Partitioning in Oracle 12c

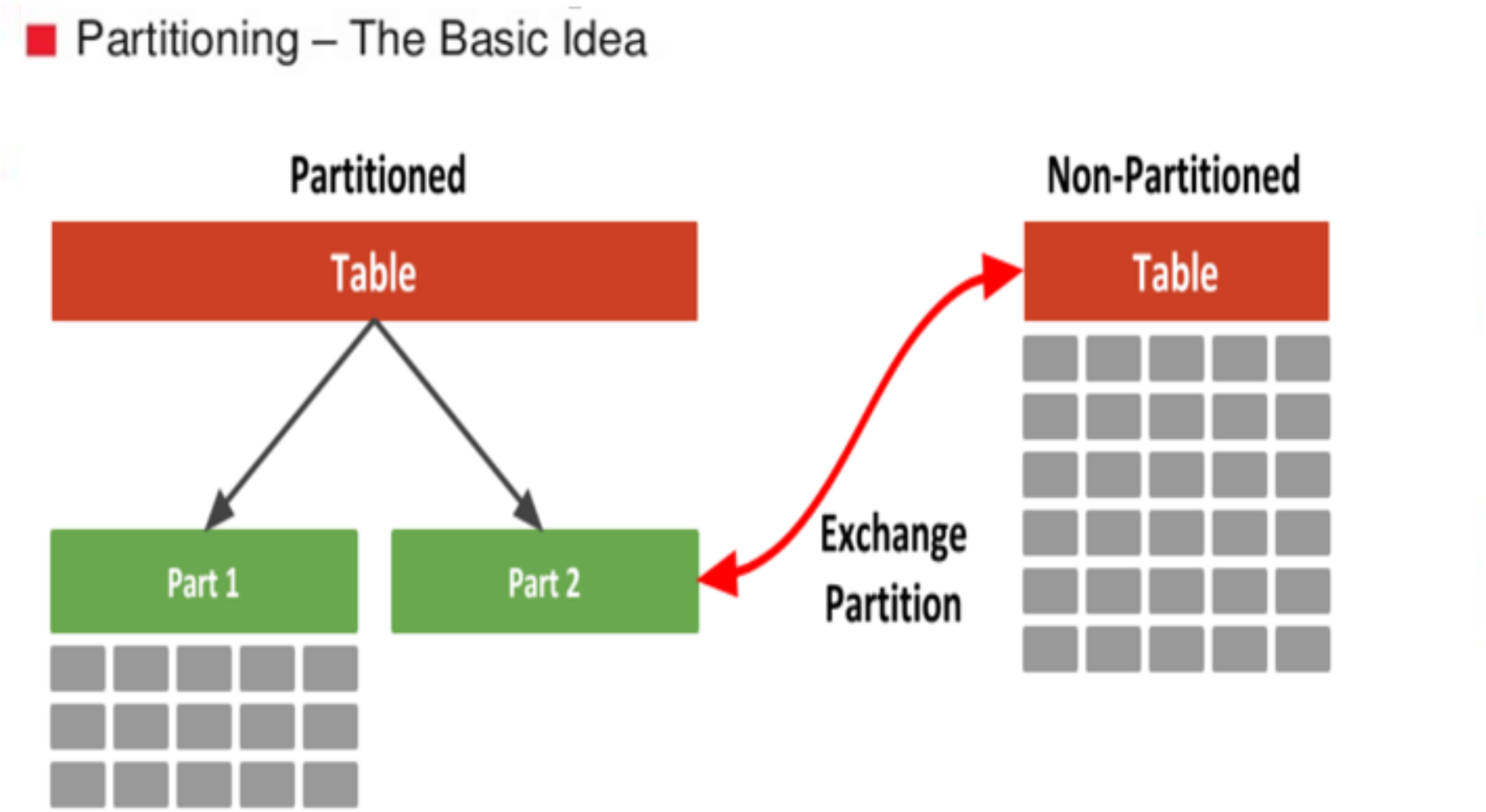
Bijaya K Pusty @ ADIENT

Partitioning in Oracle 12c

AGENDA

- Concepts of Partitttioning?
- Partitioning Basis
- Partitioning Strategy
- Additions | Improvments in 12c
- Partitioning Indexes and their usage
- Partitioning Pros & Cons
- Partition Exchange - Demonstration
- Q & A

❑ Concept of Partitioning?



