

# Exadata hibrid oszlopos adattömörítés automatizálása; DB 12c partition merge

HOUG Konferencia

2017. március 28.

Németh Márk

Principal Programmer Analyst

National Instruments

# Tömörítési módok

	OLTP	Exadata Hybrid Columnar Compression
Elérhetőség	11gR1	11gR2 + Exadata
A tömörítés egysége	Adatbázis blokk (8k)	„compression unit” (több blokk: 32-64k)
Szervezési elv	Sor	Hibrid (sor és oszlop)
Milyen műveletek esetén használható	Hagyományos INSERT	<b>Csak direkt utas betöltés (APPEND, CTAS)</b>
Tömörítési arány	2x - 4x	4x - 12x
Update	Ritkán vagy soha	Soha

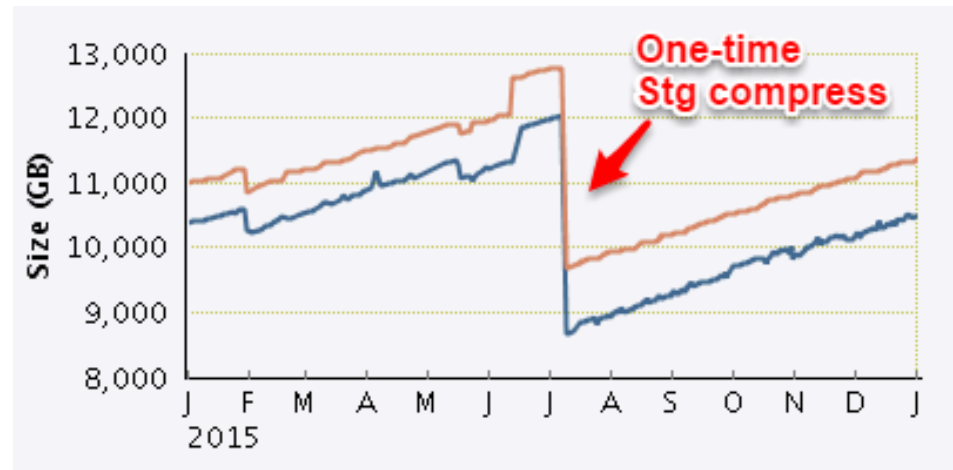
EHCC-vel jelentősen magasabb tömörítési arány érhető el, de használatához direkt utas betöltés szükséges.

