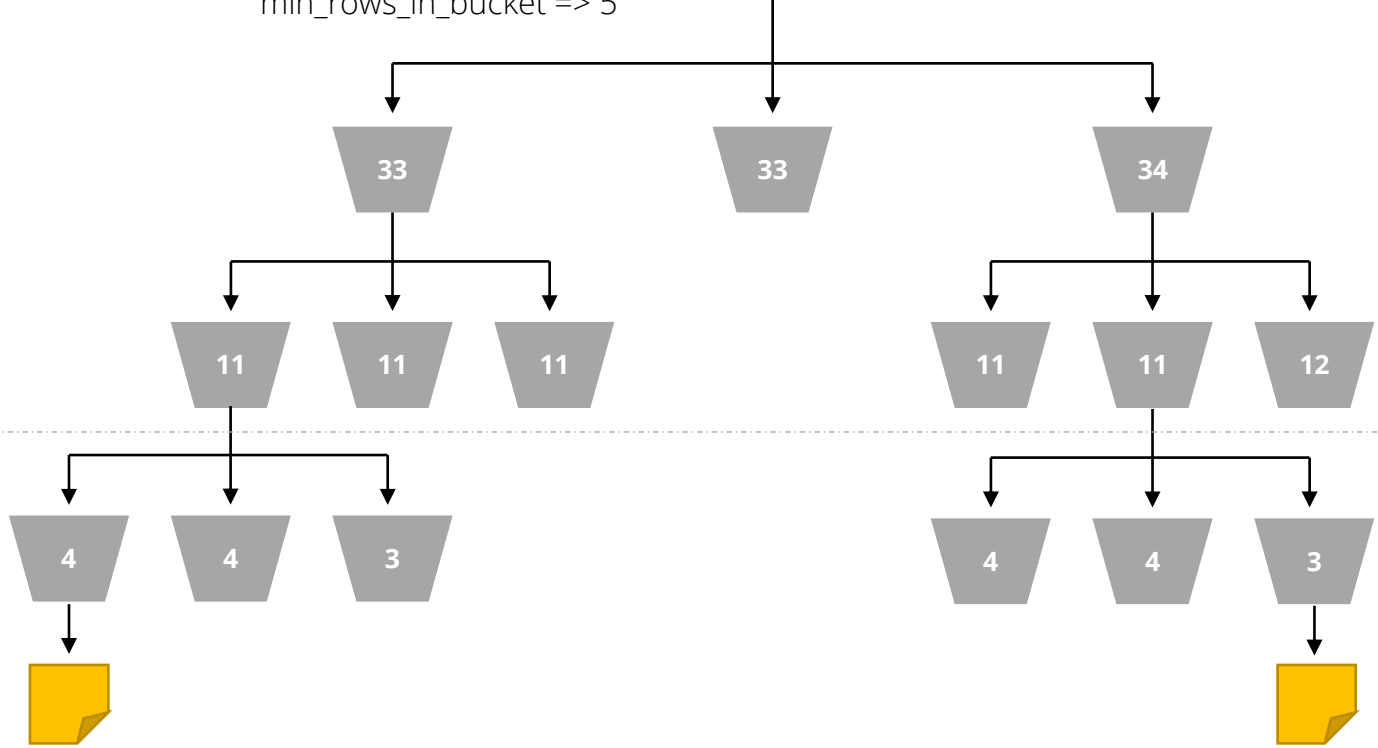


The result of an ORA_HASH call is used as seed value for the previous call.

18 SCANS & BUCKETS 1/2

Parameters
num_buckets => 3
min_rows_in_bucket => 5

EMPLOYEES 100 Rows



Minimum number of rows per bucket reached

19 SCANS & BUCKETS 2/2

- Query **DBA_COMPARISON_SCAN_SUMMARY** to get an overview of the scans

```
SQL> SELECT scan_id, parent_scan_id, status, current_dif_count, count_rows
       FROM dba_comparison_scan_summary
       WHERE comparison_name = 'MY_COMPARISON'
       ORDER BY last_update_time;
```

SCAN_ID	PARENT_SCAN_ID	STATUS	CURRENT_DIF_COUNT	COUNT_ROWS
61		BUCKET DIF	3	100
62	61	BUCKET DIF	1	33
64	62	BUCKET DIF	1	11
65	64	ROW DIF	1	4
63	61	BUCKET DIF	2	34
66	63	BUCKET DIF	2	11
67	66	ROW DIF	2	3

Root scan



Only scans with differences are visible.

EXECUTION

21 CREATE COMPARISON

- Define a comparison template

```
SQL> BEGIN
  DBMS_COMPARISON.CREATE_COMPARISON (
    comparison_name => 'MY_COMPARISON',
    schema_name => 'SOURCE_USER',
    object_name => 'EMPLOYEES',
    dblink_name => NULL,
    remote_schema_name => 'TARGET_USER',
    remote_object_name => 'EMPLOYEES',
    column_list => 'employee_id, last_name',
    scan_mode => DBMS_COMPARISON.CMP_SCAN_MODE_FULL,
    max_num_buckets => 3,
    min_rows_in_bucket => 100
  );
END;
/
```

NULL for a local comparison

22 COMPARE 1/2

- Compare the data of the source and target object

```
SQL> DECLARE
  v_scan_info    DBMS_COMPARISON.COMPARISON_TYPE;
  v_result       BOOLEAN;
BEGIN
  v_result := DBMS_COMPARISON.COMPARE (
    comparison_name => 'MY_COMPARISON',
    scan_info => v_scan_info,
    min_value => NULL, max_value => NULL,
    perform_row_dif => TRUE
  );

  IF v_result THEN
    DBMS_OUTPUT.PUT_LINE('Both tables are in sync');
  ELSE
    DBMS_OUTPUT.PUT_LINE('Differences found in scan with ID ' || v_scan_info.SCAN_ID);
  END IF;
END;
/
```

