

# Loading Historical Data Into Flashback Archive Enabled Tables

Philipp Salvisberg



BASEL BERN BRUGG LAUSANNE ZÜRICH DÜSSELDORF FRANKFURT A.M. FREIBURG I.BR. HAMBURG MUNICH STUTTGART VIENNA

# ■ AGENDA

1. History
2. Archive Tables
3. Map SCN to Timestamp
4. Copy FBDA-enabled Table Within an Instance
5. Migrate Conventionally Historized Table
6. Copy FBDA-enabled Table Across Instances
7. Core Messages

# ■ History of FBDA related Technology

- Version 3 (1983)
  - Multiple versions of data
- Version 4 (1984)
  - Read consistency (aka MVCC, Oracle uses flashback query internally!)
- Version 9iR2 (2002)
  - Flashback query
  - `Select * from t as of timestamp (systimestamp - interval '5' minute);`
- Version 10gR1 (2003)
  - Flashback version query
  - `Select versions_starttime, t.* from t versions between ... and ...;`

# ■ History of FBDA (1)

- Version 11gR1 (2007)
  - Flashback data archive (aka "Total Recall")
  - Extends flashback (version) queries to dedicated set of tables
  - Independent of data availability in the UNDO tablespace
  - Alter table t flashback archive x;
- Version 11gR2 (2009)
  - Maintain content of flashback data archive
  - `dbms_flashback_archive.disassociate_fba(owner=>..., table_name=>...);`
  - ... maintain the `sys_fba_hist_...` table
  - `dbms_flashback_archive.reassociate_fba(owner=>..., table_name=>...);`

## ■ History of FBDA (2)

- Version 11.2.0.4 (2013)
  - Flashback Data Archive - without history table optimization - is available in all editions
  - not part of Oracle Advanced Compression option anymore
- Version 12c (2013)
  - Import into flashback data archive
  - `dbms_flashback_archive.create_temp_history_table(...);`
  - `dbms_flashback_archive.extend_mappings;`
  - `dbms_flashback_archive.import_history(...);`
  - for 11gR2 a package `fba_import_tar.gz` is available on OTN  
<http://www.oracle.com/technetwork/indexes/samplecode/plsql-sample-522110.html>

# ■ Other Flashback Technologies (1)

- Flashback Transaction Query
  - Query metadata of historical transactions
  - Provides undo SQL statements if supplemental logging is enabled
  - alter database add supplemental log data;
  - `Select * from flashback_transaction_query where table_name = 'T';`
- Flashback Transaction
  - Rewind transactions
  - `dbms_flashback.transaction_backout(numtxs=>..., xids=>..., options=>...);`

## ■ Other Flashback Technologies (2)

- Flashback Table
  - Rewind a table to a specific point in time
  - flashback table t to timestamp (systimestamp - interval '5' minute);
- Flashback Drop
  - Recover an object from the recycle bin
  - flashback table t to before drop;
- Flashback Database
  - Rewind the database to a specific point in time
  - Database must run in ARCHIVELOG mode
  - Requires a flash recovery area
  - Alternative for point in time recovery
  - Flashback database to timestamp (systimestamp - interval '1' day);

# ■ AGENDA

1. History
2. Archive Tables
3. Map SCN to Timestamp
4. Copy FBDA-enabled Table Within an Instance
5. Migrate Conventionally Historized Table
6. Copy FBDA-enabled Table Across Instances
7. Core Messages



# ■ Example

## Enable FBDA

```
CREATE TABLE t (  
    oid          NUMBER(4,0) NOT NULL PRIMARY KEY,  
    c1          VARCHAR2(10) NOT NULL,  
    c2          VARCHAR2(10) NOT NULL  
) FLASHBACK ARCHIVE fba;
```

## Enforce creation of FBDA tables

```
BEGIN  
    dbms_flashback_archive.disassociate_fba(owner_name => USER,  
                                             table_name => 'T');  
    dbms_flashback_archive.reassociate_fba(owner_name => USER,  
                                            table_name => 'T');  
END;  
/
```