# A háttértörténet

- 2012 – Adattárház migrációja Exadata-ra

- Alkalmanként használtunk EHCC-t

- Karbantartás jelleggel
- Egyedi SQL scriptek
- Felügyelet mellett
- Állásidő



- Megoldandó problémák

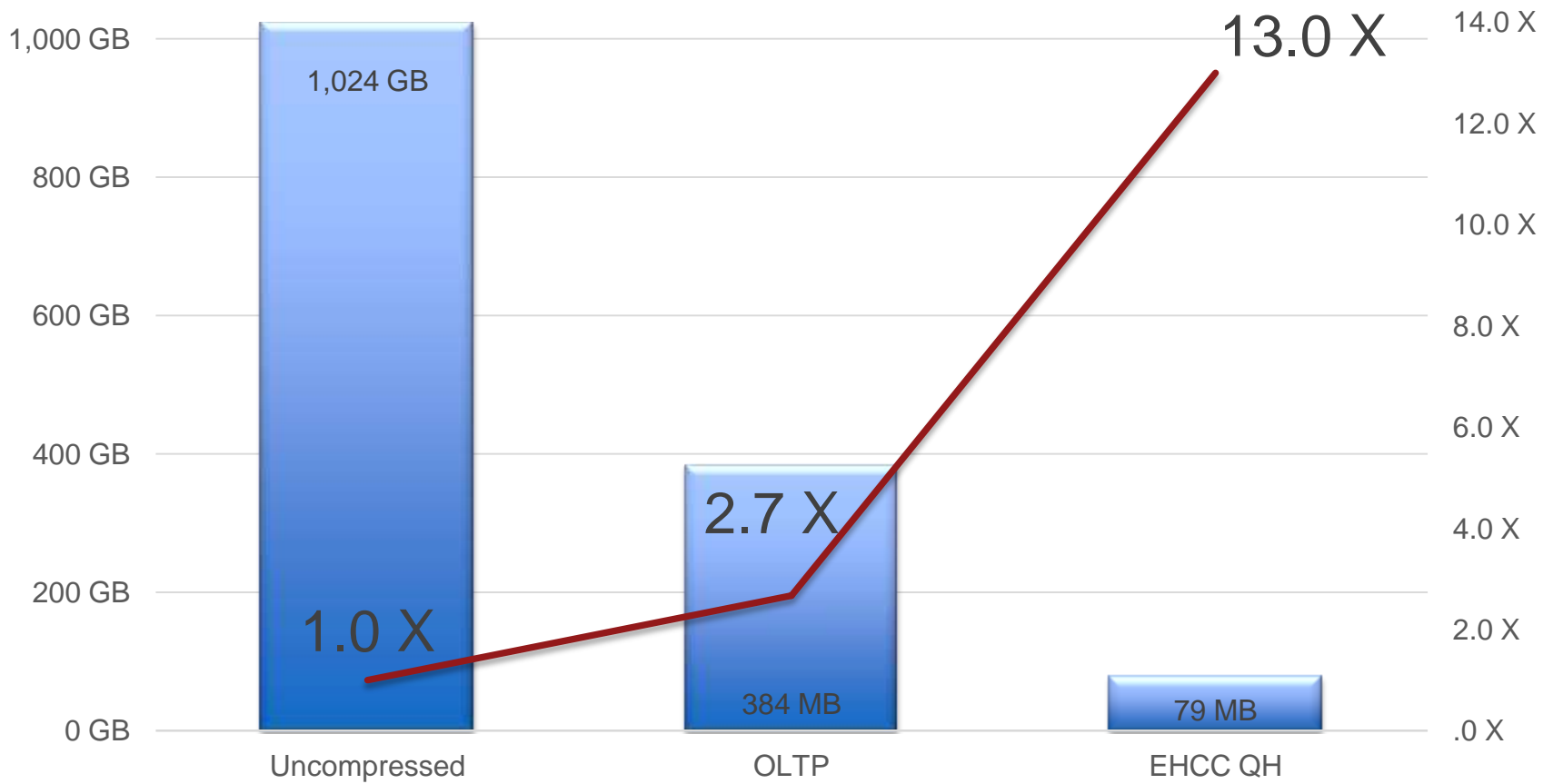
- Felügyeletet nem igénylő módon automatizálni
- Rendszeres időközönként futtatni (a táblatér soha ne hízzon túl nagyra)

# Miért lett hirtelen ilyen fontos?

- Éles rendszer
  - X2-2 Quarter Rack
- Fejlesztői és teszt
  - X2-2 Half Rack
  - **3 fejlesztői és 2 teszt rendszer**
- A fejlesztői rendszereken kezdtünk kifutni a tárhelyből
- Két E-Business Suite (forrásrendszer) integrációja jelentős tárhelyigénnyel járt az adattárházon
- X6 csak később volt esedékes

