



Database Compatibility Table Partitioning Guide

リリース *14.1.0*

EnterpriseDB

2021 年 12 月 15 日

目次

第 1 章	Selecting a Partition Type	3
1.1	Interval Range Partitioning	4
1.2	Automatic List Partitioning	5
第 2 章	Using Partition Pruning	7
2.1	Example - Partition Pruning	10
第 3 章	Partitioning Commands Compatible with Oracle Databases	13
3.1	CREATE TABLE...PARTITION BY	13
3.1.1	Example - PARTITION BY LIST	18
3.1.2	Example - AUTOMATIC LIST PARTITION	19
3.1.3	Example - PARTITION BY RANGE	20
3.1.4	Example - INTERVAL RANGE PARTITION	21
3.1.5	Example - PARTITION BY HASH	22
3.1.6	Example - PARTITION BY HASH...PARTITIONS num...	23
3.1.7	Example - PARTITION BY RANGE, SUBPARTITION BY LIST	33
3.1.8	Example - CREATING UNIQUE INDEX on PARTITION TABLE	35
3.1.9	Example - CREATING SUBPARTITION TEMPLATE	36
3.2	ALTER TABLE...ADD PARTITION	40
3.2.1	Example - Adding a Partition to a LIST Partitioned Table	42
3.2.2	Example - Adding a Partition to a RANGE Partitioned Table	43
3.2.3	Example - Adding a Partition with SUBPARTITIONS num...IN PARTITION DESCRIPTION	45
3.3	ALTER TABLE...ADD SUBPARTITION	48
3.3.1	Example - Adding a Subpartition to a LIST/RANGE Partitioned Table	50
3.3.2	Example - Adding a Subpartition to a RANGE/LIST Partitioned Table	52
3.4	ALTER TABLE...SPLIT PARTITION	53
3.4.1	Example - Splitting a LIST Partition	55
3.4.2	Example - Splitting a RANGE Partition	58
3.4.3	Example - Splitting a RANGE/LIST Partition	60
3.5	ALTER TABLE...SPLIT SUBPARTITION	65
3.5.1	Example - Splitting a LIST Subpartition	67
3.5.2	Example - Splitting a RANGE Subpartition	69
3.6	ALTER TABLE...EXCHANGE PARTITION	73
3.6.1	Example - Exchanging a Table for a Partition	74
3.7	ALTER TABLE...MOVE PARTITION	77
3.7.1	Example - Moving a Partition to a Different Tablespace	78
3.8	ALTER TABLE...RENAME PARTITION	79
3.8.1	Example - Renaming a Partition	80
3.9	ALTER TABLE...SET INTERVAL	81

3.9.1	Example - Setting an Interval Range Partition	81
3.10	ALTER TABLE...SET [PARTITIONING] AUTOMATIC	83
3.10.1	Example - Setting an AUTOMATIC List Partition	83
3.11	ALTER TABLE...SET SUBPARTITION TEMPLATE	84
3.11.1	Example - Setting a SUBPARTITION TEMPLATE	85
3.12	DROP TABLE	95
3.13	ALTER TABLE...DROP PARTITION	95
3.13.1	Example - Deleting a Partition	96
3.14	ALTER TABLE...DROP SUBPARTITION	97
3.14.1	Example - Deleting a Subpartition	98
3.15	TRUNCATE TABLE	99
3.15.1	Example - Emptying a Table	99
3.16	ALTER TABLE...TRUNCATE PARTITION	101
3.16.1	Example - Emptying a Partition	102
3.17	ALTER TABLE...TRUNCATE SUBPARTITION	104
3.17.1	Example - Emptying a Subpartition	105
3.18	Accessing a PARTITION or SUBPARTITION	107
3.18.1	PARTITION または SUBPARTITION へのアクセスにエイリアスを使用する	110
第 4 章	Handling Stray Values in a LIST or RANGE Partitioned Table	113
第 5 章	Specifying Multiple Partitioning Keys in a RANGE Partitioned Table	121
第 6 章	Retrieving Information about a Partitioned Table	123
6.1	Table Partitioning Views - Reference	124
6.1.1	ALL_PART_TABLES	124
6.1.2	ALL_TAB_PARTITIONS	125
6.1.3	ALL_TAB_SUBPARTITIONS	126
6.1.4	ALL_PART_KEY_COLUMNS	127
6.1.5	ALL_SUBPART_KEY_COLUMNS	127