# FBDA Tables

Flashback Data Archive Tables

T	
P *	OID NUMBER (4)
*	C1 VARCHAR2 (10 BYTE)
*	C2 VARCHAR2 (10 BYTE)
T_PK (OID)	

SYS_FBA_HIST_107038	
RID	VARCHAR2 (4000 BYTE)
STARTSCN	NUMBER
ENDSCN	NUMBER
XID	RAW (8)
OPERATION	VARCHAR2 (1 BYTE)
OID	NUMBER (4)
C1	VARCHAR2 (10 BYTE)
C2	VARCHAR2 (10 BYTE)

SYS_FBA_TCRV_107038	
RID	VARCHAR2 (4000 BYTE)
STARTSCN	NUMBER
ENDSCN	NUMBER
XID	RAW (8)
OP	VARCHAR2 (1 BYTE)
SYS_FBA_TCRV_IDX_107038 (RID)	

SYS_FBA_DDL_COLMAP_107038	
STARTSCN	NUMBER
ENDSCN	NUMBER
XID	RAW (8)
OPERATION	VARCHAR2 (1 BYTE)
COLUMN_NAME	VARCHAR2 (255 BYTE)
TYPE	VARCHAR2 (255 BYTE)
HISTORICAL_COLUMN_NAME	VARCHAR2 (255 BYTE)

- All FBDA tables are maintained asynchronously by the ora\_fbda\_<sid> process
- Most current changes are available in UNDO only!
- SYS\_FBA\_HIST\_<object\_id>
  - Content history
  - Range partitioned by ENDSCN
- SYS\_FBA\_TCRV\_<object\_id>
  - Validity of rows in T
- SYS\_FBA\_DDL\_COLMAP\_<object\_id>
  - Structure history (column history)

# ■ Querying Current Data – Execution Plan

```
SELECT * FROM t;
```

```
-----  
| Id  | Operation                | Name |  
-----  
|  0  | SELECT STATEMENT         |      |  
|  1  | TABLE ACCESS FULL      | T    |  
-----
```

# Flashback Query with FBDA – Execution Plan

```
SELECT * FROM t AS OF TIMESTAMP SYSTIMESTAMP - INTERVAL '1' SECOND;
```

Id	Operation	Name	
0	SELECT STATEMENT		
1	VIEW		
2	<b>UNION-ALL</b>		
* 3	<b>FILTER</b>		
4	PARTITION RANGE SINGLE		Based on
* 5	TABLE ACCESS FULL	SYS FBA <b>HIST</b> 107038	History
* 6	<b>FILTER</b>		
7	MERGE JOIN <b>OUTER</b>		
8	<b>SORT JOIN</b>		Based on
* 9	TABLE ACCESS FULL	<b>T</b>	Current
* 10	<b>SORT JOIN</b>		Table
* 11	TABLE ACCESS FULL	SYS FBA <b>TCRV</b> 107038	

# ■ SYS\_FBA\_HIST\_<object\_id>

- Content history
- Rows which are not stored in the current table (T) anymore
- Columns based on current table (T) plus
  - RID: original ROWID
  - STARTSCN: SCN when the row became valid
  - ENDSCN: SCN when the row became invalid (NOT NULL)
  - XID: Transaction identifier
  - OPERATION: I=insert, U=update, D=delete
- Range partitioned by ENDSCN
- DML allowed if FBDA is disassociated
- Limited DDL allowed if FBDA is disassociated (e.g. create index)

## ■ SYS\_FBA\_TCRV\_<object\_id>

- Validity of rows in current table (T)
- Columns:
  - RID: Rowid in current table (T)
  - STARTSCN: SCN when the row became valid (NOT NULL)
  - ENDSCN: SCN when the row became invalid
  - XID: Transaction Identifier
  - OP: I=Insert, U=Update
- Last change per operation is visible in this table
- DDL and DML are not supported on this table

## ■ SYS\_FBA\_DDL\_COLMAP\_<object\_id>

- Structure history (column history)
- Used to reconstruct the table structure for flashback query
- For flashback version query the most current structure is used to avoid problems with conflicting column names
- Columns:
  - STARTSCN: SCN when the column became valid (NOT NULL)
  - ENDSCN: SCN when the column became invalid (NULL)
  - XID: Transaction identifier, always empty
  - OPERATION: always empty
  - COLUMN\_NAME: name of the column in the most current structure
  - TYPE: data type
  - HISTORICAL\_COLUMN\_NAME: name of the column
- DDL and DML are not supported on this table