❑ Basis of Partitioning:

Here comes the tough part:

How do we decide which tables should be partitioned ?
What columns to be used for partitioning ?
What partitioning methods/type should be used ?

Partitioning Key is important:

for Partition Pruning & Partition Join.
for ETL Performance (Partition Rollout/Exchange)

The choice of partitioning has to address key partitioning objectives for manageability and performance. Important factors to consider for the basis of partitioning.

- Performance
- Data purge
- Data archiving
- Ease of administration
- Data movement
- Efficiency of backup
- Data lifecycle management

❑ Partitioning Strategy:

How to determine what type of partitioning to Implement?

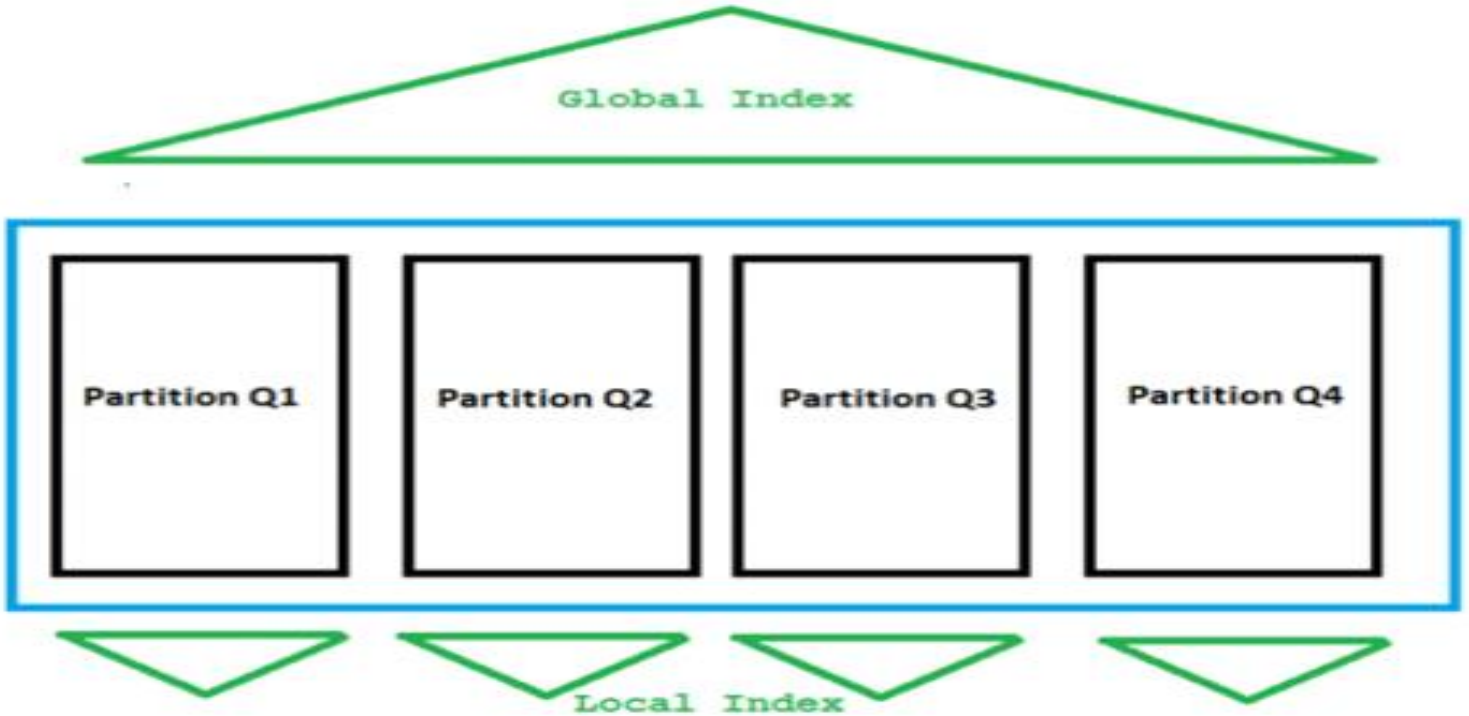
- ❑ Range: Range of values as partitioning key with (VALUES LESS THAN).
- ❑ List: More suits for a list of discrete values for the partitioning key.
- ❑ Hash: Suits for the non historical data or has no obvious key.
- ❑ Interval: Extensions to range partitions.(INTERVAL)
- ❑ Ref: Based on PK and FK relationships (PARTITION BY REFERENCE)
- ❑ Virtual col based: Suits for the non historical data or has no obvious key.
  - ❖ Single level partitioning
  - ❖ Composite partitioning

