

Oracle19c混合分区表及分区技术演进

公益讲座11: 00准时开始, 请大家先浏览云技术微信公众号技术文章。资料会在各群同步发布, 已入群客户请勿重复入群!



20-21

数据库和云讲座群



甲骨文云技术公众号



B站专家系列课程



基于 Oracle 数据库 免费企业数据健康检查

- 及时了解数据库健康状况，发现并解决潜在问题
- 维护数据库系统良好状态，保护数据资产的安全
- 提升数据库性能、稳定性和安全性，降低业务风险

免费咨询热线：

400-699-8888

* 活动最终解释权归甲骨文公司所有

Oracle 19c

混合分区表及分区技术演进

甲骨文技术公益课 - 数据库专场

2023年7月14日 11:00

线上直播

阮蓉

内容

Oracle 19c混合分区表及分区技术演进

- 为什么需要分区表
- Oracle 分区技术发展过程
- Oracle混合分区表

ORACLE

为什么需要分区表

—

为什么需求分区



高可用

大表跨越多个磁盘。即使其中一个存储卷发生故障，整个表也会变得不可用。

数据加载、索引创建和重建。
并且备份和恢复非常昂贵且耗时

可管理性

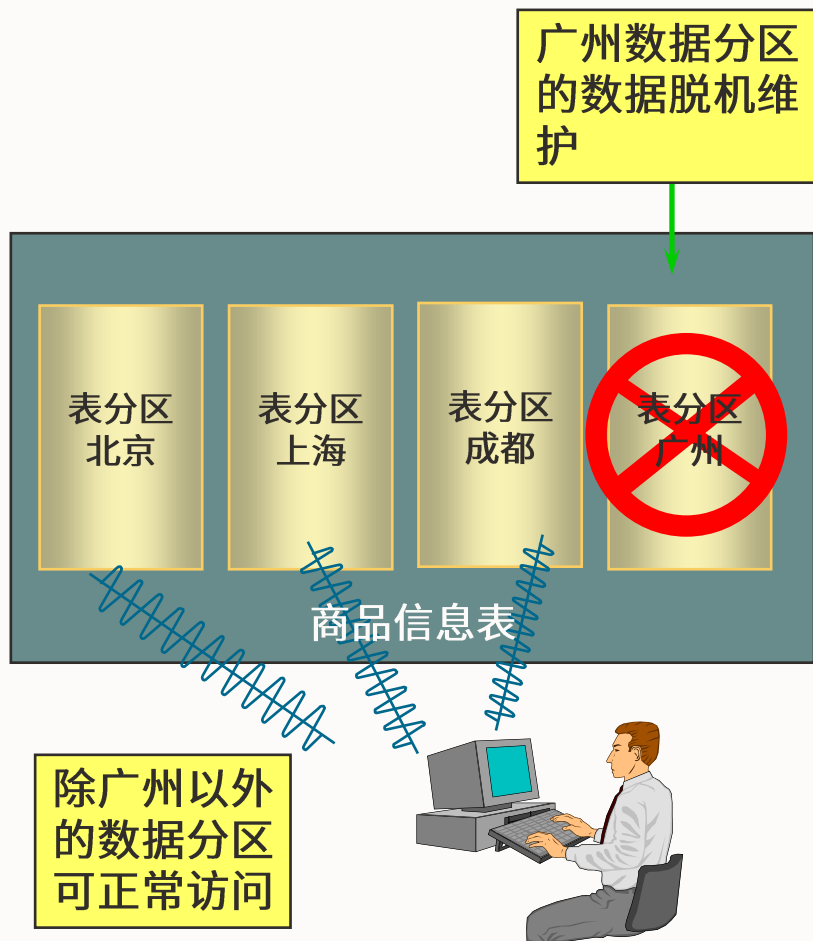


性能

大表和大索引扫描的成本很高



提高关键数据的可用性



- 数据损坏的故障隔离
- 支持在线增加、删除
- 在特定分区上的批处理
- 按分区备份
- 在维护过程中可以继续访问不受影响的分区
- 快速灾难恢复：恢复最关键的数据分区

表分区提高大表的可管理性

- 范围、hash、列表、复合分区

删除分区

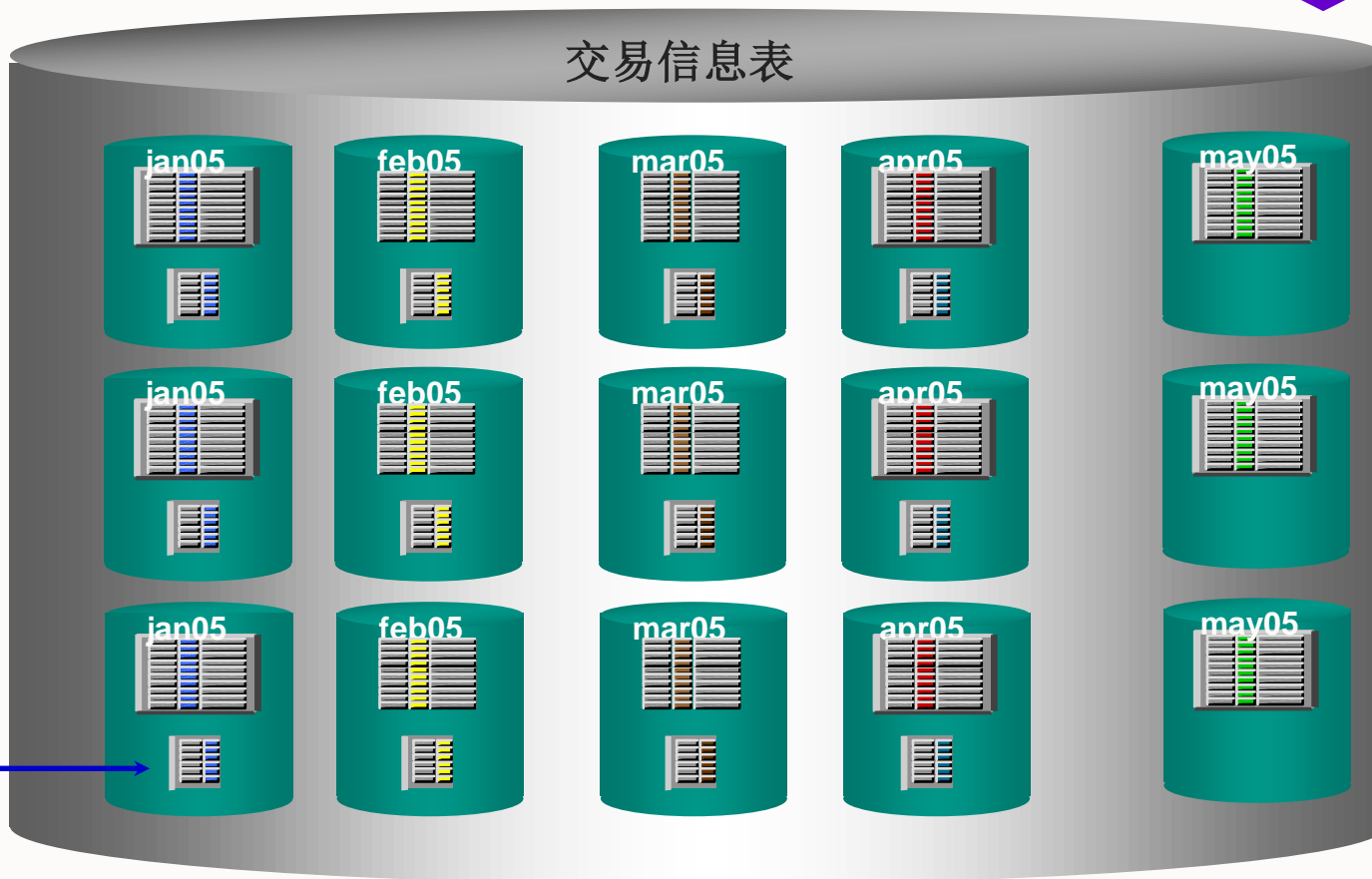
添加分区

北京

上海

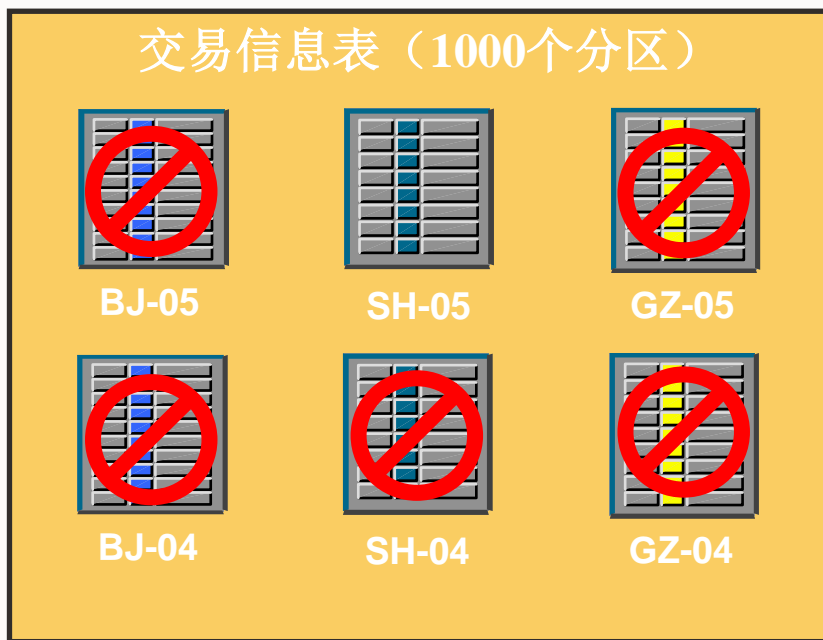
广州

本地索引



极大地提高大数据量的访问速度

查询上海市某企业在5月份的某笔交易的详细信息
(提高性能近1000倍)