# ■ AGENDA

1. History
2. Archive Tables
3. Map SCN to Timestamp
4. Copy FBDA-enabled Table Within an Instance
5. Migrate Conventionally Historized Table
6. Copy FBDA-enabled Table Across Instances
7. Core Messages



# ■ SCN – System Change Number

- Oracle uses its own time standard SCN for FBDA
- The SCN is initialized during database creation and is valid for a single Oracle instance
- The SCN is synchronized among database instances, e.g. when accessing data via a database link (see also MOS note 1376995.1)
- A SCN may not represent a date before 1988-01-01
- The time spent between two SCNs is varying,
  - It may be shorter when the database instance is executing a lot of transactions
  - It may be longer in more idle times or when the database instance is shut down

# ■ Mapping SCN to Timestamp – Example

SCN starting with 3731000



Timestamp starting with 2014-03-28 08:00:00

## ■ Table SYS.SMON\_SCN\_TIME

- Map SCNs to TIMESTAMPs and vice versa through:
  - SCN\_TO\_TIMESTAMP
  - TIMESTAMP\_TO\_SCN
- Oracle states that the usual precision of the result value is 3 seconds
- A row handles up to 100 SCNs using the column TIM\_SCN\_MAP
- The ora\_smon\_<sid> process maintains the content of the table
  - Creates a row every 5 minutes
  - Deletes old entries according UNDO and FBDA configuration
- The table is part of a cluster (not designed for large volume)
  - ORA-2475: maximum cluster chain block count of 65534 has been exceeded
  - See bug 18294320 (affects 11.2.0.4 and 12.1.0.1)
  - Workaround is to create the table smon\_scn\_time outside of the cluster

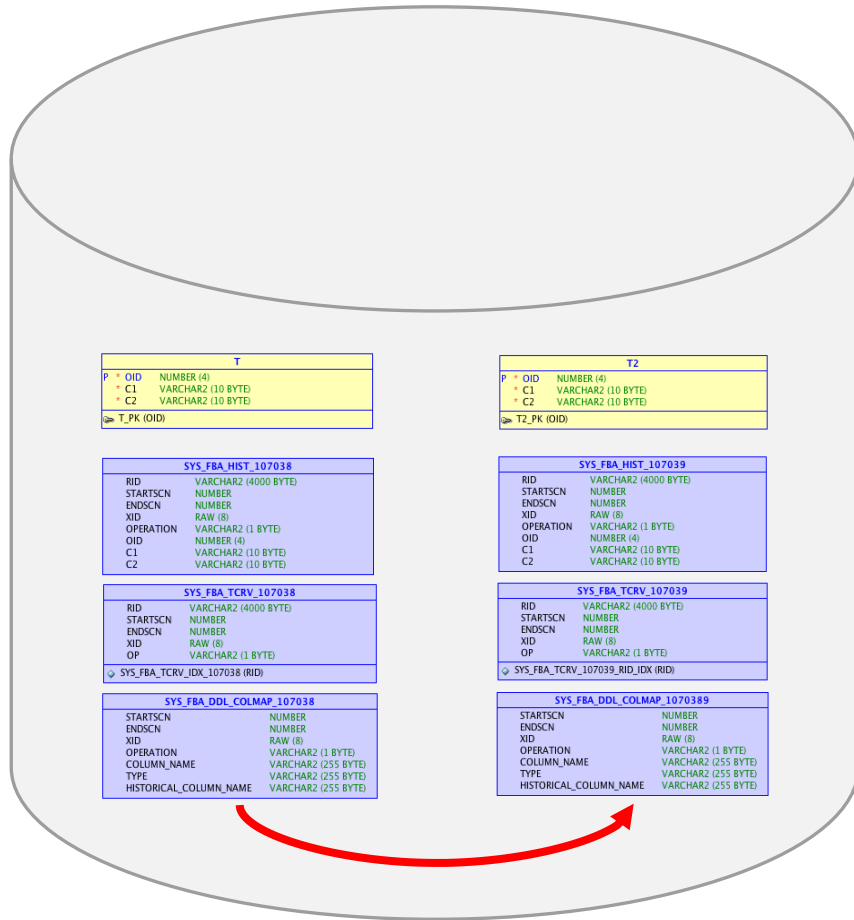
## ■ SYS.SMON\_SCN\_TIM – Example (11.2 & 12.1)