❑ Indexes for Partitioning:

- ❑ Just like partitioned tables, partitioned indexes improve manageability, availability, performance, and scalability. They can either be partitioned independently (global indexes) or automatically linked to a table's partitioning method (local indexes).

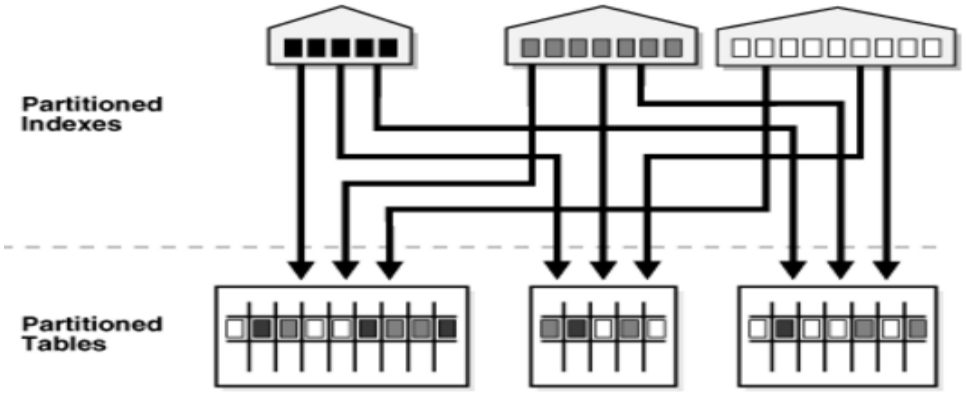
❑ ***Global Indexes:*** [Partitioned | non-partitioned]

❑ ***Local Indexes:***

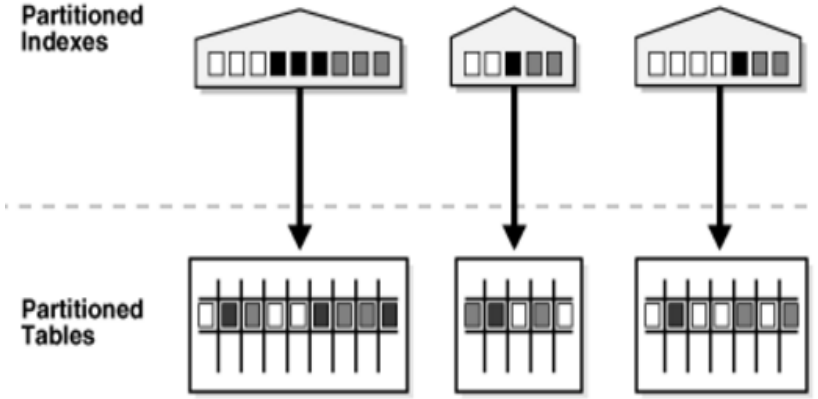


❑ Indexes for Partitioning:

❑ ***Global Indexes:*** [Partitioned | non-partitioned]



❑ ***Local Indexes:***



➢ Global partitioned index:

- Partitioned using a different partitioning-key or partitioning strategy than the table.
- These are indexes that span partitions, i.e. each partition in the index can be mapped to multiple underlying partitions.
- Used for OLTP applications.

