

# Online Statistics Gathering for ETL

Andrej Pashchenko

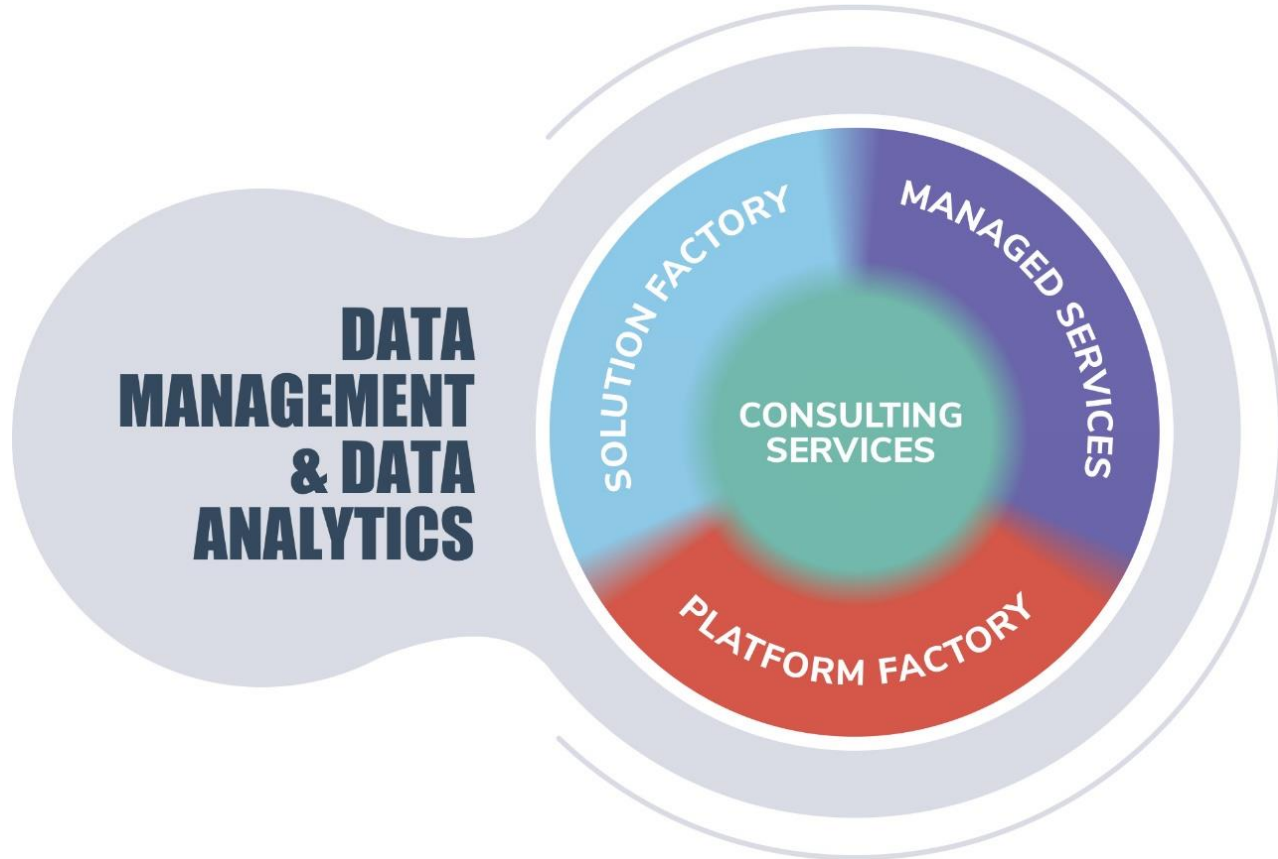


@Andrej\_SQL

# doag2018



- We help to generate added value from data



# ■ With over 650 specialists and IT experts in your region.



- 16 Trivadis branches and more than 650 employees
- Experience from more than 1,900 projects per year at over 800 customers
- 250 Service Level Agreements
- Over 4,000 training participants
- Research and development budget: CHF 5.0 million
- Financially self-supporting and sustainably profitable