- ❑ 分区对应用是透明的
- ❑ 自动忽略与访问无关的分区
- ❑ 实现分区间的并行查询
- ❑ 显著提高访问大表时的性能

500个城市12个月的数据

分区忽略

系统自动根据分区情况优化数据访问，忽略无关的数据分区

为什么需要分区



1.提高性能(主要原因)

SQL语句的where子句会过滤掉不需要的分区，不会再扫描那些不需要的分区。

可以把不同的分区映射到磁盘以平衡I/O，改善整个系统性能。

2.管理维护方便，数据维护只操作某些分区即可。

如果表的某个分区出现故障，表在其他分区的数据仍然可用。

可以只对单独分区进行数据加载、数据备份、数据恢复以及索引重建等操作。

3.对应用透明，无需业务更改

在数据库级完成，不需要应用程序做修改。

4.有助于其他扩展需求

分区有利于数据库数据的过期化处理；OLAP并行处理；只读数据业务需求。

5.Sharding

6.数据生命周期管理



ORACLE

Oracle 分区技术发展过程

—

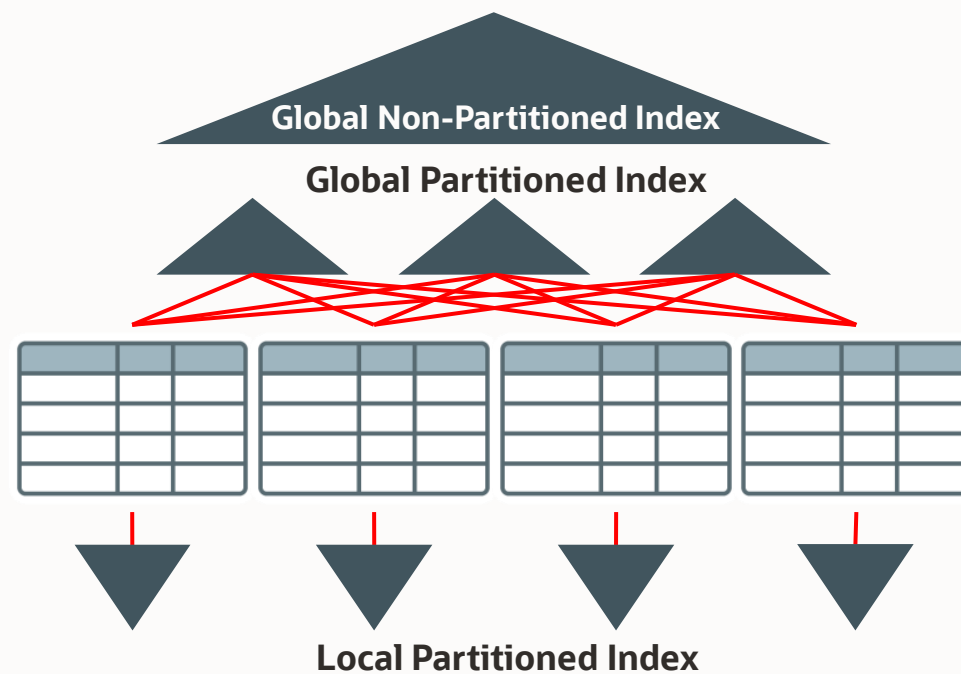


Oracle 数据库的分区技术发展

	分区方法	Performance	Manageability
Oracle 8.0	Range partitioning Local and global Range indexing	Static partition pruning	Basic maintenance: ADD, DROP, EXCHANGE
Oracle 8i	Hash partitioning Range-Hash partitioning	Partition-wise joins Dynamic partition pruning	Expanded maintenance: MERGE
Oracle 9i	List partitioning		Global index maintenance
Oracle 9i R2	Range-List partitioning	Fast partition SPLIT	
Oracle 10g	Global Hash indexing		Local Index maintenance
Oracle 10g R2	1M partitions per table	Multi-dimensional pruning	Fast DROP TABLE
Oracle 11g	Virtual column based partitioning /More composite choices /Reference partitioning		Interval partitioning /Partition Advisor /Incremental stats mgmt
Oracle 11g R2	Hash-* partitioning Expanded Reference partitioning	“AND”pruning	Multi-branch execution (aka tableor-expansion)
Oracle 12c R1	Interval-Reference partitioning	Partition Maintenance on multiple partitions Asynchronous global index maintenance	Online partition MOVE /Cascading TRUNCATE/ Partial indexing
Oracle 12c R2	Auto-list partitioning Multi-column list [sub]partitioning Read-Only Partition	Online partition maintenance operations Online table conversion to partitioned table Reduced cursor invalidations for DDL's	Filtered partition maintenance operations Read only partitions Create table for exchange
Oracle 18c	Partitioned external tables Modifying the Partitioning Strategy	Parallel partition-wise SQL operations Completion of online partition maintenance Enhanced online table conversions	Validation of Data Content
1 Oracle 19c	Hybrid Partitioned tables		Object Storage Access

那些对象能够做分区?

- 表
 - Heap tables
 - Index-organized tables
- 索引
 - Global Indexes
 - Local Indexes
- 物化视图 Materialized Views
- 哈希集群 Hash Clusters



分区方法

- 单一（一级）分区
 - 范围Range
 - 列表 List
 - 哈希 Hash
- 组合分区
 - [Range | List | Hash | Interval] – [Range | List | Hash]

分区扩展

间隔分区 Interval

引用分区 Reference

间隔引用分区 Interval Reference

基于虚拟的列 Virtual Column Based

自动列表分区 Auto



范围分区



- 数据按范围组织
- 在必要的时候Split 和merge
- 每个分区定义了上限值
- 非常适合按时间顺序排列的数据



列表分区



数据按照列表值的形式组织

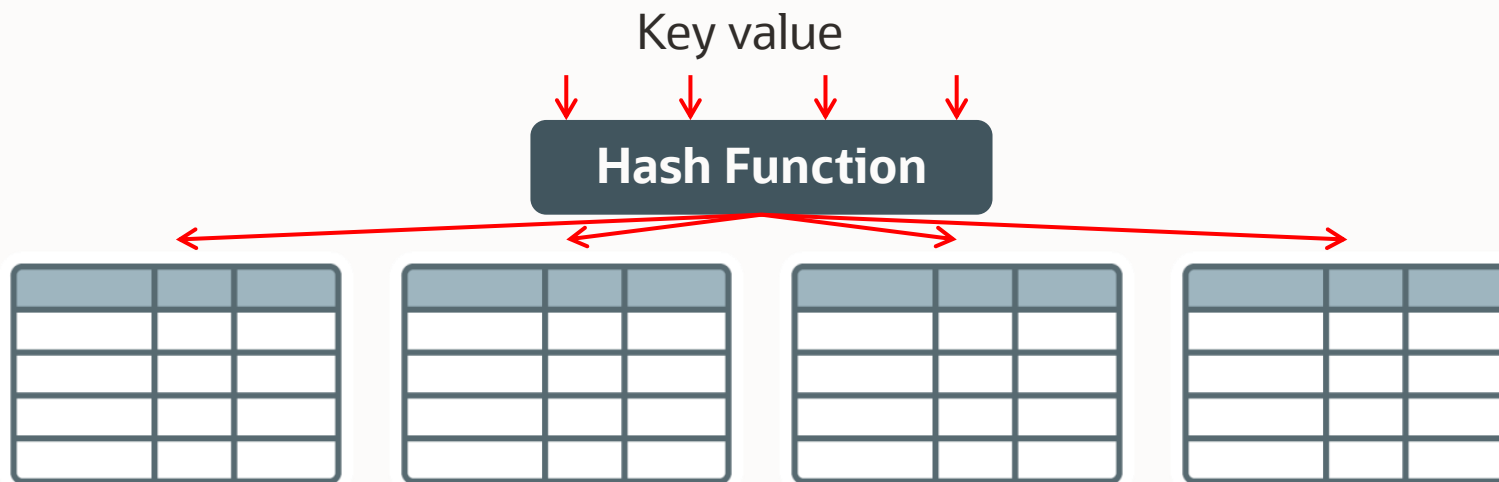
- 每个分区存储预先定义的键值
- 每个列表有一个或多个无序的不同值
- DEFAULT 分区的功能（捕获所有未指定值）
- 检查 DEFAULT 分区的内容 - 根据需要创建新分区