Column	Data Type	Description	Example
Thread	Number	Not yet known	0
Time_mp	Number	Number of seconds since 1970-01-01 (UTC)	1384332055
Time_dp	Date	Date value of Time_mp (UTC)	2013-11-13 08:40:55
Scn_wrp	Number	Number of times the Scn_bas has exceeded its max. 32 bit value: floor(scn / power(2, 32))	1877
Scn_bas	Number	SCN as original 32-bit representation: mod(scn, power(2, 32))	57569150
Num_mappings	Number	Number of timestamp to SCN mappings in tim_scn_map	84
Tim_scn_map	Raw(1200)	Array of timestamp to SCN mappings, 12 bytes for each entry, 4 bytes for Time_mp, 4 bytes for Scn_bas, 2 bytes for Scn_wrp, 2 bytes not yet known (example from a "Little Endian" system)	1A3B83527F6F6E0355 07C7B61D3B8352806F 6E0355070000...
Scn	Number	System Change Number	8061711183742
Orig_thread	Number	Not yet known	0

# ■ AGENDA

1. History
2. Archive Tables
3. Map SCN to Timestamp
4. Copy FBDA-enabled Table Within an Instance
5. Copy FBDA-enabled Table Across Instances
6. Migrate Conventionally Historized Table
7. Core Messages

# ■ Copy FBDA-enabled Table Within an Instance?



## Reason

- Move table to another schema
- Make a copy for testing purposes
- Save table including history before migration

# Content of Source Table

demo1.sql

```
SQL> SELECT oid, c1, c2, versions_operation AS op,  
2         versions_startscn AS startscn,  
3         versions_endscn AS endscn,  
4         versions_starttime AS starttime,  
5         versions_endtime AS endtime  
6 FROM t versions BETWEEN scn minvalue AND maxvalue  
7 ORDER BY oid, versions_startscn;
```

OID	C1	C2	OP	STARTSCN	ENDSCN	STARTTIME	ENDTIME
1	A	A1	I	3790144	3790146	2014-03-28 18:58:29	2014-03-28 18:58:29
1	A	A2	U	3790146	3790148	2014-03-28 18:58:29	2014-03-28 18:58:29
1	A	A3	U	3790148	3790150	2014-03-28 18:58:29	2014-03-28 18:58:32
1	A	A4	U	3790150		2014-03-28 18:58:32	
2	B	B1	I	3790152		2014-03-28 18:58:35	
3	C	C1	I	3790154	3790156	2014-03-28 18:58:35	2014-03-28 18:58:35
3	C	C2	U	3790156	3790158	2014-03-28 18:58:35	2014-03-28 18:58:38
3	C	C3	U	3790158	3790160	2014-03-28 18:58:38	2014-03-28 18:58:41
3	C	C3	D	3790160	3790160	2014-03-28 18:58:41	2014-03-28 18:58:41

9 rows selected.

# ■ 1) Copy Current Rows

```
SQL> INSERT INTO t2
  2     (OID, c1, c2)
  3     SELECT OID, c1, c2
  4     FROM t;
```

2 rows created.



## ■ 2) Disassociate FBDA

```
SQL> BEGIN
  2      dbms_flashback_archive.disassociate_fba(owner_name => USER,
  3                                              table_name => 'T2');
  4  END;
  5  /
```

PL/SQL procedure successfully completed.