# ■ About Me

- Working at Trivadis GmbH, Düsseldorf
- Focus on Oracle:
  - Data Warehousing
  - Application Development
  - Application Performance
- Course instructor „Oracle 12c New Features for Developers“ und „Beyond SQL and PL/SQL“



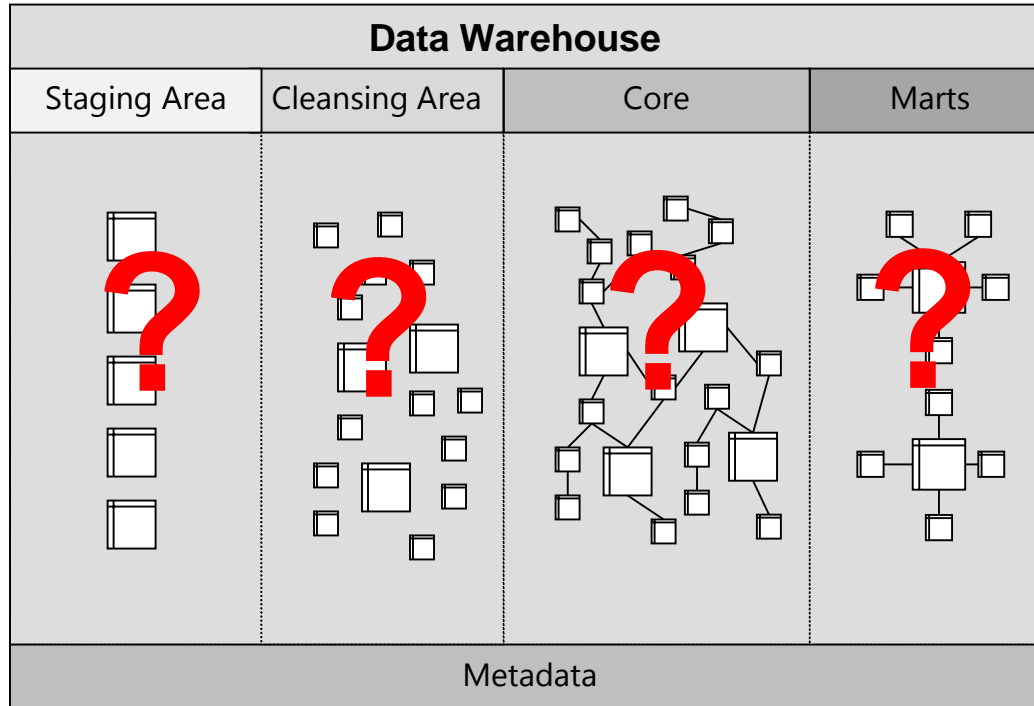
Blog: <http://blog.sqlora.com>

# ■ Online Statistics Gathering for Bulk Loads

- new feature of Oracle 12c
- for direct path writes
  - **C**reate **T**able **A**s **S**elect (CTAS)
  - Bulk Inserts
- into an empty segment
- table and basic columns statistics are collected „on the fly“ (piggyback)
- no additional table scan is required

**Sounds good for ETL?**

# ■ What ETL scenarios can benefit?



# Control, Preconditions, Scope

## ■ Switch ON/OFF, selective enable/disable

- The behaviour is controlled by the parameter `_optimizer_gather_stats_on_load`
- The default is TRUE

```
ALTER SYSTEM(SESSION) SET "_optimizer_gather_stats_on_load" = TRUE (FALSE);
```

- Hints: `(NO_) GATHER_OPTIMIZER_STATISTICS`

```
INSERT /*+ APPEND GATHER_OPTIMIZER_STATISTICS */ INTO ...
```

- These hints are only meaningful to override the parameter setting for particular SQL
- **Do not enforce** gathering without preconditions (empty table, direct path)



# How do I know it works?

- New step in explain plan, but not sufficient to judge
- It only means: the feature is turned on and will be considered
- Also check LAST\_ANALYZED for table and NOTES for column statistics:

Id	Operation	Name
0	INSERT STATEMENT	
1	LOAD AS SELECT	T2
2	OPTIMIZER STATISTICS GATHERING	
3	TABLE ACCESS FULL	T

```
SQL> SELECT table_name, num_rows,
2         last_analyzed
3         FROM user_tab_statistics
4         WHERE table_name = 'T2';
```

TABLE	NUM_ROWS	LAST_ANALY
T2	10000	09-OCT-18

```
SQL> SELECT column_name, num_distinct,
2         num_buckets, sample_size, notes
3         FROM user_tab_col_statistics
4         WHERE table_name = 'T2';
```

COLUM	NUM_DIST	NUM_B	SAMPLE_S	NOTES
N	10000	1	10000	STATS_ON_LOAD

- Also reflected in (ALL|DBA)USER\_TAB\_STATS\_HISTORY

# ■ How do I know it works?

- Oracle Documentation suggests to look at USER\_TAB\_MODIFICATIONS.INSERTS column. If the query returns a row indicating the number of rows loaded, then the most recent bulk load **did not gather** statistics automatically.

```
INSERT /*+ APPEND */ ...

SQL> begin
  2     dbms_stats.FLUSH_DATABASE_MONITORING_INFO;
  3 end;
  4 /

PL/SQL procedure successfully completed.

SQL> SELECT inserts
  2 FROM   user_tab_modifications
  3 WHERE  table_name = 'T';

   INSERTS
-----
      10000
```

# ■ Direct Path Insert (I)

- DB-blocks are written to disk by user server process (not DBWR), bypassing the buffer cache
- CTAS is always performing direct path
- Use APPEND hint for **INSERT AS SELECT** (IAS)

```
INSERT /*+ append */ INTO ... SELECT
```



Hints are evil ?!

Not this one! It is a **non-optimizer hint** and the official and only possible way to tell the DB to do a direct path insert

- Also INSERT mit APPEND\_VALUES from FORALL statement in PL/SQL

# ■ Direct Path Insert (II) – Parallel DML

- If INSERT is done in parallel (DML) the default mode is direct path
- Enable parallel DML for your session or with the hint just for your SQL
- Specify PARALLEL clause for the table or use PARALLEL hint in your INSERT:
- Parallel DML would not be possible in some cases
- Do not rely on that and always specify APPEND hint altogether

```
ALTER SESSION ENABLE PARALLEL DML;  
or  
INSERT /*+ enable_parallel_dml */ INTO ...
```

```
ALTER TABLE tx2 PARALLEL 4;  
or  
INSERT /*+ parallel */ INTO tx2 ...
```

```
INSERT /*+ append enable_parallel_dml  
parallel */  
INTO ...
```

## ■ Direct Path Insert (III)

- There are some cases where APPEND hint can be ignored (e.g. triggers, enabled foreign keys).
- Reliable FK-constraints: valuable for the CBO, but not disruptive for ETL (RELY DISABLE NOVALIDATE)
- Trigger: redesign, drop 😊
- How to proof?

– LOAD AS SELECT in explain plan

```
-----  
| 0 | INSERT STATEMENT |  
| 1 | LOAD AS SELECT |  
| 2 | OPTIMIZER STATISTICS GATHERING |
```

– You cannot read the table in the same transaction after direct path insert

```
ORA-12838: cannot read/modify an object  
after modifying it in parallel
```

# Direct Path and MERGE?

```
SQL> MERGE /*+ append */
  2 INTO tx2 USING tx ON (tx.n = tx2.n)
  3 WHEN NOT MATCHED THEN INSERT (N)
VALUES (tx.n) ;
10,000 rows merged.
```

Id	Operation	Name
0	MERGE STATEMENT	
1	MERGE	TX2
2	VIEW	
* 3	HASH JOIN RIGHT OUTER	
4	TABLE ACCESS FULL	TX2
5	TABLE ACCESS FULL	TX

```
SQL> SELECT count(*) FROM tx2;
```

```
ORA-12838: cannot read/modify an object
after modifying it in parallel
```

direct path write has  
happend

```
SQL> INSERT /*+ append*/ INTO tx2
  2 SELECT n FROM tx
  3 WHERE n NOT IN (SELECT n FROM tx2);
10,000 rows inserted.
```