非常适合不同值的分割，例如 地区



哈希分区

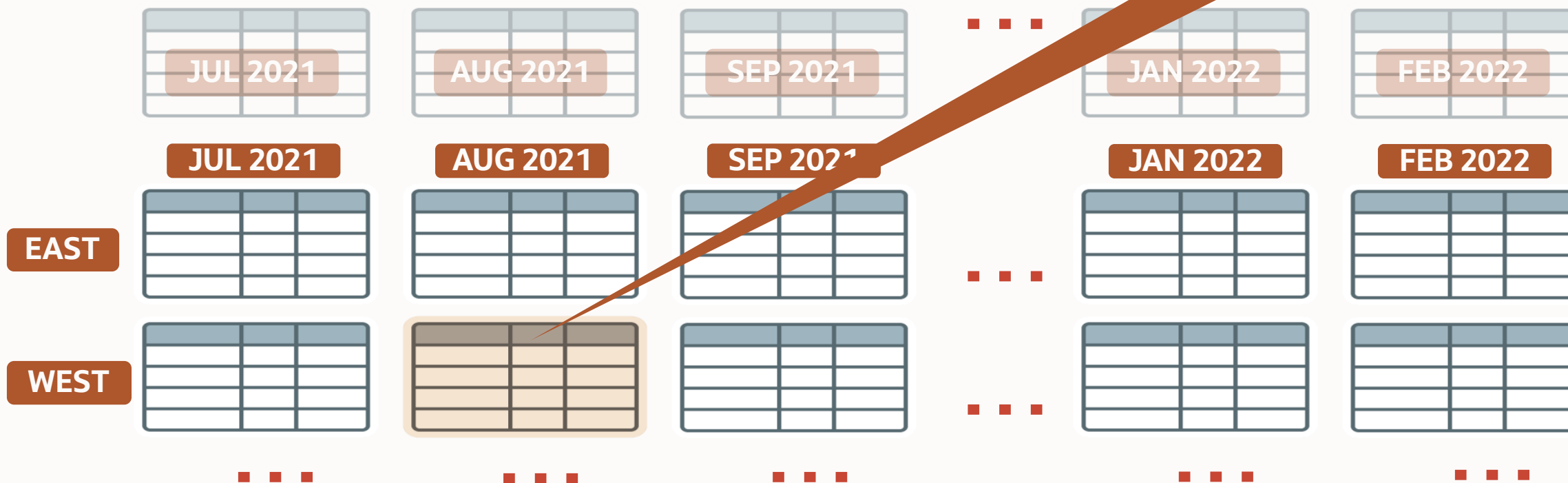
- 数据按分区键的哈希值分布
- 对分区键应用一个散列函数，以此确定数据应当放在 N 个分区中的哪一个分区中
- 平均数据分布的理想选择
- 分区数应为 2 的幂（2、4、8、16 等），以实现最佳的总体数据分布



组合分区

数据按照两个分区维度组织

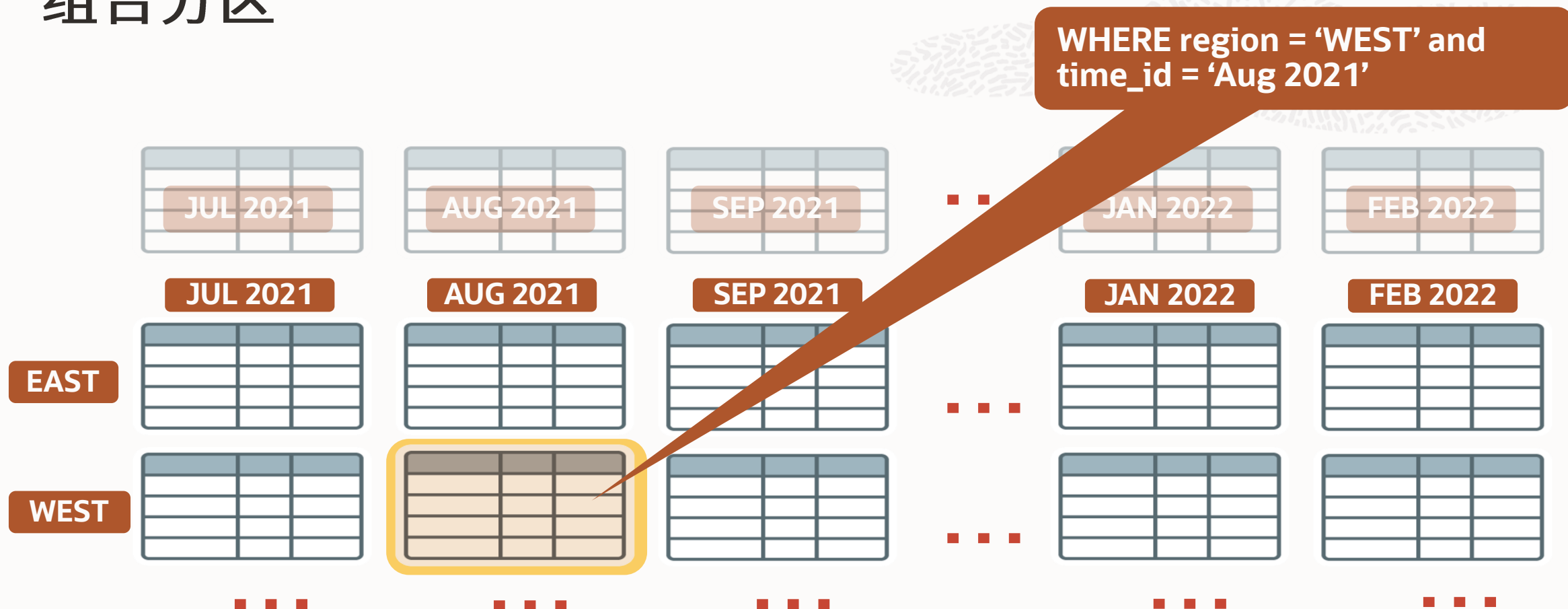
- 范围-列表组合分区例子—数据按一级两时间范围和二级范围分区组织



```
CREATE TABLE EVENTS ..PARTITION BY RANGE (time_id)
                    SUPARTITION BY LIST (region)
```



组合分区

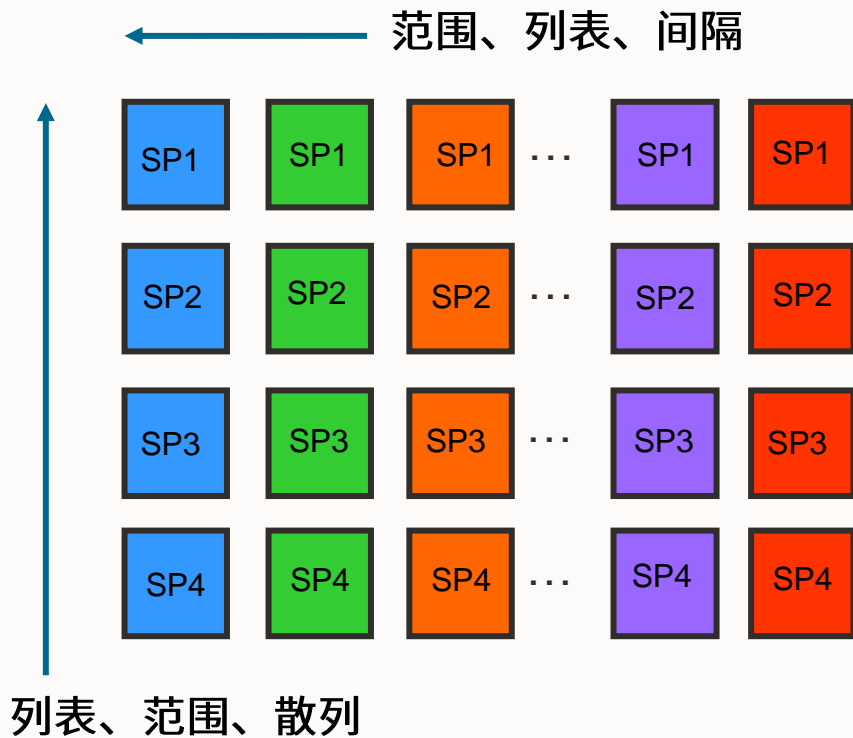


分区裁剪与组合顺序无关



组合分区

- 范围顶级
 - 范围-范围
 - 范围-列表
 - 范围-哈希
- 列表顶级
 - 列表-列表
 - 列表-散列
 - 列表-范围
- 间隔顶级
 - 间隔-范围
 - 间隔-列表
 - 间隔-散列



间隔分区

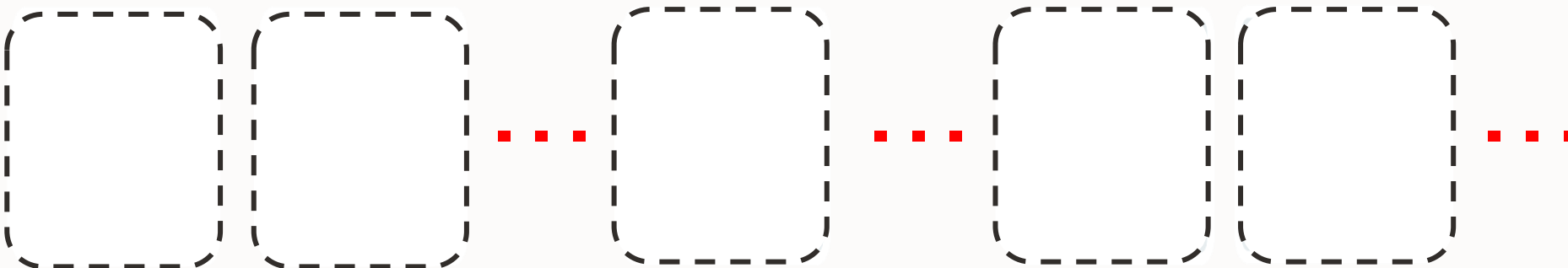
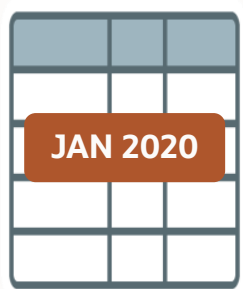
当数据插入时自动创建分区