## ■ 3) Insert History

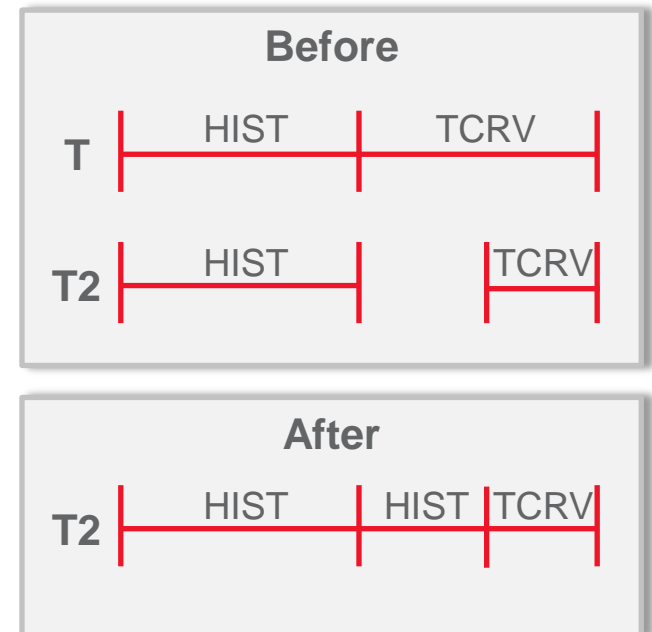
```
SQL> INSERT INTO t2_hist
  2      (rid, startscn, endscn, xid, operation, OID, c1, c2)
  3      SELECT NULL                AS rid,
  4             versions_startscn   AS startscn,
  5             versions_endscn     AS endscn,
  6             NULL                 AS xid,
  7             versions_operation  AS operation,
  8             OID,
  9             c1,
 10             c2
 11      FROM t versions BETWEEN scn minvalue AND maxvalue
 12      WHERE versions_endscn IS NOT NULL;
```

7 rows created.

## ■ 4) Insert Validity of Current Record

```
SQL> INSERT INTO t2_hist
  2      (rid, startscn, endscn, xid, operation, OID, c1, c2)
  3      SELECT NULL          AS rid,
  4          h.startscn,
  5          h2.startscn AS endscn,
  6          NULL          AS xid,
  7          h.op          AS operation,
  8          t2.oid,
  9          t2.c1,
 10         t2.c2
 11     FROM t
 12     INNER JOIN t_tcrv h
 13         ON h.rid = t.rowid
 14     INNER JOIN t2
 15         ON t2.oid = t.oid
 16     INNER JOIN t2_tcrv h2
 17         ON h2.rid = t2.rowid
 18     WHERE h.endscn IS NULL
 19         AND h2.endscn IS NULL;
```

2 rows created.



## ■ 5) Reassociate FBDA

```
SQL> BEGIN
  2      dbms_flashback_archive.reassociate_fba(owner_name => USER,
  3                                              table_name => 'T2');
  4  END;
  5  /
```

PL/SQL procedure successfully completed.

## ■ Done?

```
SQL> SELECT * FROM t AS OF SCN 3790144;
```

```
OID C1 C2
```

```
--- -- --
```

```
1 A A1
```

```
1 row selected.
```

```
SQL>
```

```
SQL> SELECT * FROM t2 AS OF SCN 3790144;
```

```
SELECT * FROM t2 AS OF SCN 3790144
```

```
*
```

```
ERROR at line 1:
```

```
ORA-01466: unable to read data - table definition has changed
```

## ■ But AS OF TIMESTAMP Works Better

```
SQL> SELECT * FROM t2 AS OF TIMESTAMP scn_to_timestamp(3790144);
```

```
no rows selected
```

```
SQL> SELECT * FROM t2 AS OF TIMESTAMP scn_to_timestamp(3790146);
```

```
no rows selected
```

```
SQL> SELECT * FROM t2 AS OF TIMESTAMP scn_to_timestamp(3790148);
```

```
no rows selected
```

```
SQL> SELECT * FROM t2 AS OF TIMESTAMP scn_to_timestamp(3790150);
```

OID	C1	C2
-----	-----	-----
1	A	A3

```
1 row selected.
```

# ■ What is the Problem?

```
SQL> SELECT MIN(startscn) AS startscn
       2     FROM t_hist;
```

```
STARTSCN
```

```
-----
3790144
```

```
1 row selected.
```

```
SQL>
```

```
SQL> SELECT startscn, column_name
       2     FROM t2_ddl_colmap
       3     WHERE column_name = 'OID';
```

```
STARTSCN COLUMN_NAME
```

```
-----
3790291  OID
```

```
1 row selected.
```

Columns in T2 do  
not exist at SCN 3790144!