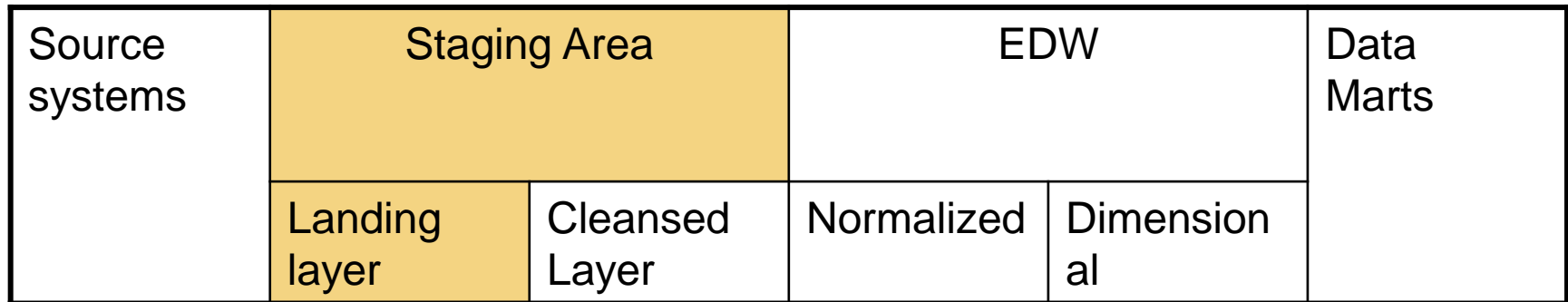
# Tömörítési arány és tárhelyigény

Tömörítési arányok az NI-nál  
Tárhelyigény 1 TB tömörítetlen adatra vetítve



# Mit tömörítettünk?

ETL



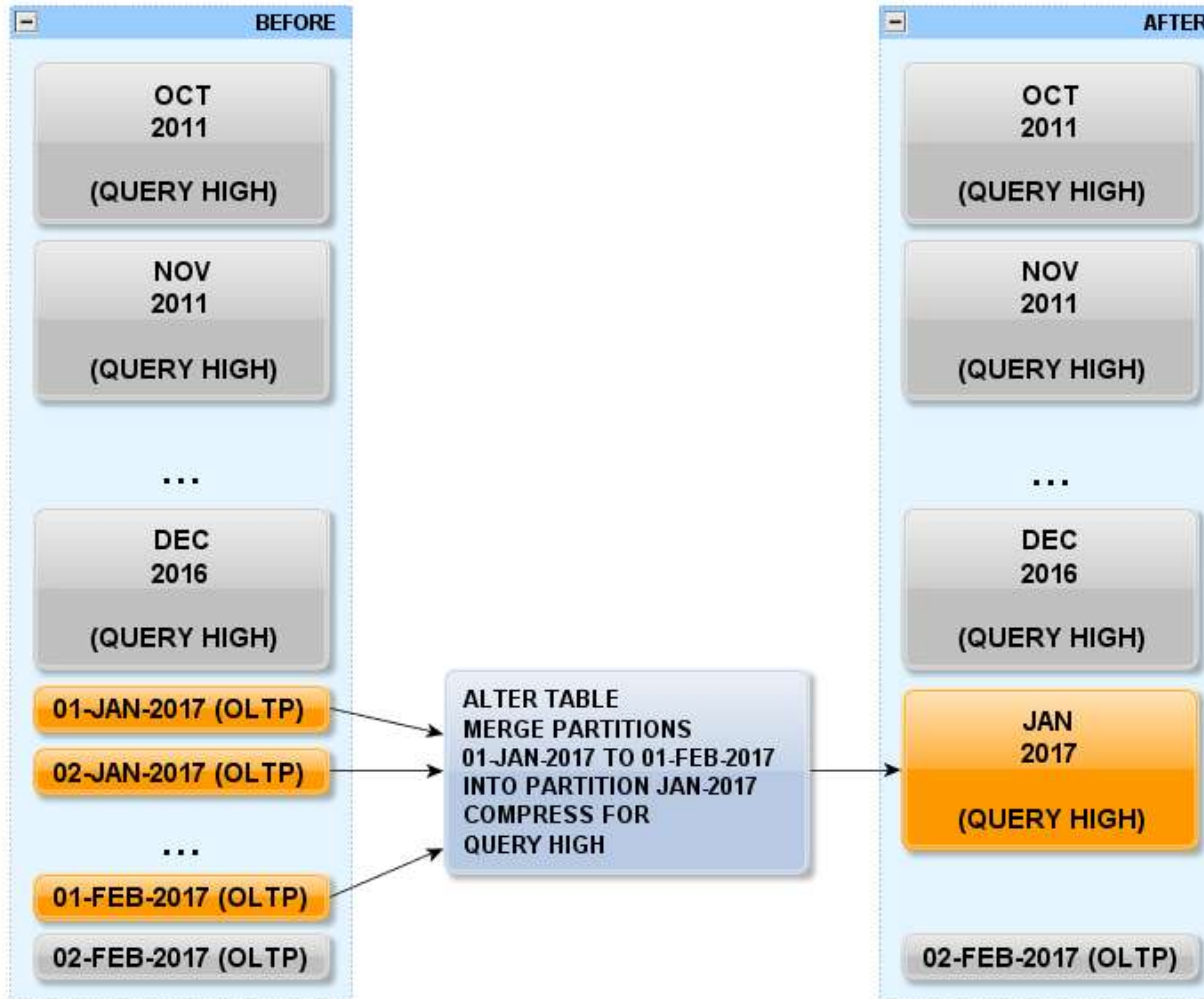
- Perzisztens staging réteg
  - A forrásrendszerrel megegyező táblaszerkezet + sorszintű meta-adat
  - Rekordok változásának követése időbélyeggel
  - Csak INSERT műveletek
  - Jelentős adatmennyiség
  - Tökéletes jelölt a tömörítésre