- 范围分区的扩展



间隔分区

非常简单



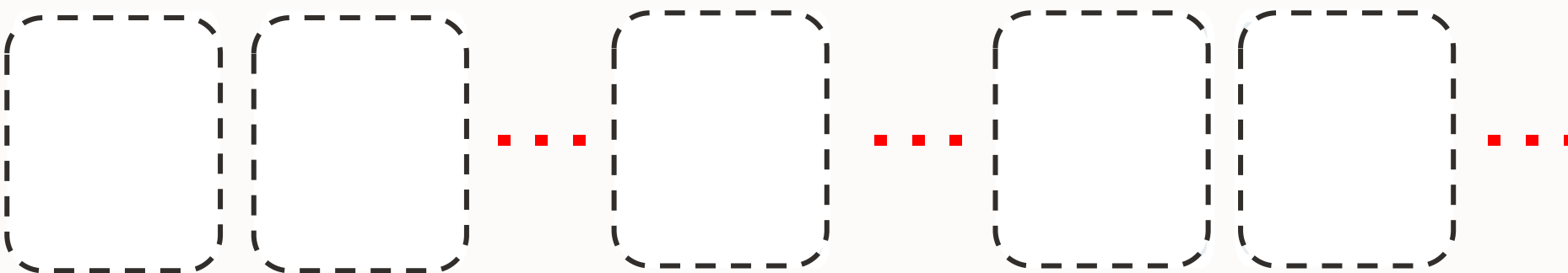
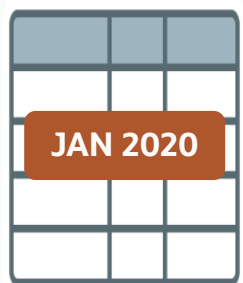
↑
**First
partition
is created**

```
CREATE TABLE EVENTS (order_date DATE, ...)  
PARTITION BY RANGE (order_date)  
INTERVAL (NUMTOYMINTERVAL(1, 'month'))  
(PARTITION p_first VALUES LESS THAN ('01-FEB-2020'));
```



间隔分区

非常简单

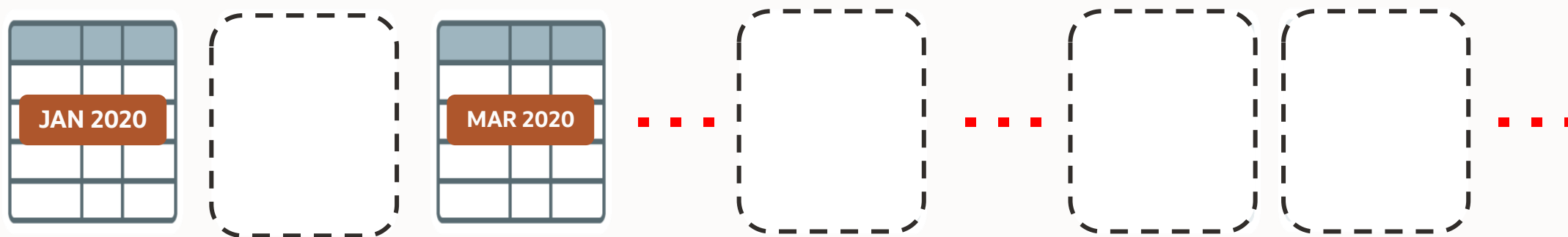


Other partitions only exist in table metadata



间隔分区

非常简单

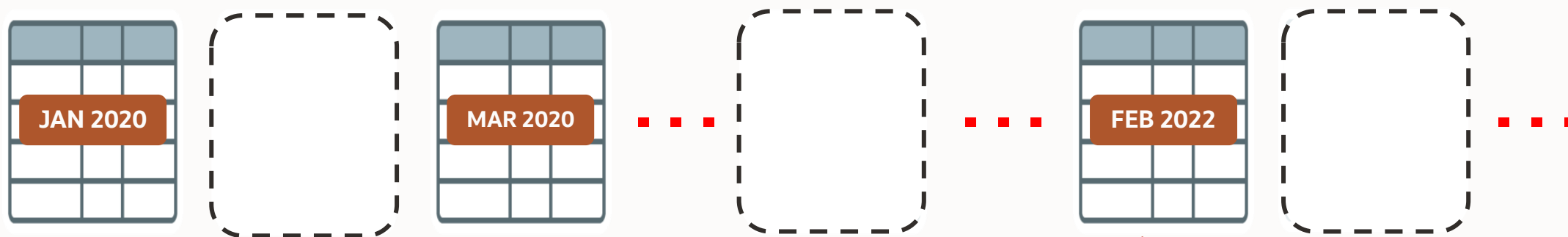


新分区是
自动实例化

```
INSERT INTO EVENTS (order_date DATE, ...)
VALUES ('15-MAR-2020', ...);
```


间隔分区

非常简单



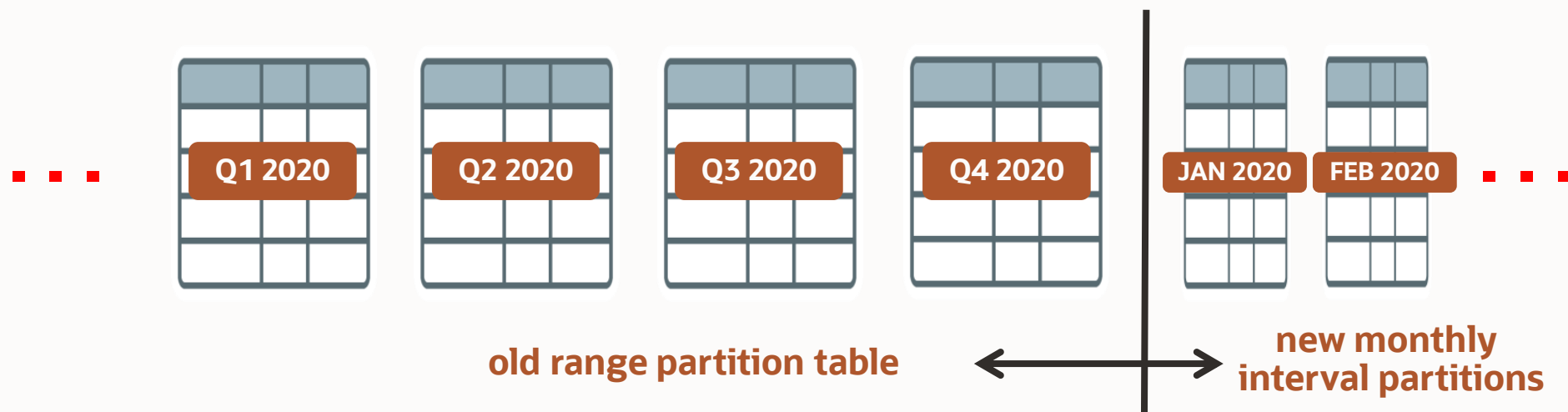
每当数据插入时
新分区会自动创建

```
INSERT INTO EVENTS ( order_date DATE, ... )  
VALUES ('04-FEB-2022', ...);
```



间隔分区

对于范围分区表的原表可以通过命令更改为间隔分区



```
ALTER TABLE EVENTS  
SET INTERVAL (NUMTOYMINTERVAL(1, 'month'));
```

自动列表分区Auto-List Partitioning



当数据插入时自动创建分区

- LIST 分区的扩展
- 每个不同的分区键值将存储在单独的分区中

自动列表分区表



```
CREATE TABLE EVENTS( sensor_type VARCHAR2(50),  
                      channel   VARCHAR2(50), ...)  
PARTITION BY LIST (sensor_type) AUTOMATIC  
( partition p1 values ('GYRO'));
```



引用分区：优点

不使用引用分区

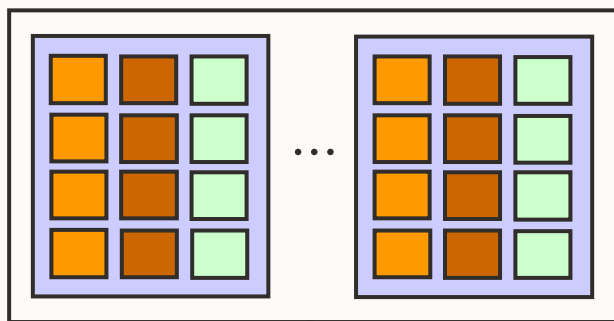
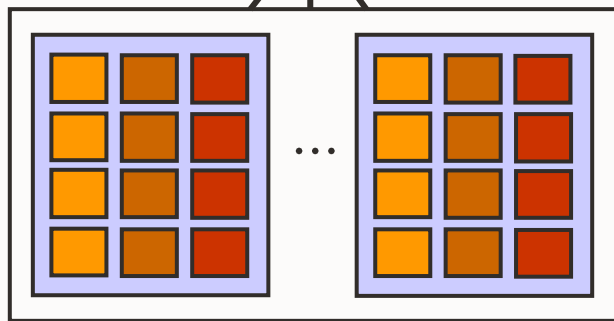