Id	Operation	Name
0	INSERT STATEMENT	
1	LOAD AS SELECT	TX2
2	OPTIMIZER STATISTICS GATHERING	
* 3	HASH JOIN RIGHT ANTI NA	
4	TABLE ACCESS FULL	TX2
5	TABLE ACCESS FULL	TX

# ■ Empty Segment

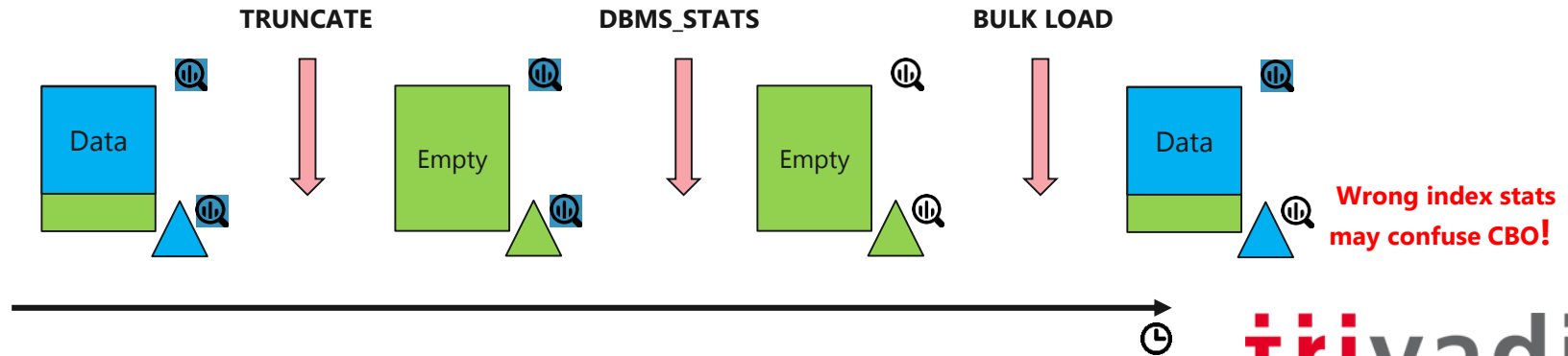
- What does „empty“ exactly mean?

Case	Online Statistics?
New Table	YES
DELETE all rows	NO
INSERT and ROLLBACK	NO
TRUNCATE (DROP STORAGE)	YES
TRUNCATE REUSE STORAGE	NO

- The rules apply to partitions as well

# ■ What kind of statistics are gathered?

- Table Statistics
- Base column statistics, also virtual columns and column groups (12.2)
- No histograms – not often used in Stage and Cleansing but can be important in Core and Mart areas
- No Index Statistics – must be gathered separately. Interactions with the default statistics maintenance job can be dangerous:





# Restrictions and Pitfalls

# ■ Restrictions and Pitfalls (I)

- Restrictions are documented
- Check your version's documentation.
- Even if you are on 12.1 also check the 12.2 documentation.
- It has better explanations, some restrictions were lifted in 12.2 and backported to 12.1, so check MOS and test it!
- Some examples:
  - IOT, nested tables
  - Statistics are locked or PUBLISH preference is set to FALSE
  - Multitable Insert (INSERT ALL/FIRST)