# Az eredeti particionálási stratégia

- Exadata migrációt megelőző megoldás
- Napi partíciók
- Az ETL nagymértékben kihasználja
  - Az utolsó partíciót írja
  - Az utolsó partíciót olvassa
- Ad-hoc lekérdezések a teljes táblára
  
- Problémák
  - Tárhely: szerény tömörítési arány csak OLTP-vel
    - Kis batch-ekben töltjük
    - Hagyományos INSERT-ekkel
  - Gyakran túl kicsi partíciók
    - Elpocsékolt tárhely (→ initial 8k?)
    - Teher a data dictionary-nek is
    - SMART SCAN „gyilkos”

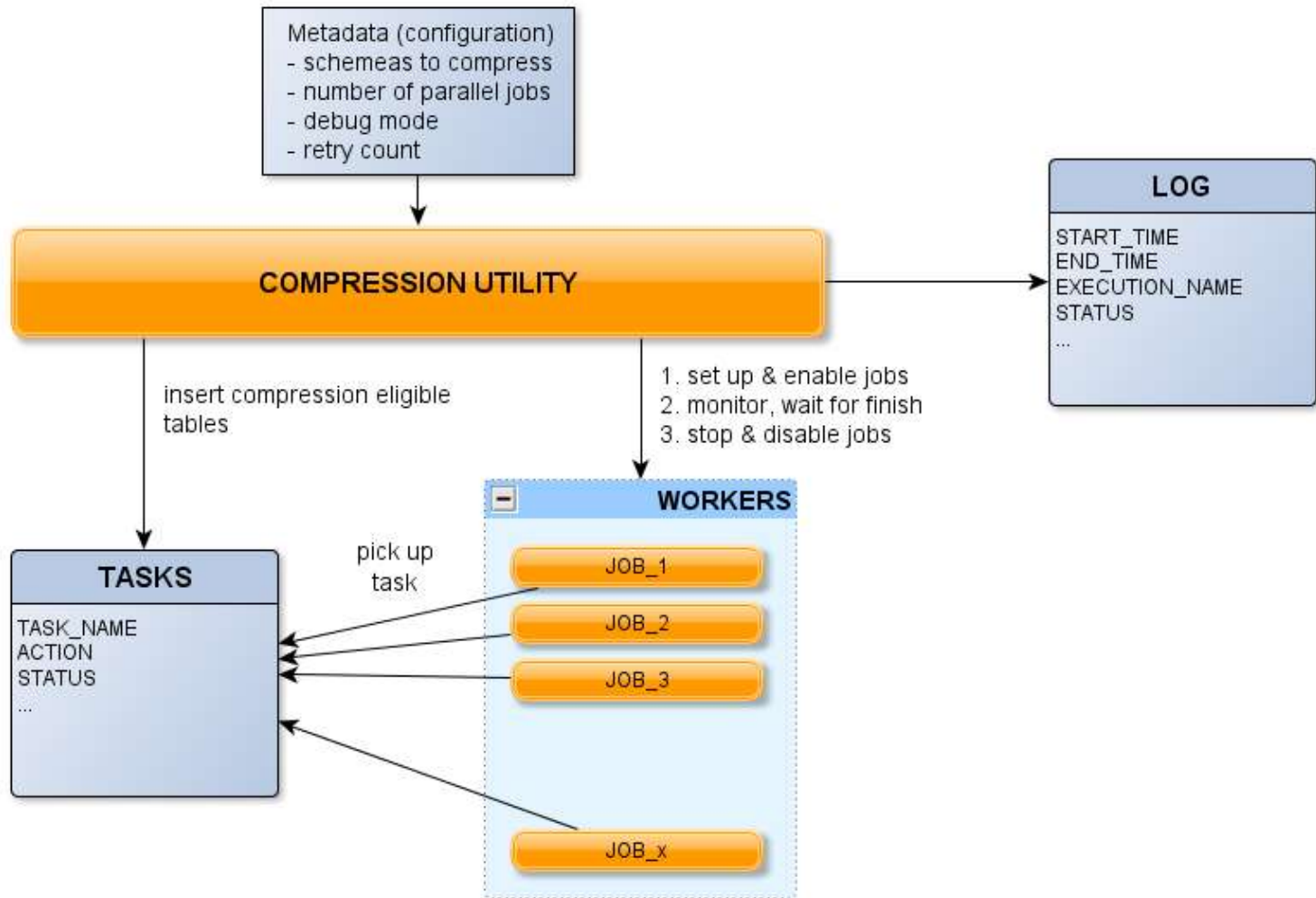


# 12c ALTER MERGE partitions





# Az automatizált tömörítés



# Partíciók összevonása egy táblában

- Range / interval partíciók
- Az interval particionálás a range particionálás kiterjesztése
- Legalább egy range partíció
- A legutolsó range partíció: „transition point”

```
ALTER TABLE m_test  
MERGE PARTITIONS p_start TO p_end  
INTO PARTITION p_new;
```

```
ALTER TABLE m_test  
MERGE PARTITIONS p1, p2, ... pX  
INTO PARTITION p_new;
```

# Range-interval átmenet (transition point)

```
CREATE TABLE m_test (  
    dt DATE  
    , id NUMBER  
)  
PARTITION BY RANGE (dt)  
INTERVAL (NUMTODSINTERVAL (1, 'DAY'))  
(PARTITION p_init  