表 ORDERS

表
ORDER_ITEMS

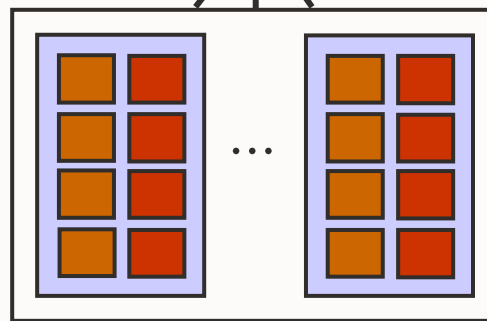
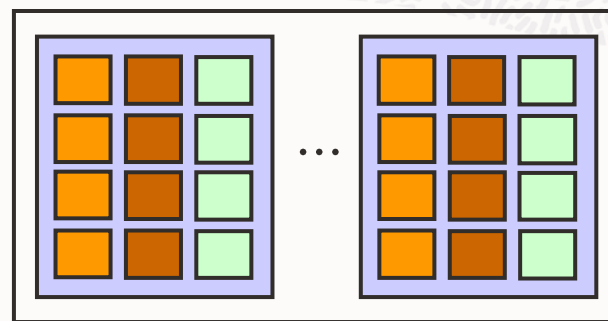


ORDER_DATE 的冗余存储/维护

范围 (ORDER_DATE)
主键 (ORDER_ID)

范围 (ORDER_DATE)
外键 (ORDER_ID)

引用分区



通过 PK/FK 关系继承的分区键



引用分区：示例

```
CREATE TABLE orders
( order_id      NUMBER(12) , order_date  DATE,
  order_mode    VARCHAR2(8) , customer_id NUMBER(6) ,
  order_status  NUMBER(2)   , order_total NUMBER(8,2) ,
  sales_rep_id  NUMBER(6)   , promotion_id NUMBER(6) ,
  CONSTRAINT   orders_pk PRIMARY KEY(order_id)
)
PARTITION BY RANGE(order_date)
(PARTITION Q105 VALUES LESS THAN (TO_DATE('1-4-2005','DD-MM-YYYY')) ,
 PARTITION Q205 VALUES LESS THAN (TO_DATE('1-7-2005','DD-MM-YYYY')) ,
 PARTITION Q305 VALUES LESS THAN (TO_DATE('1-10-2005','DD-MM-YYYY')) ,
 PARTITION Q405 VALUES LESS THAN (TO_DATE('1-1-2006','DD-MM-YYYY')));
```

```
CREATE TABLE order_items
( order_id      NUMBER(12) NOT NULL, line_item_id NUMBER(3) NOT NULL,
  product_id    NUMBER(6) NOT NULL, unit_price     NUMBER(8,2) ,
  quantity      NUMBER(8) ,
  CONSTRAINT   order_items_fk
              FOREIGN KEY(order_id) REFERENCES orders(order_id)
) PARTITION BY REFERENCE(order_items_fk);
```



分区原则

分区表的考量

- 表大于 2 GB。这些表应始终被视为分区的候选表。
- 表包含历史数据，其中新数据添加到最新的分区中。
一个典型的例子是历史表，其中只有当前月份的数据可更新，其他 11 个月的数据是只读的。
- 大表数据需要分布在不同类型的存储设备上。
- 分区的字段要是经常用以提取数据的字段，否则会在提取过程中导致遍历多个分区。
- 分区字段要选择合适，数据较为均匀分布到各个分区。
- 分区的方式根据实际环境而定。



分区设计的挑战



- 对于海量数据库，分区设计至关重要
- 分区表可能带来积极和消极的影响
- 数据库架构师一项关键的技能是，设计分区策略，使得查询性能和数据维护不会数据规模的影响
- 现实中，很多系统的分区策略是被个人经验爱好随意确定下来；好的架构师能够根据业务负载，评估多种分区策略，选取最优的方案



常见的分区策略



- 大表按时间范围分区
- 分区字段应该在大部分的查询中用来做数据过滤条件
- 该分区字段不应该经常有改动，分区字段的改动可能导致数据在不同的分区中移动。
- 频繁的分区字段改动通常意味着糟糕的数据设计
- 考虑二级分区方式
- 二级哈希分区以支持关联、排序和布隆过滤
- 二列范围和列表分区已支持更细致的分区裁剪

分区策略：常见的错误



- 日期范围/间隔分区太大或者太小
 - Exadata一体机的Io请求单位是 1MB. 我们经常看到分区大小小于1MB的情况.
- 范围/间隔分区策略导致分区大小极度不均衡
- 哈希分区的大小应该总是2的幂次方 比如： 8, 16, 32, 64, 128等.
 - 这是为了确定分区数据均衡
 - 分区字段应该有大量的唯一值确保均衡的分布，通常是主键或者外键
- 哈希分区的个数跟以下因素有关：
 - 超过默认的并行度两倍
 - 是的基于分区级别的哈希关联成为可能
- 分区技术通常被过度的的用来解决索引竞争
- 分区数过多
 - Oracle支持一个表或者索引最多 1,048,575个分区
 - 当分区数超过10000之后，DML, DDL或者其他数据字典的操作会对性能导致显著的负面影响



分区裁剪的用处



- 减少需要扫描的数据，降低处理时间
- 取决具体的SQL，Oracle支持静态和动态的分区裁剪
- 分区裁剪影响对象级别的统计信息和SQL的执行计划
- 对于组合分区，只要有合适的过滤条件，分区裁剪可以在两个分区维度都发生
- 对于引用分区表，分区裁剪可以通过表关联触发
- 虚拟字段作为分区键也可使用分区裁剪

分区裁剪的建议：如何获取最大的收益

- SQL语句尽量使用静态分区裁剪，性能比动态裁剪好
- 避免扫描数据时对数据做类型转换，会导致分区裁剪无法触发
- 避免在分区字段使用显示或者隐式的函数或者表达式 如果确实需要，可使用虚拟列作为分区键.
- Oracle收集表(Collection tables)可能限制优化器使用分区裁剪

分区表在线操作



分区维护在线完成

Move: change location and storage attributes

Merge: many partitions become one

Split: one partition becomes many



在线进行表转换操作

Modify nonpartitioned table to become partitioned table

Change shape of partitioned table



所有在线操作均支持索引维护



所有的分区维护操作现在都可以Online

在18c和19c中**在线功能**进一步增强，减少联机 DDL 对正在运行的系统的影响

11.2 &
Prior

**Create index online, rebuild index online, rebuild index partition online
Add Column, Add Constraint enable novalidate**

12.1

Online move partition
Drop index online
Set unused column online, alter column visible/invisible,
alter index unusable online, alter index visible/invisible
alter index parallel/noparallel

12.2

Alter table move online for non-partitioned tables
Alter table from non-partitioned to partitioned online
Alter table split partition online
Create table for exchange (usable for online partition exchange)
Move/merge/split partition maintenance operations can now do data filtering

18c,19c

Alter table modify partitioned table to a different partitioning method (e.g., hash to range)
Alter table merge partition/subpartition online