➢ Global non-partitioned index:

- Essentially identical to an index on a non-partitioned table.

➢ Local index:

- Coupled with the underlying partitioned table; the index 'inherits' the partitioning strategy from the table.
- Each partition of a local index corresponds to one - and only one - partition of the underlying table.

- Used for DSS and Data W/H Applications.

➢ Indexes for Partitioning: [some TIPs]

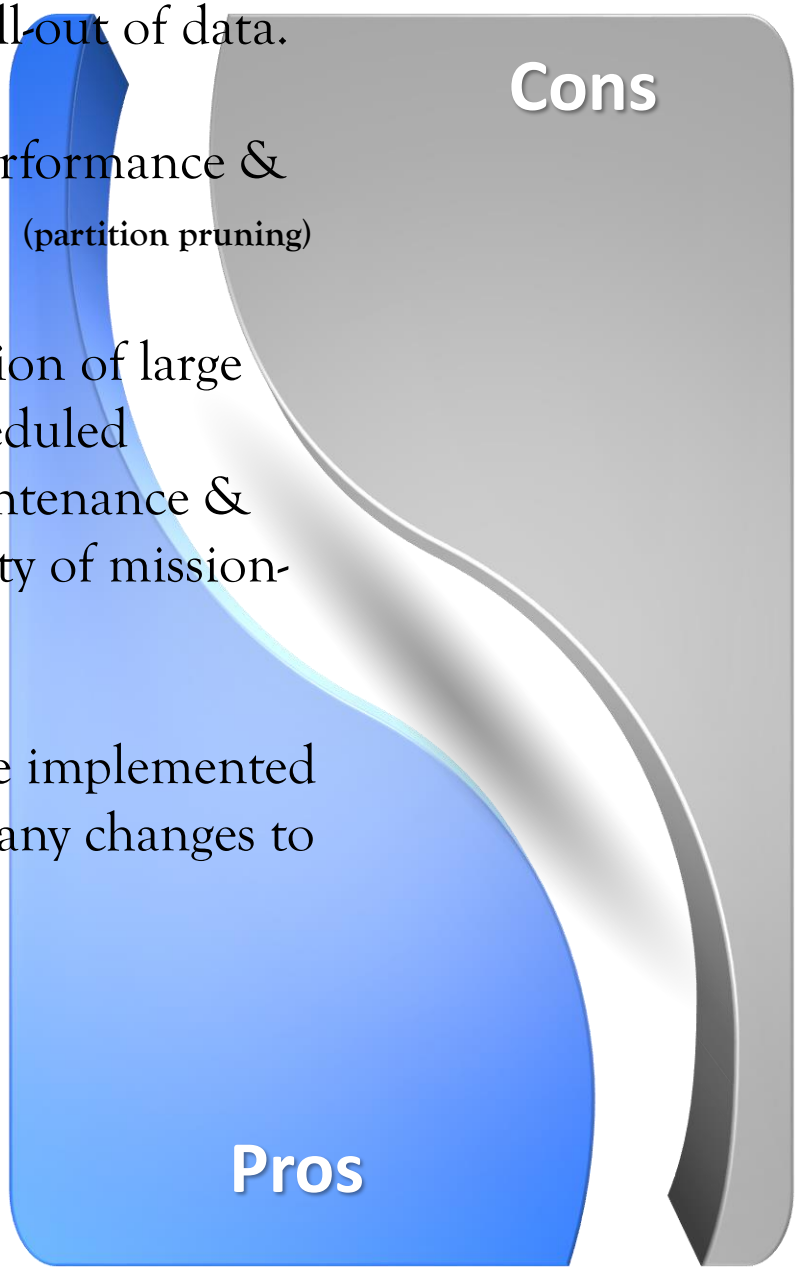
- Try to use local indexes wherever possible as they are easier to maintain and are useful for the partition pruning technique, which can help in performance improvement.

- Create bitmap indexes on partitioned tables, with the restriction that the bitmap indexes must be local to the partitioned table. They cannot be global indexes.