## ■ Restrictions and Pitfalls (II)

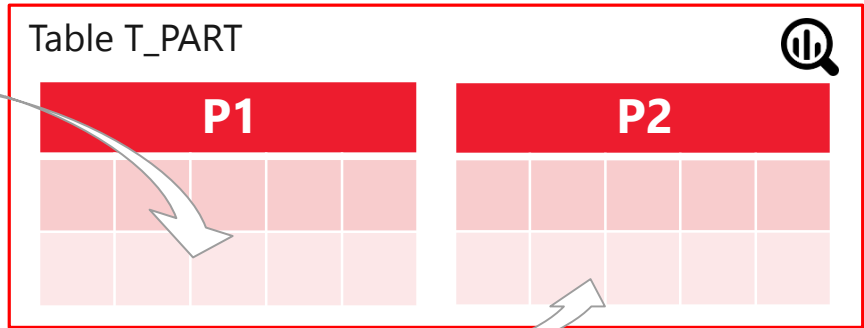
- Statistics are gathered only if all skipped columns have a default value (e.g. extending the tables and not changing the ETL process at the same time)
- Virtual columns are claimed to be a restriction in 12.1 documentation, but not in 12.2. In fact it also works in 12.1.0.2 (backport?)
- The presence of extended statistics prevents online statistics gathering (bug 18425876), fixed in 12.2, but also works in 12.1.0.2 (backport?)

# Partitioned Tables

# ■ Partitioned Tables (simple case)

- Insert into an empty partitioned table:

```
INSERT /*+ append*/  
INTO   t_part  
SELECT ...
```



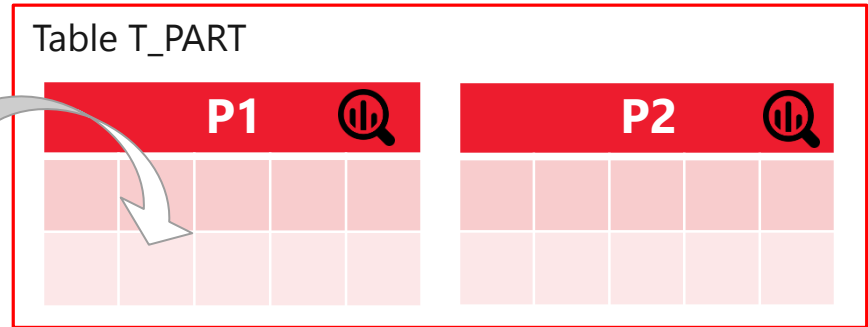
- Only **global table statistics** are gathered
- No online statistics gathering for subsequent inserts (the table is not empty anymore)  
Even if inserting into an empty partition

# ■ Partitioned Tables (using extended syntax)

- Insert into an empty partition, explicitly specifying partition name:

```
INSERT /*+ append*/  
INTO   t_part PARTITION (P1)  
SELECT ...
```

```
INSERT /*+ append*/  
INTO   t_part PARTITION (P2)  
SELECT ...
```



- Only **partition level statistics** are gathered
- Online statistics gathering is also active for subsequent inserts into yet empty partitions

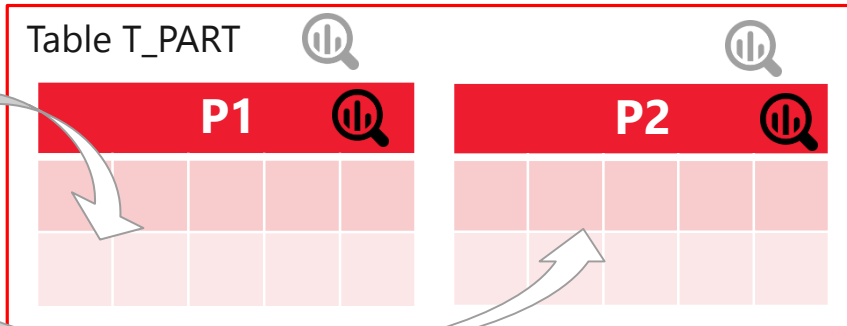
# ■ Partitioned Tables (using extended syntax+incremental)

- Insert into an empty partition, explicitly specifying partition name, INCREMENTAL-preference is TRUE:

```
exec dbms_stats.set_table_prefs(null,'T_PART','INCREMENTAL','TRUE')
```

```
INSERT /*+ append*/  
INTO t_part PARTITION (P1)
```

```
INSERT /*+ append*/  
INTO t_part PARTITION (P2)
```



- Partition level statistics and synopsis for global stats are gathered
- Online statistics gathering is also active for subsequent inserts into yet empty partitions