## ■ 6) Fix DDL\_COLMAP Validity (Avoid ORA-1466)

```
SQL> BEGIN
  2     sys.tvd_fba_helper.fix_col_map_validity(in_owner_name => USER,
  3                                           in_table_name => 'T2');
  4 END;
  5 /
```

PL/SQL procedure successfully completed.

```
SQL> SELECT startscn, column_name
  2     FROM t2_ddl_colmap
  3     WHERE column_name = 'OID';
```

STARTSCN	COLUMN_NAME
----------	-------------

3790144	OID
---------	-----

```
SQL> SELECT * FROM t2 AS OF SCN 3790144;
```

OID	C1	C2
-----	----	----

---	--	--
1	A	A1

Experimental!

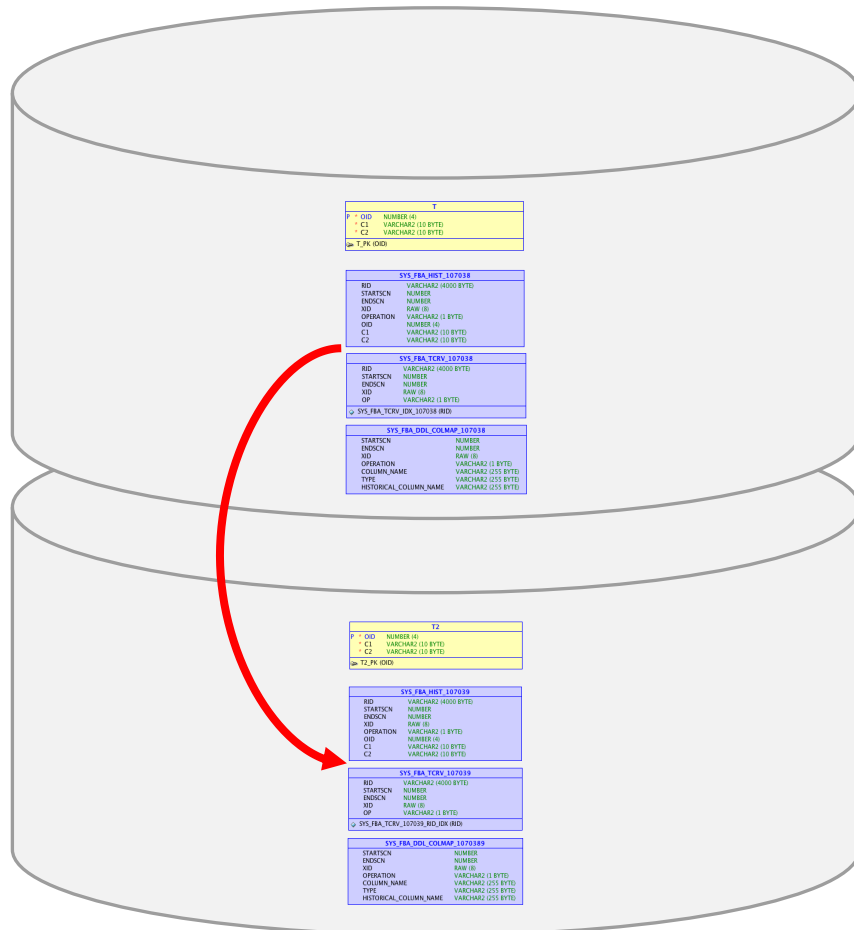
Ask Oracle Support  
how to proceed  
if use in production is considered



# ■ AGENDA

1. History
2. Archive Tables
3. Map SCN to Timestamp
4. Copy FBDA-enabled Table Within an Instance
5. Copy FBDA-enabled Table Across Instances
6. Migrate Conventionally Historized Table
7. Core Messages