- When using partitioned indexes on composite partitions: Subpartitioned indexes are always local and stored with the table subpartition by default. Tablespaces can be specified at either index or index subpartition levels.

❑ Pros & Cons of Partitioning

- ❑ Easy roll-in and roll-out of data.
- ❑ Improves query performance & eliminate large i/o **(partition pruning)**
- ❑ Easier administration of large tables (reduce scheduled downtime for maintenance & increases availability of mission-critical databases)
- ❑ Partitioning can be implemented without requiring any changes to your applications



- ❑ The one that immediately comes to mind: it costs additional $$$.
- ❑ Unless you specify UPDATE INDEXES, any global indexes are marked UNUSABLE and must be rebuilt.
- ❑ Not a constraint but determining the correct type of partition and key on which to build the partition is important.

❑ Pros & Cons of Partitioning

- ❑ Easy roll-in and roll-out of data:
  - ➢ Enables data management operations such data loads, index creation, rebuilding, and backup/recovery at the partition level, rather than on the entire Table. This results in significantly reduced times for these operations.
- ❑ Partitioning improves query performance:

Pros & Cons of Partitioning

- Easier administration of large tables:
  - Significantly reduce the impact of scheduled downtime for maintenance operations.
  - Lets you perform concurrent maintenance operations on different partitions of the same table or index.
  - We can also perform concurrent SELECT & DML operations against partitions that are unaffected by maintenance operations.
- Can be implemented without any code change:
  - Non-partitioned table to a partitioned table without needing to modify application code.

❑ Additions

❑ Additions

12c Release 1

- Partial Indexes for Partitioned Tables
- Partition Maintenance Operations on Multiple Partitions.
- Asynchronous (Delayed) Global Index Maintenance for DROP and TRUNCATE Partition
- Cascade Functionality for TRUNCATE PARTITION and EXCHANGE PARTITION
- Interval-Reference Partitioning
- Online Move of Partitions and Sub-Partitions.

❑ Additions

12c Release 2

- Multi-Column List Partitioning
- Automatic List Partitioning
- Read-Only Partitions & Subpartitions
- Online Conversion of a Non-Partitioned Table to a Partitioned Table
- Create Table for Exchange With a Partitioned Table
- Filtered Partition Maintenance Operations
- Online Split Partition and Split Subpartitions
- Partitioned External Tables

❑ Additions

12c Release 2

- Partitioning an Existing Table using EXCHANGE PARTITION
- Partitioning an Existing Table using DBMS_REDEFINITION

❑ Additions

Some of the Demos

- We will see them in
  : Oracle 12 2 0 1

❑ Thanks You!

thank you
danke
謝謝
ngiyabonga
teşekkür ederim
gracias
tapadh leat
спасибо
Баярлалаа
рахмат
faafetai lava
dank je
mochchakkeram
go raibh maith agat
dziękuje
bedankt
obrigado
sukriya
kop khun krap
arigatō
dakujem
takk
grazie
terima kasih
merci
мерси
감사합니다
ありがとう
ευχαριστώ
hvala
mauruuru
sagolun
enkosi
хвала
asante
manana
trugarez
shukriya
dhanyavadagalu
diolch
rahmet
najis tuke
sobodi
dekuji
chnorakaloutioun
gracies
sulpay
taiku
mersi
barka
tack
vinaka
blagodaram
misaotra
matondo
paldies
grazzi
kiitos
dankie
dhanyavad
djiere dieuf
mamnun
xiexie
tanemirt
mesi

❑ Questions and Answers:

❑ References

http://www.oracle.com/technetwork/database/options/partitioning/partitioning-wp-12c-1896137.pdf