パーティション化されたテーブルでは、論理的にラージ 1 つのテーブルが小さな物理的断片に分割されます。パーティション化にはいくつかの利点があります。

- クエリのパフォーマンスは、特定の状況、特にテーブルの頻繁にアクセスされる行のほとんどが単一のパーティションまたは少数のパーティションにある場合に劇的に改善できます。パーティション化により、インデックスの前からパーティション列を省略できるため、インデックスサイズが小さくなり、インデックスの頻繁に使用される部分がメモリに収まる可能性が高くなります。
- クエリーまたは更新が単一のパーティションのラージ部分にアクセスする場合、インデックスおよびテーブル全体に散在するランダムアクセス読み取りを使用する代わりに、サーバーがパーティションのシーケンシャルスキャンを実行するため、パフォーマンスが向上する場合があります。
- パーティショニングのデザインにその要件を計画している場合、パーティションを追加または削除することにより、バルクロード（またはアンロード）を実装できます。ALTER TABLE は、一括オペレーションよりもはるかに高速です。また、バルク DELETE に起因する VACUUM のオーバーヘッドを完全に回避します。
- あまり使用されないデータは、安価な（または低速の）ストレージメディアに移行できます。

テーブルのパーティショニングは、テーブルが非常にラージ場合にのみ価値があります。テーブルがパーティショニングの恩恵を受ける正確なポイントは、アプリケーションによって異なります。ルールとして、テーブルのサイズはデータベースサーバーの物理的メモリを超える必要があります。

このドキュメントでは、Advanced Server でサポートされているオラクルデータベースと互換性のあるテーブルパーティショニングの側面について説明します。

PostgreSQL このドキュメント、特にこの章で説明するパーティショニングパーティショニングは、PostgreSQL バージョン 10 で導入された宣言 *Advanced Server* パーティショニング機能については説明していません。この章で説明するデータベース。宣言的パーティショニングの詳細については、次の URL で入手可能な PostgreSQL コア文書を参照してください。< <https://www.postgresql.org/docs/current/ 静的/ddl-partitioning.html> >

PostgreSQL 9.6 INSERT... ON CONFLICT DO NOTHING/UPDATE 句（一般に UPSERT として知られています）は、Oracle スタイルのテーブルパーティションではサポートされていません。INSERT コマンドを呼び出してパーティションテーブルにデータを追加するときに ON CONFLICT DO NOTHING/UPDATE 句を含めると、サーバーはエラーを結果します。

!!! Note * Advanced Server はグローバルインデックスをサポートしていないため、パーティションキー列を含まないプライマリキーがパーティションテーブルに定義されている場合、インデックスは継承されません。ただし、CREATE TABLE 内で定義されたすべてのパーティションには、列に独立したプライマリインデックスがあります。ALTER TABLE ... ADD CONSTRAINT を使用して、新しく追加されたすべてのパーティションでプライマリキーを再作成できます。このプライマリインデックスは、パーティション階層全体ではなく各パーティション内でのみ一意性を適用します（基本的に、2 つ以上のパーティションのプライマリインデックス列に対して同じ値を繰り返すことができます）。

第1章 Selecting a Partition Type

</div>

パーティションテーブルを作成するときは、LIST、RANGE または HASH パーティショニングルールを指定します。パーティショニングルールは、各パーティションに格納されるデータを定義する一連のコンストレインを提供します。パーティションテーブルに新しい行が追加されると、サーバーはパーティショニングルールを使用して、各パーティションにどのパーティションを含めるかを決定します。

Advanced Server は、パーティショニングルールを使用してパーティションプルーニングを実施し、ユーザーのクエリに応答するときのパフォーマンスを向上させることもできます。テーブルのパーティショニングタイプとパーティショニングキーを選択するときは、テーブル内に格納されているデータのクエリ方法を考慮し、パーティショニングルールに頻繁にクエリされる列を含める必要があります。

リストのパーティション分割

リストパーティションテーブルを作成するときは、単一のパーティショニングキー列を指定します。テーブルに行を追加するとき、サーバーはパーティショニングルールで指定されたキー値を行の対応する列と比較します。列の値がパーティショニングルールの値と一致する場合、行はルールで名前付けされたパーティションに格納されます。

!!! Note * List Partitioning は、複数列リストのパーティショニングをサポートしていません。

範囲分割

範囲パーティションテーブルを作成する場合、1つ以上のパーティショニングキー列を指定します。テーブルに新しい行を追加すると、サーバーはパーティショニングキーの値をテーブルエントリーの対応する列と比較します。列の値がパーティショニングルールで指定された条件を満たす場合、行はルールで名前付けされたパーティションに格納されルール。

ハッシュ分割

ハッシュパーティションテーブルを作成する場合、1つ以上のパーティショニングキー列を指定します。データは、指定されたパーティション間で（ほぼ）同じサイズのパーティションに分割されます。ハッシュパーティションテーブルに行を追加すると、サーバーは指定された列のデータのハッシュ値を計算し、ハッシュ値に従ってパーティションに行を格納します。

!!! Note * Advanced Server をアップグレード処理ときは、アップグレードされたバージョンのサーバーで各ハッシュパーティションテーブルを再構築する必要があります。

サブパーティション分割

サブパーティション化は、パーティション化されたテーブルをより小さなサブセットに分割します。すべてのサブセットは、同じデータベースサーバクラスターに格納する必要があります。テーブルは通常、異なる列セットによってサブパーティション化され、親パーティションとは異なるサブパーティション化タ