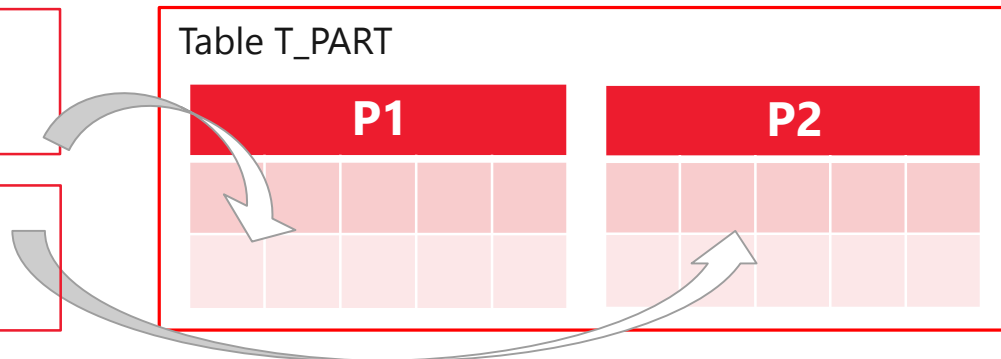
# ■ Partitioned Tables (using extended syntax+incremental)

- Insert into an empty partitioned table, no partition-extended syntax (partition name), INCREMENTAL-preference is **TRUE**:

```
exec dbms_stats.set_table_prefs(null,'T_PART','INCREMENTAL','TRUE')
```

```
INSERT /*+ append*/  
INTO   t_part
```

```
INSERT /*+ append*/  
INTO   t_part
```



- **NO** Online statistics gathering at all!
- Documented restriction



# ■ Partitioned Tables

## ■ Loading into an empty partition

	<b>INCREMENTAL=TRUE</b>	<b>INCREMENTAL=FALSE</b>
Partition name specified	partition-level stats + synopsis	partition-level stats
No partition name specified	<b>No stats at all!</b>	global stats

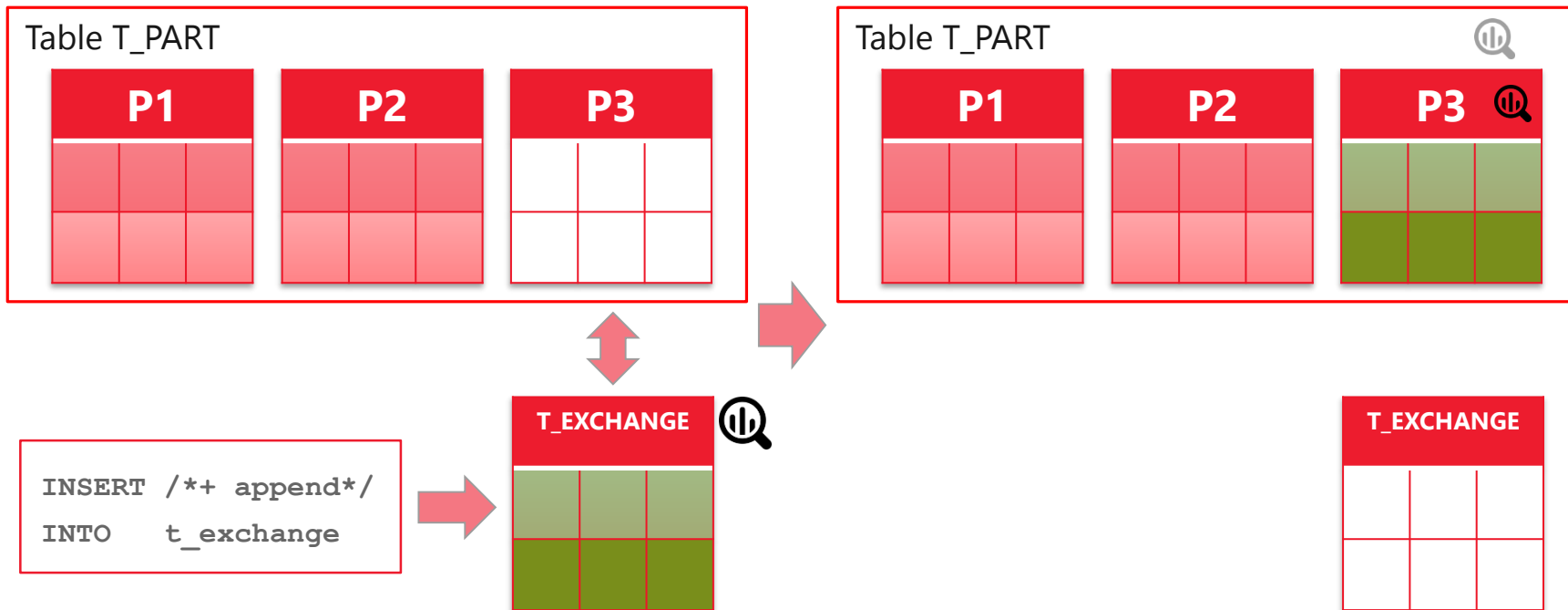
# ■ Partitioned Tables – Interval Partitioning