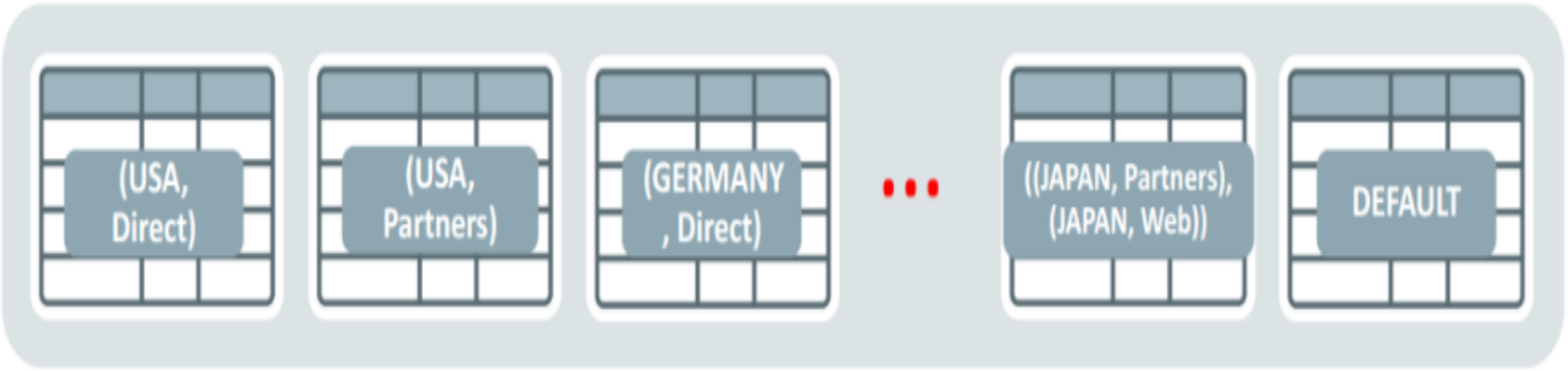
https://www.youtube.com/watch?v=W71G0H91n_k

https://www.slideshare.net/trivadis/partitioning-your-oracle-data-warehouse-just-a-simple-task

https://asktom.oracle.com/pls/asktom/f%3Fp%3D100:11:0::::P11_QUESTION_ID:1238800184155