VALUES LESS THAN (DATE '2017-03-20'))  
);
```

TABLE_NAME	PARTITION_NAME	PARTITION_POSITION	HIGH_VALUE	INTERVAL
M_TEST	P_INIT	1	3/20/2017	NO
M_TEST	SYS_P2638577	2	3/21/2017	YES
M_TEST	SYS_P2638578	3	3/22/2017	YES
M_TEST	SYS_P2638579	4	3/23/2017	YES
M_TEST	SYS_P2638580	5	3/24/2017	YES



# Merge partition - példa


TABLE_NAME	PARTITION_NAME	PARTITION_POSITION	HIGH_VALUE	INTERVAL
M_TEST	P_INIT	1	3/20/2017	NO
M_TEST	SYS_P2638581	2	3/21/2017	YES
M_TEST	SYS_P2638582	3	3/22/2017	YES
M_TEST	SYS_P2638583	4	3/23/2017	YES
M_TEST	SYS_P2638584	5	3/24/2017	YES

```
ALTER TABLE m_test  
MERGE PARTITIONS  
FOR (DATE '2017-03-20')  
TO FOR (DATE '2017-03-23')  
INTO PARTITION p_1;
```

TABLE_NAME	PARTITION_NAME	PARTITION_POSITION	HIGH_VALUE	INTERVAL
M_TEST	P_INIT	1	3/20/2017	NO
M_TEST	P_1	2	3/24/2017	NO

# A „hiányzó” interval partíció okozta problémák

TABLE_NAME	PARTITION_NAME	PARTITION_POSITION	HIGH_VALUE	INTERVAL
M_TEST	P_INIT	1	3/20/2017	NO
M_TEST	SYS_P2638585	2	3/21/2017	YES
M_TEST	SYS_P2638586	3	3/23/2017	YES
M_TEST	SYS_P2638587	4	3/24/2017	YES



```
ALTER TABLE m_test  
MERGE PARTITIONS  
    FOR (DATE '2017-03-20')  
TO FOR (DATE '2017-03-23')  
INTO PARTITION p_1;
```