- Partitions are automatically created as data arrives
- No online statistics in 12.1, fixed in 12.2
- Partition names are system generated and are not known in advance
- No way for online gathering of partition level stats?
- Actually there is one – using extended syntax with values:

```
INSERT /*+ append*/  
INTO    t_part PARTITION FOR (DATE '2018-11-20')
```

- But the value is hard-coded (no bind variables), so that dynamic SQL is needed

# ■ Partitioned Tables – Partition Exchange



# ■ Partitioned Tables – Partition Exchange

- Since 12.1 synopses can be gathered on non-partitioned tables
- This happens also with online statistics gathering

```
CREATE TABLE t_exchange FOR EXCHANGE WITH TABLE t_part;

BEGIN
  dbms_stats.set_table_prefs(null,'t_exchange','INCREMENTAL','TRUE');
  dbms_stats.set_table_prefs(null,'t_exchange','INCREMENTAL_LEVEL','TABLE');
END;

INSERT /*+ append */ INTO t_exchange ...
-- Online statistics including synopsis are gathered
ALTER TABLE T_PART EXCHANGE PARTITION P3 WITH TABLE t_exchange;
-- Partition statistics and synopsis are exchanged
```

# Autonomous Data Warehouse Cloud Service (ADWC) What's new?

# ■ Autonomous Data Warehouse Cloud – What's new?

Statistics are gathered automatically

Unlike 12c, this works also

- for non-empty tables
- for histograms

Id	Operation	Name
0	INSERT STATEMENT	
1	LOAD AS SELECT	TARGET
2	PX COORDINATOR	
3	PX SEND QC (RANDOM)	:TQ10000
4	<b>OPTIMIZER STATISTICS GATHERING</b>	
5	PX BLOCK ITERATOR	
6	TABLE ACCESS STORAGE FULL	SOURCE

Two new undocumented parameters

- `_optimizer_gather_stats_on_load_all` (default: TRUE)
- `_optimizer_gather_stats_on_load_hist` (default: TRUE)

# ■ Summary

😊 very useful feature if loading in combination with truncate (Stage, Cleansing)

😊 becomes better with every release since 12.1

😊 can save time and resources during ETL runs

😏 becomes more complicated in case of partitioned tables

😞 No support for MERGE /\*+ append \*/

👉 subject to check and test with **your** load scenario and **your** ETL-tool





<http://blog.sqlora.com/en/tag/osg/>

<https://blogs.oracle.com/optimizer/efficient-statistics-maintenance-for-partitioned-tables-using-incremental-statistics-part-2>

<https://danischnider.wordpress.com/2018/07/11/gathering-statistics-in-the-autonomous-data-warehouse-cloud/>



# Trivadis @ DOAG 2018

## #opencompany

- Booth: 3rd Floor – next to the escalator
- We share our Know how!  
Just come across, Live-Presentations  
and documents archive
- T-Shirts, Contest and much more
- We look forward to your visit