https://docs.oracle.com/cd/A58617_01/server.804/a58227/ch_pti.htm

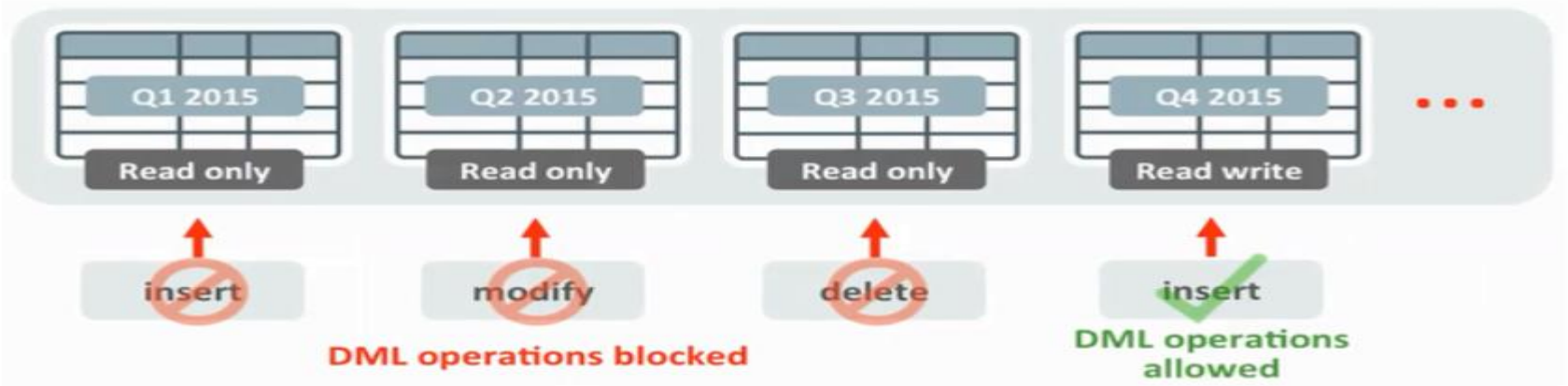
Multi-column list partitioning



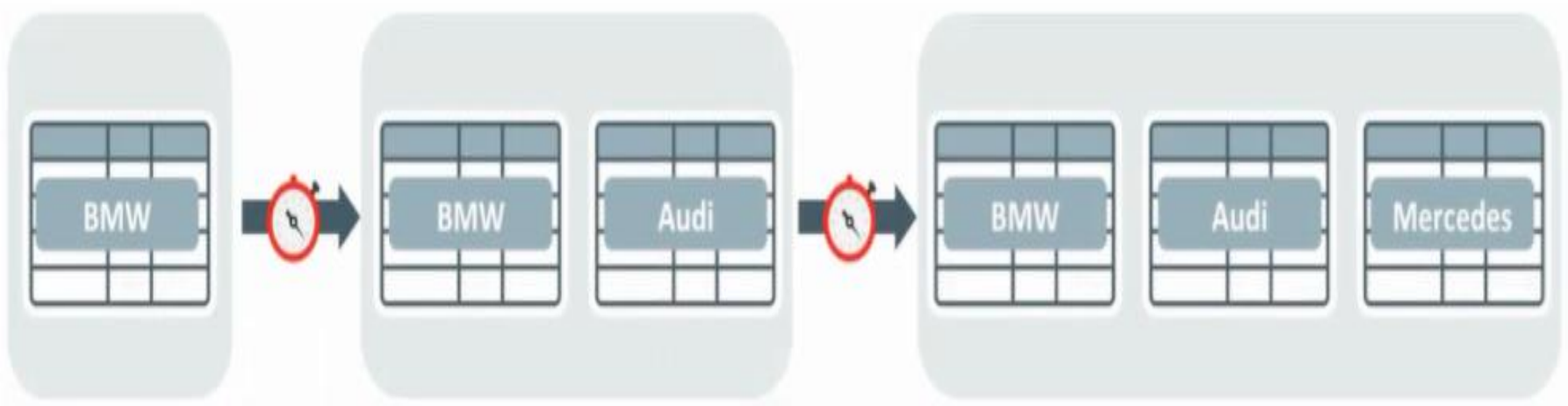
With the first release of 12c it wasn't possible to create list partitioned tables based on multi-column partition key :

ORA-14304: List partitioning method expects a single partitioning column

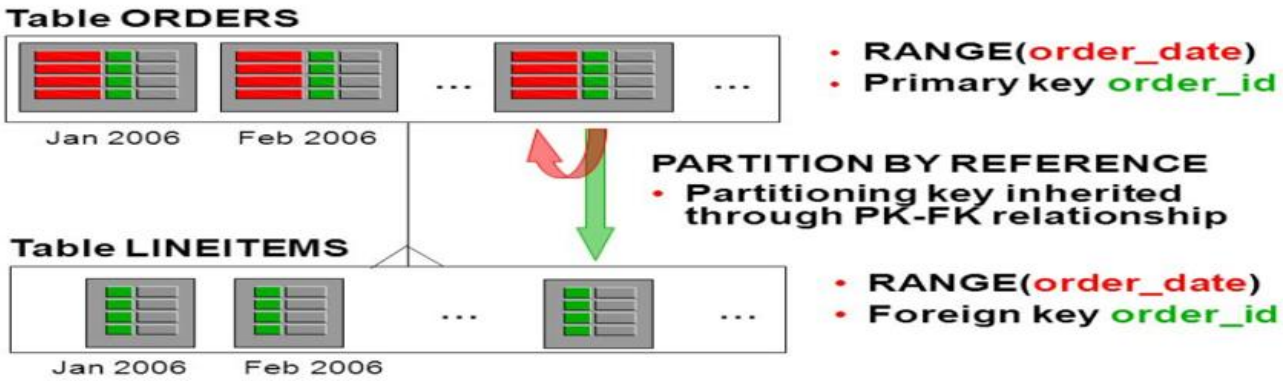
Read-only partitions



Auto-list partitioning is an extension of list partitioning. It enable the automatic creation of partitions for new values inserted into the partitioned table.



REF Partitioning



Interval Partitioning

How it works

- Interval partitioned table can have classical range and automated interval section
  - Automated new partition management plus full partition maintenance capabilities: ***"Best of both worlds"***

Table SALES