# ■ Copy FBDA-enabled Table Across Instances?



## Reason

- Copy production data to test environment
- Move application or part of the application
- Setup new database
  - version, block size change
  - overcome export/import limitations for FBDA

# ■ What is Different to Copying Within the Same Instance?

The content of SYS.SMON\_SCN\_TIM in the source and the target instance is not identical

Three Options:

- Option 1: Automatically Extend SYS.SMON\_SCN\_TIM
- Option 2: Copy SYS.SMON\_SCN\_TIM
- Option 3: Manually Extend SYS.SMON\_SCN\_TIM

# ■ Option 1: Automatically Extend SYS.SMON\_SCN\_TIM

## Pro:

- The supported way
  - `dbms_flashback_archive.extend_mappings()` is part of 12c
  - `dbms_fda_mappings.extend_mappings()` is available via OTN for 11gR2 (similar but not equal to the 12c algorithm)

## Con:

- Maps all missing timestamps from 1988-01-01 to the lower bound of `sys.smon_scn_tim`
  - Max. 10 Mio SCNs, evenly distributed
  - Max. precision is ~82 seconds and increasing (12.1.0.1)
  - Huge gap at the original lower bound is likely because of rounding differences
- Not feasible for a short history with a required precision of around 3 sec.
- SCN to timestamp conversion required for history load
- Ambiguous history (contains history records which were never valid)

# ■ What is a Huge Gap?

```
SQL> SELECT timestamp_to_scn(TIMESTAMP '2013-12-25 17:43:55') ts1,  
2         timestamp_to_scn(TIMESTAMP '2014-03-28 13:52:35') ts2,  
3         timestamp_to_scn(TIMESTAMP '2014-03-28 13:52:36') ts3  
4         FROM dual;
```

```
2013-12-25 17:43:55 2014-03-28 13:52:35 2014-03-28 13:52:36  
-----  
                9999998                9999999                8061717616701
```

1 row selected.

More than 3 Months!

## ■ Option 2: Copy SYS.SMON\_SCN\_TIM

### Pro:

- Simple (technically)
- No conversion of SCN is required for history load
- Precision of source and target are identical
- Unambiguous history (for the SCN based period)

### Con:

- Not supported
  - Approval required
  - Should be reserved to Oracle
- Not feasible if target database has already a flashback data archive
- To be used with care if other flashback features are used in the target database which relies on the SCN to timestamp mapping for data in UNDO

# ■ Option 3: Manually Extend SYS.SMON\_SCN\_TIM

## Pro:

- Works for all use cases, e.g. migrating conventionally historized tables
- Allows to enforce 3 second precision for historic data

## Con:

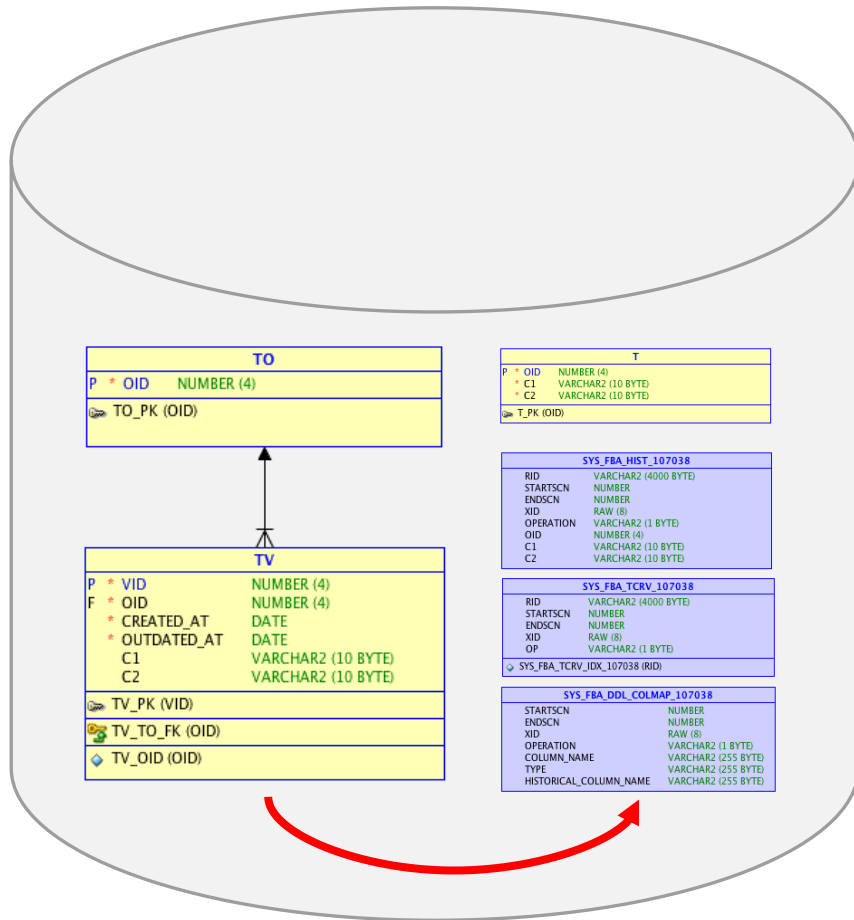
- Not supported
  - Approval required
  - Should be reserved to Oracle
- Population strategy required, e.g. keep precision of 3 seconds for current date minus 3 years.
- One row per mapping as long as the tim\_scn\_map RAW column is not fully understood
- SCN to timestamp conversion required for history load
- Ambiguous history (contains history records which were never valid)