ORA-02149: specified  
partition does not exist

```
ALTER TABLE m_test  
MERGE PARTITIONS  
    FOR (DATE '2017-03-20')  
, FOR (DATE '2017-03-22')  
, FOR (DATE '2017-03-23')  
INTO PARTITION p_1;
```

ORA-14274: partitions being  
merged are not adjacent

# A range-interval átmenet áthelyezése

TABLE_NAME	PARTITION_NAME	PARTITION_POSITION	HIGH_VALUE	INTERVAL
M_TEST	P_INIT	1	3/20/2017	NO
M_TEST	SYS_P2638585	2	3/21/2017	YES
M_TEST	SYS_P2638586	3	3/23/2017	YES
M_TEST	SYS_P2638587	4	3/24/2017	YES

```
ALTER TABLE m_test SET INTERVAL ( );
```

```
ALTER TABLE m_test
```

```
SET INTERVAL (NUMTODSINTERVAL (1, 'DAY' ) );
```

TABLE_NAME	PARTITION_NAME	PARTITION_POSITION	HIGH_VALUE	INTERVAL
M_TEST	P_INIT	1	3/20/2017	NO
M_TEST	SYS_P2638585	2	3/21/2017	NO
M_TEST	SYS_P2638586	3	3/23/2017	NO
M_TEST	SYS_P2638587	4	3/24/2017	NO

# Mi történik, ha a tömörítés és a betöltés egymásra futnak?

## Tömörítés

ALTER TABLE  
SET INTERVAL()