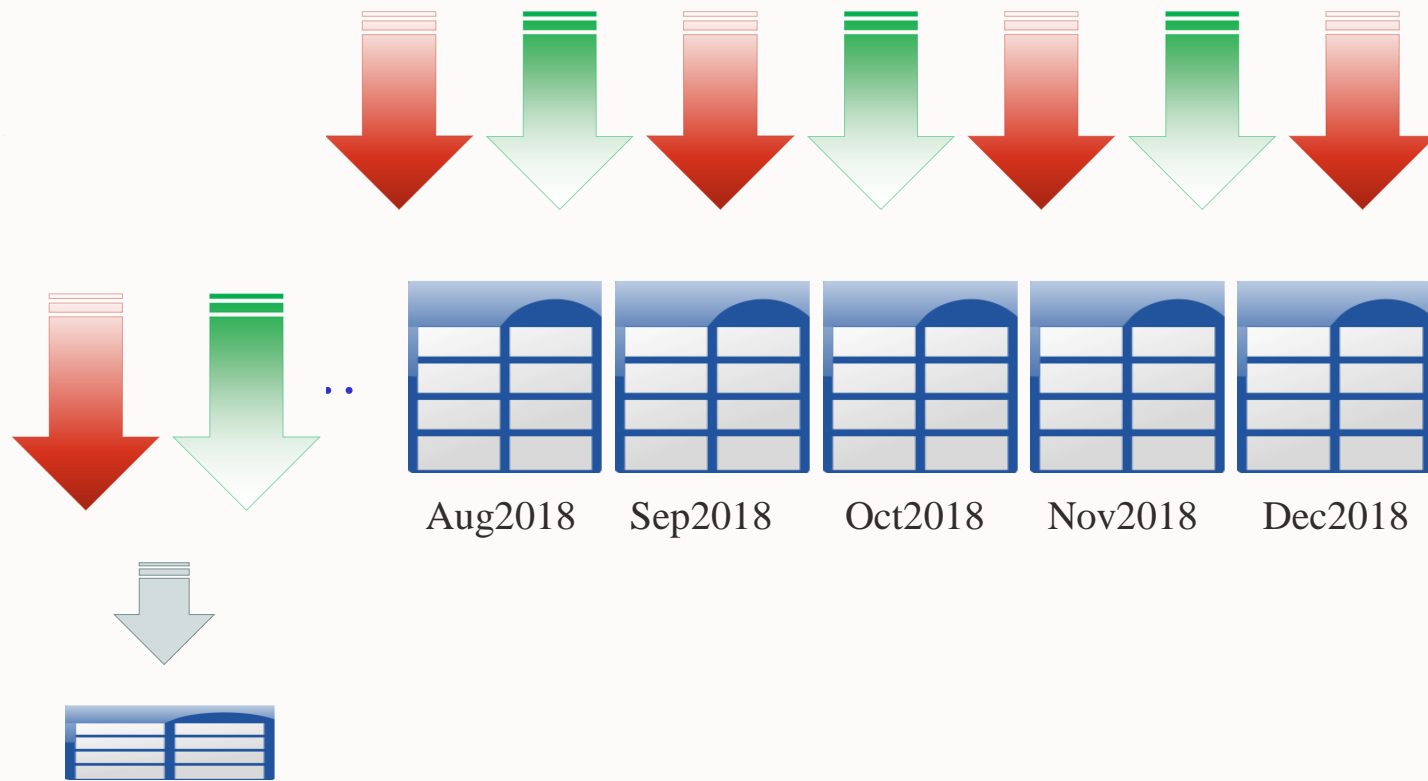


在线分区移动

透明的 MOVE PARTITION
ONLINE操作

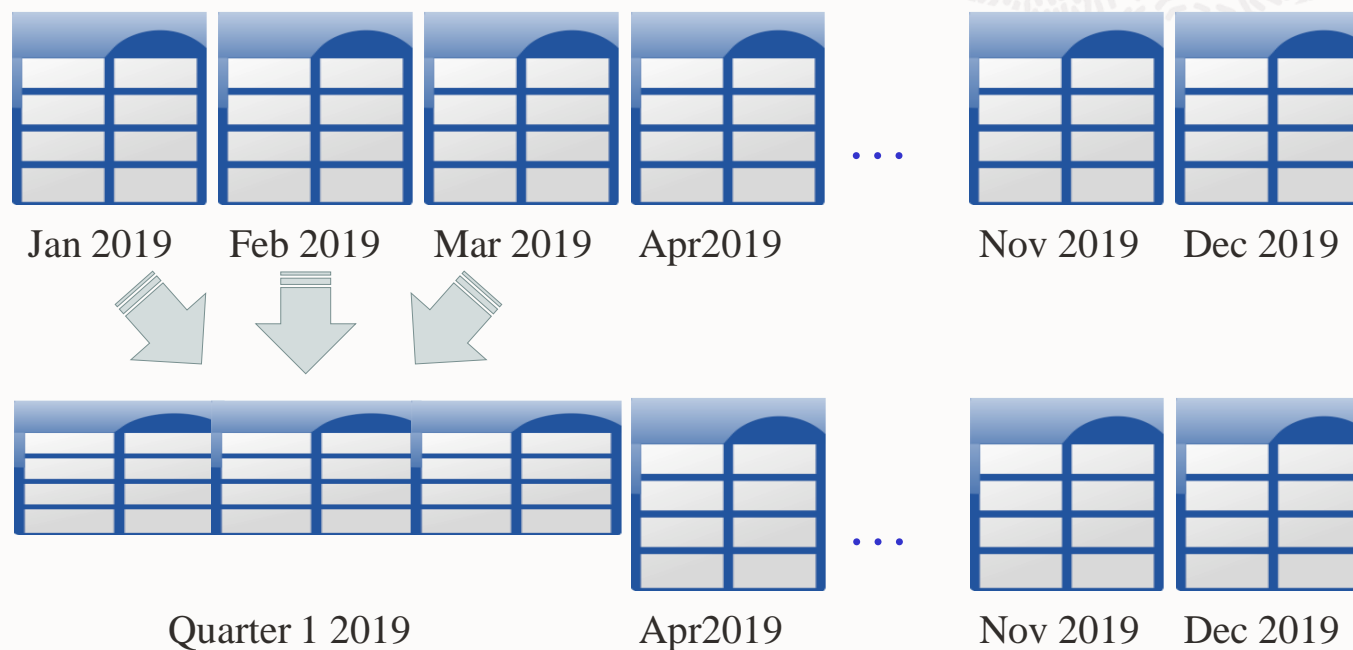
并发的DML和查询

对于本地和全局索引的索引
维护



在线分区移动

在多个分区上的操作
在一个操作中维护多个分区
全部并行
本地和全局索引的透明维护

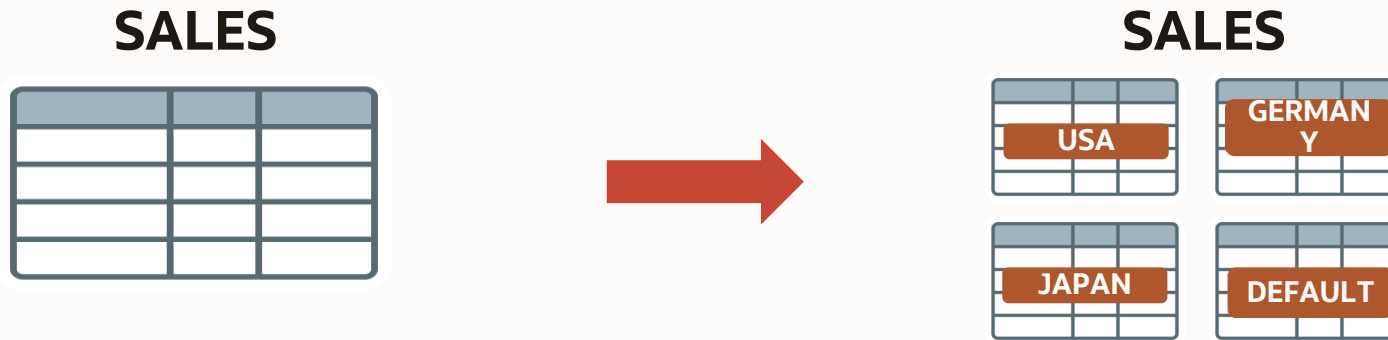


```
ALTER TABLE orders  
MERGE PARTITIONS Jan2019, Feb2019, Mar2019  
INTO PARTITION Quarter1_2019 COMPRESS FOR ARCHIVE HIGH;
```



在线表格转换

非分区表转换为分区表



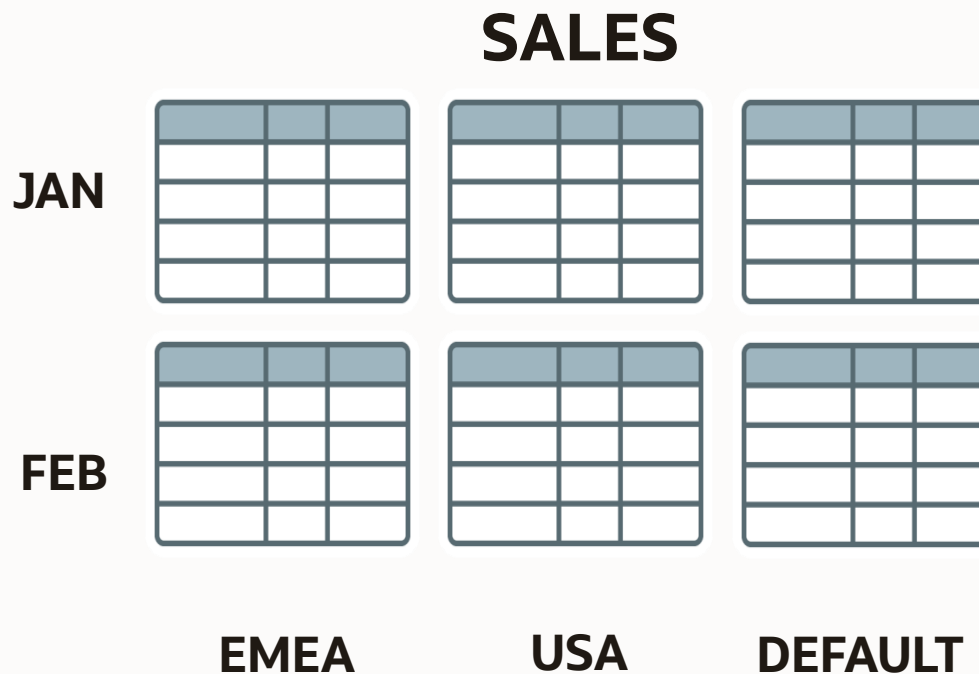
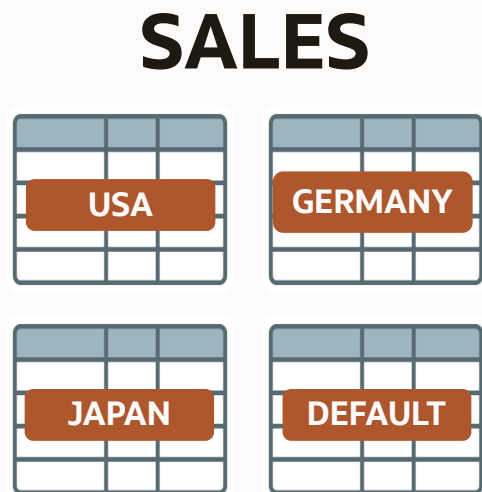
```
CREATE TABLE sales ( order_num NUMBER,
                      region VARCHAR2 (10), ... );
```

```
ALTER TABLE sales MODIFY
PARTITION BY LIST (region)
  (partition p1 values ('USA'),
   partition p2 values ('Germany'),
   partition p3 values ('Japan'),
   partition p4 values (DEFAULT))
UPDATE INDEXES ONLINE;
```