# ■ AGENDA

1. History
2. Archive Tables
3. Map SCN to Timestamp
4. Copy FBDA-enabled Table Within an Instance
5. Copy FBDA-enabled Table Across Instances
6. Migrate Conventionally Historized Table
7. Core Messages



# ■ Migrate Conventionally Historized Table?



## Reason

- Simplify application
  - Use vendor solution instead of homegrown framework which is difficult to understand and maintain
- Improve query performance
  - When the latest versions are queried in typical use cases
  - FBDA causes no overhead when querying the most recent versions
  - Archive tables are access only for flashback (version) queries
- Improve DML performance, if row level triggers are used

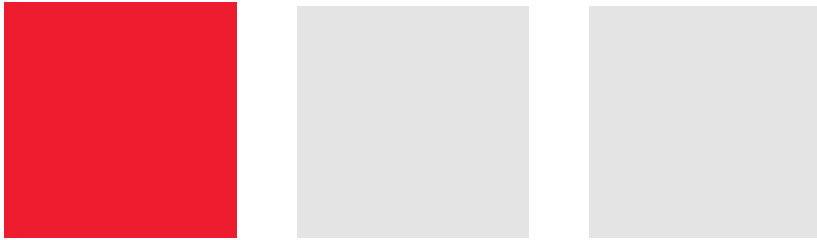
# ■ What is Different to Copying?

- No SCN based period is available in the source
- Granularity change
  - E.g. from seconds to 3 seconds
  - Might lead to ambiguous history (history records which were never valid)
- Object and version table are migrated to a single target table, all columns may be versioned, incl. the primary key
- Providing the same API (set of columns) to the application might be challenging, e.g. primary key of a version
- Beside that it, is similar to copying a FBDA table

# ■ AGENDA

1. History
2. Archive Tables
3. Map SCN to Timestamp
4. Copy FBDA-enabled Table Within an Instance
5. Copy FBDA-enabled Table Across Instances
6. Migrate Conventionally Historized Table
7. Core Messages

# ■ Core Messages



- Populating the history table itself is quite simple
- Avoiding ORA-1466 might not be possible without amending the "locked" DDL\_COLMAP table
- Loading historical data into FBDA-enabled tables requires a strategy to populate historical SCN to timestamp mappings in SYS.SMON\_SCN\_TIME
- Contact Oracle Support to elaborate how to proceed
- Further improvement is expected for coming releases

# Questions and answers ...

Philipp Salvisberg  
Senior Principal Consultant

[philipp.salvisberg@trivadis.com](mailto:philipp.salvisberg@trivadis.com)



BASEL BERN BRUGES LAUSANNE ZÜRICH DÜSSELDORF FRANKFURT A.M. FREIBURG I.BR. HAMBURG MUNICH STUTTGART VIENNA