ALTER TABLE  
SET INTERVAL(1 Day)  
**Resource busy**

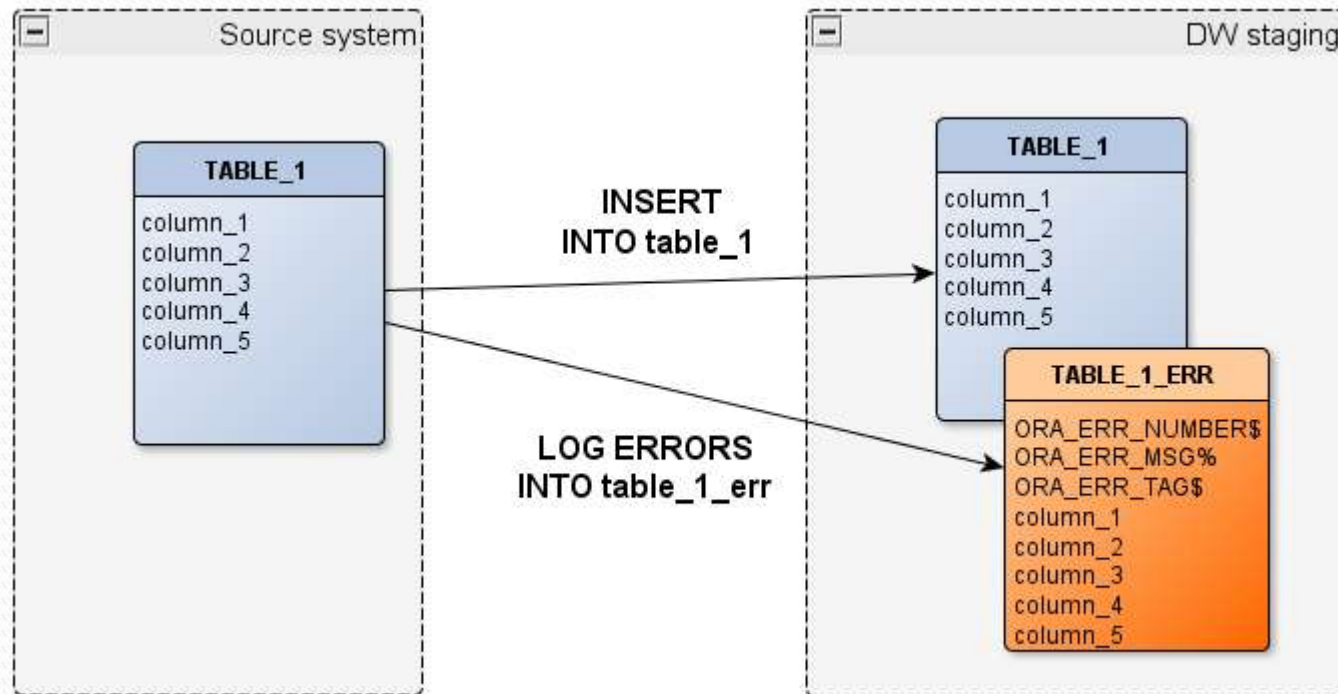
## Betöltés

INSERT INTO

INSERT INTO (next day)  
**Partition key does not map  
to any partition**

# DML error logging

- Hiányzó partíció esetén a rekordok az error táblába kerülnek
- Nehéz visszaállítani





# Hogyan tegyük a két ALTER utasítást „atomi” műveletté?

1. ALTER TABLE m\_test READ ONLY
2. ALTER TABLE m\_test SET INTERVAL ()
3. ALTER TABLE m\_test  
SET INTERVAL (NUMTODSINTERVAL (1, 'DAY' )
4. ALTER TABLE m\_test READ WRITE

Így a betöltés hibával megáll  
(az error tábla használata helyett):

- ORA-12081: update operation not allowed on table  
"M\_TEST"."M\_TEST"

A következő futásnál már működni fog, nincs teendő vele  
A tábla nem marad range particionált

# Mi történik, ha a tömörítés és a betöltés egymásra futnak?

Tömörítés



```
ALTER TABLE  
SET INTERVAL()  
Resource busy
```

Betöltés



```
INSERT INTO
```

- ORA-00054: resource busy and acquire with NOWAIT specified or timeout expired
- Megoldás:

```
ALTER SESSION SET DDL_LOCK_TIMEOUT=300
```

# Spatial adattípus

- SDO\_GEOMETRY
- Az interval particionálás nem támogatja
  - [http://docs.oracle.com/database/121/SQLRF/statements\\_7002.htm#SQLRF54559](http://docs.oracle.com/database/121/SQLRF/statements_7002.htm#SQLRF54559)

## *Restrictions on Interval Partitioning*

...

*This clause is not supported for tables containing nested table columns, varray columns, or XMLType columns.*

...

- A CREATE TABLE mégis működik
- De az ALTER már nem

# Spatial adattípus vs. Interval particionálás

```
SQL> CREATE TABLE p_test (  
SQL>     rld DATE  
SQL>     ,id NUMBER  
SQL>     ,geometry mdsys.sdo_geometry  
SQL> )  
SQL> TABLESPACE m_test  
SQL> PARTITION BY RANGE (rld)  
SQL> INTERVAL (NUMTODSINTERVAL(1,'DAY'))  
SQL> (  
SQL> PARTITION p_init VALUES LESS THAN (DATE '2017-03-01')  
SQL> );
```

Table created.

```
SQL> ALTER TABLE p_test SET INTERVAL();
```

Table altered.