在线表转换

Partitioned Tables

表和索引的完全非阻塞（在线）DDL



在线表转换

Partitioned Table

```
CREATE TABLE sales ( order_num NUMBER,  
                      region VARCHAR2 (10), ... )  
PARTITION BY LIST (...)
```

```
ALTER TABLE sales MODIFY  
PARTITION BY RANGE (order_date)  
SUBPARTITION BY LIST (region)...  
UPDATE INDEXES  
(i1_ordernum GLOBAL, i2_region LOCAL,  
  i3 GLOBAL PARTITION BY RANGE (rev)  
  (PARTITION p1 VALUES LESS THAN (1000),  
   PARTITION p2 VALUES LESS THAN (15000),  
   PARTITION p3 VALUES LESS THAN (MAXVALUE) ) )  
ONLINE;
```

异步全局索引维护

DROP和TRUNCATE PARTITION之后全局索引依然可用，而不需要索引维护操作

- 内部知道受影响的分区，并且在访问时过滤掉

DROP 和 TRUNCATE变的更快, 只有元数据操作

- 明显的加速和减少初始资源消耗

延迟全局索引维护

- 通过ALTER INDEX REBUILD|COALESCE延迟的维护
- 使用scheduled job自动清理

异步全局索引维护

11.2

```
SQL> select count(*) from pt partition for (9999);
```

COUNT(*)
25341440

```
Elapsed: 00:00:01.00
SQL> select index_name, status, orphaned_entries from user_indexes;
```

INDEX_NAME	STATUS	ORPHANED_ENTRIES
I1_PT	VALID	NO

```
Elapsed: 00:00:01.04
SQL>
SQL> alter table pt drop partition for (9999) update indexes;

Table altered.

Elapsed: 00:02:04.52
SQL>
SQL> select index_name, status, orphaned_entries from user_indexes;
```

INDEX_NAME	STATUS	ORPHANED_ENTRIES
I1_PT	VALID	NO

```
Elapsed: 00:00:00.10
```

12.1

```
SQL> select count(*) from pt partition for (9999);
```

COUNT(*)
25341440

```
Elapsed: 00:00:00.98
SQL> select index_name, status, orphaned_entries from user_indexes;
```

INDEX_NAME	STATUS	ORPHANED_ENTRIES
I1_PT	VALID	NO

```
Elapsed: 00:00:00.33
SQL>
SQL> alter table pt drop partition for (9999) update indexes;

Table altered.

Elapsed: 00:00:00.04
SQL>
SQL> select index_name, status, orphaned_entries from user_indexes;
```

INDEX_NAME	STATUS	ORPHANED_ENTRIES
I1_PT	VALID	YES

```
Elapsed: 00:00:00.05
```

ORACLE

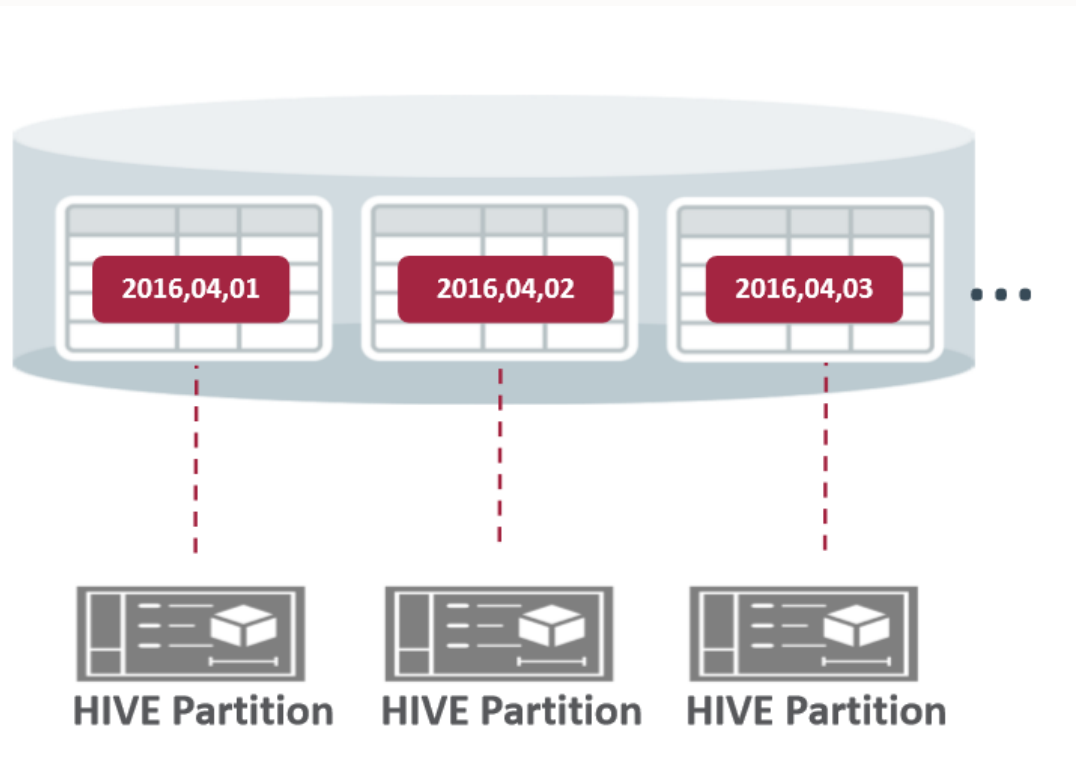
Oracle混合分区表

—



外部分区表

NEW IN
12.2



映射 Hive & HDFS 类型的分区表到Oracle的生态系统

展现Oracle分区对外部基于HDFS数据的强大功能

- 分区裁剪和维护
- 大数据感知优化

实现更快的查询和增强的数据维护功能



混合分区表 - Hybrid Partitioned Tables(HyPT)