Define range
for custom scan
mode

If set to TRUE, each
row is compared
individually

23 COMPARE 2/2

- Query the comparison results to get the differences

```
SQL> SELECT scan_id, parent_scan_id, status, current_dif_count, count_rows
       FROM dba_comparison_scan_summary
       WHERE comparison_name = 'MY_COMPARISON'
       ORDER BY last_update_time;
```

SCAN_ID	PARENT_SCAN_ID	STATUS	CURRENT_DIF_COUNT	COUNT_ROWS
12		BUCKET DIF	2	107
13	12	ROW DIF	1	36
14	12	ROW DIF	1	35

```
SQL> SELECT scan_id, index_value, local_rowid, remote_rowid, status
       FROM dba_comparison_row_dif
       WHERE comparison_name = 'MY_COMPARISON';
```

SCAN_ID	INDEX_VALU	LOCAL_ROWID	REMOTE_ROWID	STA
13	100	AAATFUAAALAAAjXLAAA	AAATFVAALAAAjXTAAA	DIF
14	156	AAATFUAAALAAAjXLAA4	AAATFVAALAAAjXTAA4	DIF

24 CONVERGE

- If wished, synchronize data between source and target object

```
SQL> DECLARE
  v_scan_info      DBMS_COMPARISON.COMPARISON_TYPE;
BEGIN
  DBMS_COMPARISON.CONVERGE (
    comparison_name => 'MY_COMPARISON',
    scan_id => 12,
    scan_info => v_scan_info,
    converge_options => DBMS_COMPARISON.CMP_CONVERGE_LOCAL_WINS,
    perform_commit => TRUE
  );

  DBMS_OUTPUT.PUT_LINE('Rows merged (local)..: ' || v_scan_info.LOC_ROWS_MERGED);
  DBMS_OUTPUT.PUT_LINE('Rows merged (remote)..: ' || v_scan_info.RMT_ROWS_MERGED);
  DBMS_OUTPUT.PUT_LINE('Rows deleted (local)..: ' || v_scan_info.LOC_ROWS_DELETED);
  DBMS_OUTPUT.PUT_LINE('Rows deleted (remote): ' || v_scan_info.RMT_ROWS_DELETED);
END;
/
```

Decide which
object wins.

If set to TRUE, a
commit is performed

25 RECHECK 1/2

- Use the RECHECK function to recheck differences in a specific scan
 - Or continue a partial scan

```
SQL> DECLARE
  v_result          BOOLEAN;
BEGIN
  v_result := DBMS_COMPARISON.RECHECK (
    comparison_name => 'MY_COMPARISON',
    scan_id => 12,
    perform_row_dif => TRUE
  );

  IF v_result THEN
    DBMS_OUTPUT.PUT_LINE('Both tables are in sync');
  ELSE
    DBMS_OUTPUT.PUT_LINE('Differences found');
  END IF;
END;
/
```

If set to TRUE, each row is compared individually

26 RECHECK 2/2

- Query the comparison results again – all differences are fixed

```
SQL> SELECT scan_id, parent_scan_id, status, current_dif_count, count_rows
       FROM dba_comparison_scan_summary
       WHERE comparison_name = 'MY_COMPARISON'
       ORDER BY last_update_time;
```

SCAN_ID	PARENT_SCAN_ID	STATUS	CURRENT_DIF_COUNT	COUNT_ROWS
13	12	SUC	0	36
14	12	SUC	0	35
12		SUC	0	107

```
SQL> SELECT scan_id, index_value, local_rowid, remote_rowid, status
       FROM dba_comparison_row_dif
       WHERE comparison_name = 'MY_COMPARISON';
```

SCAN_ID	INDEX_VALU	LOCAL_ROWID	REMOTE_ROWID	STA
13	100	AAATfUAALAAAjxLAAA	AAATfVAALAAAjxTAAA	SUC
14	156	AAATfUAALAAAjxLAA4	AAATfVAALAAAjxTAA4	SUC

27 CLEANUP

- To keep the comparison template for future scans, purge the results

```
SQL> BEGIN
  DBMS_COMPARISON.PURGE_COMPARISON (
    comparison_name => 'MY_COMPARISON',
    scan_id => NULL
  );
END;
/
```

Optional a specific scan can be purged.

- If a comparison template is not required anymore, it can be dropped

```
SQL> BEGIN
  DBMS_COMPARISON.DROP_COMPARISON (
    comparison_name => 'MY_COMPARISON'
  );
END;
/
```

FURTHER INFORMATION

29 LINKS

- **Oracle Database PL/SQL Packages and Types Reference 21c**
https://docs.oracle.com/en/database/oracle/oracle-database/21/arpls/DBMS_COMPARISON.html#GUID-4876F3C2-B410-49DF-8823-38963163D899
- **Oracle Replication Administrator's Guide 18c**
<https://docs.oracle.com/en/database/oracle/oracle-database/18/strep/comparing-and-converging-data.html#GUID-A6644AD9-8EB7-4528-9E53-A6CF0616A5E9>
- **My Oracle Support**
<https://support.oracle.com>

30 QUESTIONS AND ANSWERS



CHRISTIAN GOHMANN

Principal Consultant

 +49 211 58 6664 702

 christian.gohmann@trivadis.com

 <https://www.linkedin.com/in/christian-gohmann/>

 @CGohmannDE

 <https://www.christian-gohmann.de>



trivadis Part of Accenture