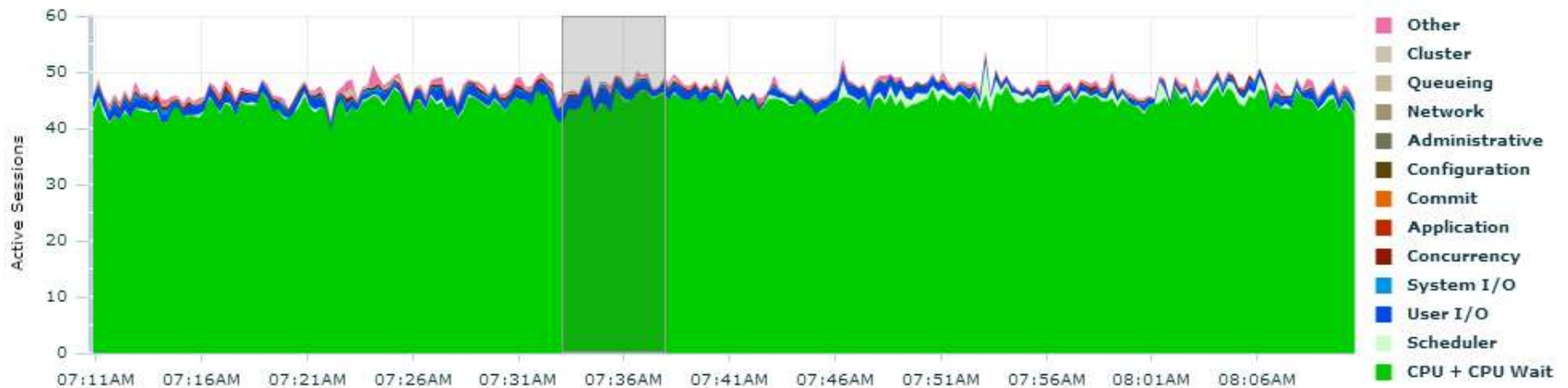
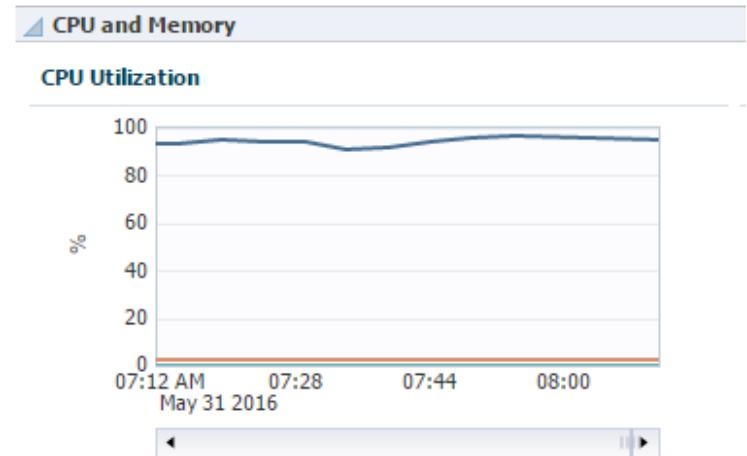
```
SQL> ALTER TABLE p_test SET INTERVAL(NUMTODSINTERVAL(1,'DAY'));  
ALTER TABLE p_test SET INTERVAL(NUMTODSINTERVAL(1,'DAY'))  
*
```

**ERROR at line 1:  
ORA-14769: cannot interval partition nested table or XMLType with  
object-relational storage**

```
SQL>
```

# Egyéb megfontolások

- A táblatéren és a disk groupon lévő szabad hely nem ugyanaz
- CPU-intenzív művelet
- 2 TB ~ 6 óra



# Eredmények

- Egyszeri tömörítés 2015-ben
- Automatizált megoldás 2016-ban
- 2,5 TB tárhely szabadult fel egyetlen rendszeren 2016 közepe óta
- Ez az összesen 6 (1 éles + 3 dev + 2 test) rendszer esetében már 15 TB
- Havonta fut
- Statisztikák:
  - ~ 1300 tábla tömörítve
  - ~ 650 esetben az adatbetöltések a mai napig futnak
  - 20-30 perc futásidő havonta
- Teljesen automatikus
- Log táblákon is használjuk a módszert

# Összegzés

- Az INSERT-only táblák a legalkalmasabb jelöltek a tömörítésre
- Interval particionálás
- 12c Partition merge
- OLTP és EHCC tömörítési módok vegyes használata
- DBMS\_SCHEDULER  
(figyelembe véve, hogy CPU-intenzív)
- 3X (OLTP) -12X (EHCC) tömörítési arányok
- Komplikációk az interval partíciók összevonásával  
(Alter set interval)
  - DDL lock vs. DML

# Kérdések?