融合数据库内部分区与外部分区于一体

可以轻松地将内部和外部的数据集成到单个分区表中，同时也可以将非活跃数据移动到外部文件，降低存储成本的同时也更加方便数据交换



COLD DATA

HOT DATA

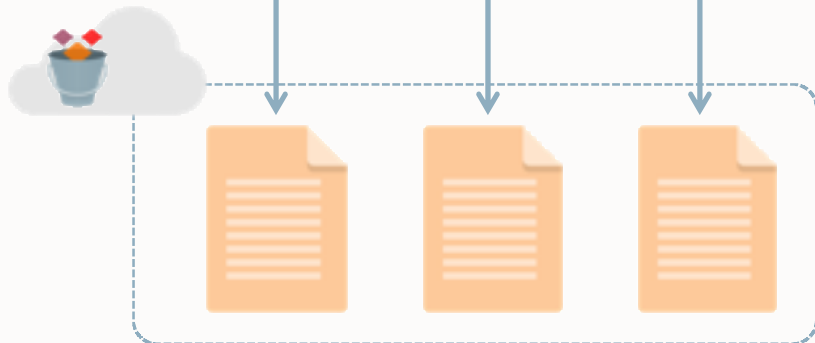
ORDERS

Q4_2016 Q1_2017 Q2_2017 Q3_2017 Q4_2017 Q1_2018 Q2_2018 Q3_2018 Q4_2018



UPDATES

TABLE PARTITIONED BY QUARTER

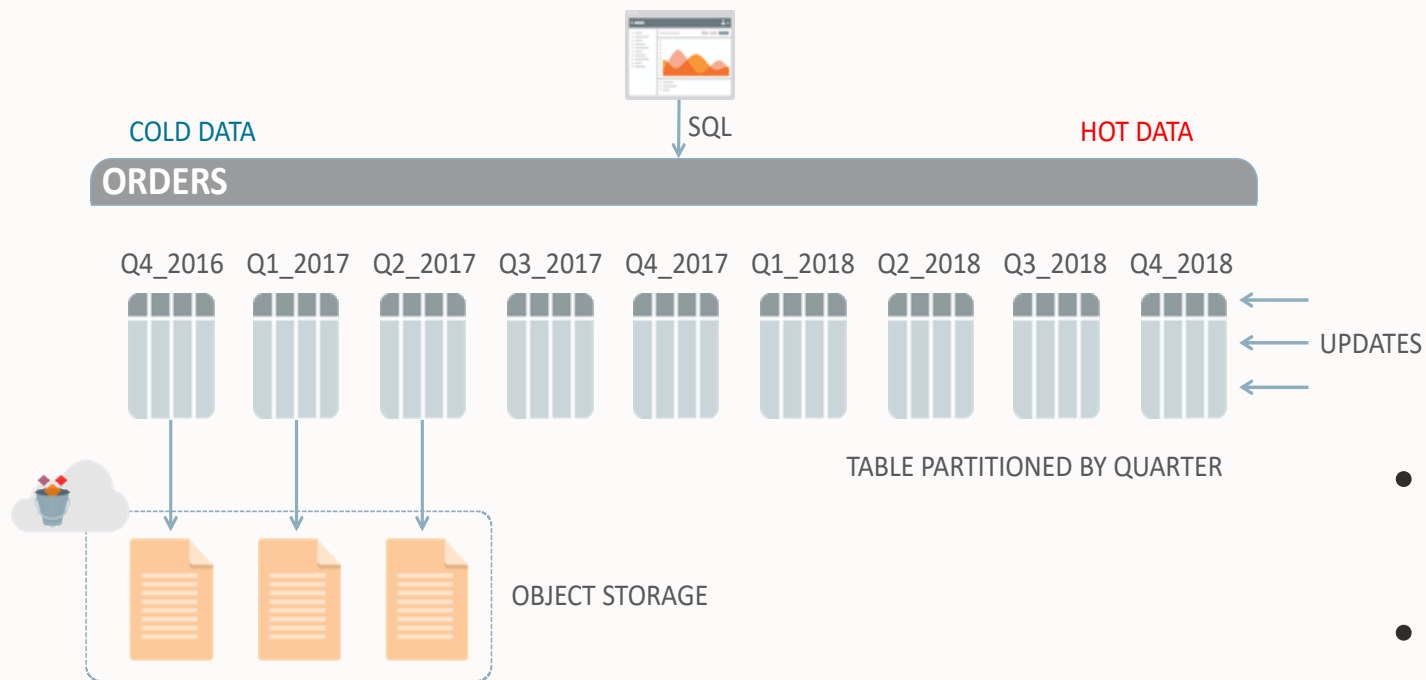


OBJECT STORAGE

```
CREATE TABLE orders ( order_idnumber,  
                      order_dateDATE, ... )  
EXTERNAL PARTITION ATTRIBUTES  
( TYPE oracle_loaderDEFAULTDIRECTORY data_dir  
  ACCESS PARAMETERS (..) REJECT LIMIT unlimited)  
PARTITION BY RANGE(order_date)  
( partition q1_2015 values less than ('2014-10-01')  
  EXTERNAL LOCATION ('order_q1_2015.csv'),  
  partition q2_2015 values less than ('2015-01-01'),  
  partition q3_2015 values less than ('2015-04-01'),  
  partition q4_2015 values less than ('2015-07-01'));
```


混合分区表 - Hybrid Partitioned Tables(HyPT)

NEW IN
19c



- 混合分区表支持外部表分区的所有驱动类型：
 - ORACLE_DATAPUMP
 - ORACLE_LOADER
 - ORACLE_HDFS
 - ORACLE_HIVE
- 所有外部表参数均适用于混合分区表的外部分区。
- 混合分区表可以跨内部、外部分区使用基于分区的优化技术，典型的比如：静态分区修剪、动态分区修剪、布隆修剪
- 通过DBMS_CLOUD可以直接加载和访问Object Storage

混合分区表-初始创建

访问文件系统



```
CREATE TABLE orders ( order_id number, order_date DATE, ... )  
EXTERNAL PARTITION ATTRIBUTES  
( TYPE oracle_loader DEFAULT DIRECTORY data_dir ACCESS PARAMETERS (..) REJECT LIMIT unlimited )  
PARTITION BY RANGE(order_date)  
( partition q1_2015 values less than ('2014-10-01') EXTERNAL LOCATION ('order_q1_2015.csv'),  
  partition q2_2015 values less than ('2015-01-01'),  
  partition q3_2015 values less than ('2015-04-01'),  
  partition q4_2015 values less than ('2015-07-01')  
);
```



混合分区表-初始创建

访问对象存储

```
CREATE TABLE orders ( order_id number, order_date DATE, ... )
ORGANIZATION EXTERNAL
( TYPE oracle_loader DEFAULT DIRECTORY data_dir
  ACCESS PARAMETERS (
    CREDENTIAL 'DEFAULT_ACCESS' )
  ) REJECT LIMIT unlimited
PARTITION BY RANGE(order_date)
( partition q1_2015 values less than ('2014-10-01')
  LOCATION ( 'https://swiftobjectstorage.us-ashburn-1 ...' ),
  partition q2_2015 values less than ('2015-01-01') LOCATION ('...'),
  partition q3_2015 values less than ('2015-04-01') LOCATION ('...'),
  partition q4_2015 values less than ('2015-07-01')
  );
```

混合分区表

- 扩展Oracle分区，允许分区存储在Oracle数据库段和外部文件中
- 外部分区可以存储在标准文件系统、HDFS或对象存储上
 - 将Hive & HDFS 分区表映射到Oracle生态系统中
 - 展示了Oracle分区对于基于外部HDFS数据存储的强大功能(分区修剪和维护/大数据感知优化)
- 混合分区带来的收益
 - 该模式支持在低成本存储上对数据进行长期归档，同时允许对其继续在线访问
 - 支持长期访问不经常访问的数据，以满足行业法规需求
- 表级别的外部参数应用于混合分区表的所有分区
- 混合分区表可以跨内部和外部分区使用基于分区的优化，比如Static partition pruning/Dynamic partition pruning/Bloom pruning

管理混合分区表



- 创建混合分区表
- 添加新的外部分区
- 将只有内部分区的表转换为混合分区表
- 将混合分区表转换为内部分区表
- 在混合分区表上启用ADO
 - 注意只有内部分区生效
- 在混合分区表中分裂分区
- 在混合分区表中交换数据
 - 外部分区与外部非分区表交换
 - 内部分区与内部非分区表交换
 - 内部分区与外部非分区表交换（有限支持，可能需要临时中间表）



管理混合分区表



添加新的外部分区

```
ALTER TABLE hybrid_sales  
  ADD PARTITION sales_2020 VALUES LESS THAN (TO_DATE('01-01-2021','dd-mm-yyyy'))  
  EXTERNAL DEFAULT DIRECTORY sales_data_acfs LOCATION ('sales2020_data.txt');
```

管理混合分区表

将只有内部分区的表转换为混合分区表

将内部范围分区表转换为混合分区表时，首先需要向现有表添加外部分区属性，然后添加外部分区，并且至少有一个分区必须是内部分区。

```
CREATE TABLE internal_to_hypt_table (  
  prod_id    NUMBER    NOT NULL,  
  cust_id    NUMBER    NOT NULL,  
  time_id    DATE      NOT NULL,  
  channel_id NUMBER    NOT NULL,  
  promo_id   NUMBER    NOT NULL,  
  quantity_sold NUMBER(10,2) NOT NULL,  
  amount_sold NUMBER(10,2) NOT NULL  
)  
PARTITION by range (time_id)  
(PARTITION sales_2014 VALUES LESS THAN (TO_DATE('01-01-2015','dd-mm-  
yyyy'))  
);
```

```
ALTER TABLE internal_to_hypt_table  
  ADD EXTERNAL PARTITION ATTRIBUTES  
(TYPE ORACLE_LOADER  
  DEFAULT DIRECTORY sales_data  
  ACCESS PARAMETERS (  
    FIELDS TERMINATED BY ','  
    (prod_id,cust_id,time_id DATE 'dd-mm-  
yyyy',channel_id,promo_id,quantity_sold,amount_sold)  
  ));  
ALTER TABLE internal_to_hypt_table  
  ADD PARTITION sales_2015 VALUES LESS THAN (TO_DATE('01-01-  
2016','dd-mm-yyyy'))  
  EXTERNAL LOCATION ('sales2015_data.txt');  
  
ALTER TABLE internal_to_hypt_table  
  ADD PARTITION sales_2016 VALUES LESS THAN (TO_DATE('01-01-  
2017','dd-mm-yyyy'))  
  EXTERNAL LOCATION ('sales2016_data.txt');  
  
SELECT HYBRID FROM USER_TABLES WHERE TABLE_NAME =  
'INTERNAL_TO_HYPT_TABLE';
```

HYB

YES

免费咨询热线：400-699-8888



混合分区表限制

- 应用于外部表的限制也适用于混合分区表，除非明确说明
- 在表级别上定义的In-Memory只对混合分区表的内部分区产生影响
- 在混合分区表级别定义的Automatic Data Optimization (ADO)策略只影响内部分区
- 只能对混合分区表的内部分区进行DML操作(外部分区被视为只读分区)
- 在外部分区上不允许SPLIT, MERGE以及MOVE 维护操作
- 不支持Reference & System 分区类型
- 只支持Single level list & Range partitioning
- 对HIVE只支持Single level list partitioning
- 没有唯一索引或者全局唯一索引。只支持部分索引，唯一索引不能是部分索引。



混合分区表限制



- 没有列默认值
- 只允许RELY 约束
- 不支持LOB， LONG和ADT类型
- 不支持cluster表属性(CLUSTERING clause)
- 不支持虚拟列
- 不支持CELLMEMORY 子句
- 不支持常见约束， 只允许RELY constraints



Q&A





基于 Oracle 数据库 免费企业数据健康检查

- 及时了解数据库健康状况，发现并解决潜在问题
- 维护数据库系统良好状态，保护数据资产的安全
- 提升数据库性能、稳定性和安全性，降低业务风险

免费咨询热线：

400-699-8888

* 活动最终解释权归甲骨文公司所有

Oracle Sharding 可扩展的多模型分布式 数据库



范宏伟

- 资深解决方案工程师
- 电信行业架构师
- 15年以上数据相关工作经验

内容简介

Oracle RAC和Data Guard可以满足数据库应用的绝大部分需求，但是仍然有一些用户的特别应用，需要更高的可扩展性和故障隔离性，因此Oracle从12.2开始支持Sharding部署模式。主要介绍sharding部署模式的主要组件、生命周期管理和技术演进。



Zoom直播

直播时间：7月21日 11:00 - 12:00

扫描二维码进入直播

Zoom ID: 957 9669 6723

密码：20212023



微信扫一扫预约



数据库和云讲座群

20-21



甲骨文云技术公众号



技术专家1V1深入交流



ORACLE