イブになる場合があります。1つのパーティションがサブパーティション化されている場合、各パーティションには少なくとも1つのサブパーティションがあります。

テーブルがサブパーティション化されている場合、どのパーティションテーブルにもデータは保存されません。データは代わりに対応するサブパーティションに保存されます。

1.1 Interval Range Partitioning

</ div>

間隔範囲パーティション化は、レンジパーティショニングの拡張機能であり、新しく挿入されたデータが既存のパーティショニングのレンジを超えると、データベースが新しいパーティションを自動的に作成できるようにします。インターバルレンジパーティショニングを実装するには、**INTERVAL** 句を含めて、新しいパーティションのレンジサイズを指定します。

レンジパーティション（遷移ポイントとも呼ばれる）の高い値は、レンジパーティショニングキー値によって決まります。データベースは、挿入されたデータに対して、その高い値を超える値を持つ新しいパーティションを作成します。

インターバルが1か月に設定され、現在の移行ポイントの2か月後にデータが挿入される場合、その月のパーティションのみが作成され、介在パーティションは作成されません。例、月間インターバルと2019年2月15日の現在の移行ポイントを使用してインターバルレンジパーティションテーブルを作成できます。2019年2月15日のパーティションにデータが挿入されます。2019年2月15日から2019年3月15日、および2019年3月15日から2019年4月15日のパーティションはスキップされたされます。

間隔範囲パーティション化構文の詳細については、**CREATE TABLE...PARTITION BY** を参照してください。

間隔範囲の分割に関する制限

INTERVAL 句には、次の制限が適用されます。

- インターバルレンジのパーティショニングは、単一のパーティションキーに制限されています。そのキーは数値または日付レンジでなければなりません。
- 少なくとも1つのレンジパーティションを定義する必要があります。
- **INTERVAL** 句は、索引構成表ではサポートされていません。
- インターバルレンジパーティションテーブルにドメインインデックスを作成できません。
- 複合パーティショニングでは、インターバルレンジパーティショニングはプライマリパーティショニングメカニズムとしては有用ですが、サブパーティションレベルではサポートされていません。
- **DEFAULT** および **MAXVALUE** は、インターバルレンジパーティションテーブルに定義できません。
- **NULL**, **Not-a-Number**, および **Infinity** の値は、パーティショニングキー列に指定できません。
- 間隔レンジパーティショニング式は定数数値を生成する必要があり、負の値であってはなりません。
- インターバルレンジパーティションテーブルのパーティションは、オーダーでのみ作成されます。

1.2 Automatic List Partitioning

</div>

自動リストパーティショニングは、LIST パーティショニングの拡張機能であり、データベースがリストパーティショニングキーの新しい個別の値に対してパーティションを自動的に作成できるようにします。データが LIST パーティションテーブルに挿入され、挿入された値が既存のテーブルパーティションのいずれともマッチしない場合、新しいパーティションが作成されます。AUTOMATIC 句を使用して、自動リストパーティショニングを実装します。

例、既存のパーティション値 CALIFORNIA および FLORIDA を含む sales_state 列を持つ sales 名前付けのテーブルを考えます。sales_state の各値は、州ごとの売上高の増加とともに増加します。新しい状態（例や OHIO など）での販売には、新しいパーティションの作成が必要になります。自動リストパーティショニングを実装している場合、新しいパーティション INDIANA および OHIO が自動的に作成され、データが sales テーブルに入力されます。

自動リストパーティショニング構文の詳細については、[CREATE TABLE…PARTITION BY](#) を参照してください。

自動リストパーティション分割の制限

AUTOMATIC 句には、次の制限が適用されます。

- 自動リストパーティショニングを有効にするテーブルに DEFAULT パーティションを含めることはできません。
- 自動リストパーティショニングは、複数列のリストパーティショニングをサポートしていません。
- 複合パーティショニングでは、自動リストパーティショニングはプライマリパーティショニングメカニズムとして有用ですが、サブパーティションレベルではサポートされていません。

第2章 Using Partition Pruning

</ div>

Advanced Server のクエリープランナは、パーティションプルーニングを使用して、SELECT ステートメントの WHERE 句で指定された条件に一致する行を見つけるための効率的な計画を計算します。実行計画からパーティションを正常にプルーニングするには、WHERE 句は、パーティション表の作成時に指定されたパーティショニングキー列と比較される情報を制約する必要があります。クエリする場合：

- パーティション分割リスト、WHERE 句が equal (=) や AND. などの演算子を使用してリテラル値をパーティショニングキーと比較する場合、パーティションプルーニングが有効です。
- 範囲パーティションテーブル、WHERE 句が等号 (=)、より小さい (<)、より大きい (>) などの演算子を使用してリテラル値をパーティショニングキーと比較する場合、パーティションプルーニングが有効です。
- ハッシュパーティションテーブル。パーティションのプルーニングは、WHERE 句が等号 (=) などの演算子を使用してリテラル値をパーティショニングキーと比較するときに有効です。

パーティションプルーニングメカニズムは、2つの最適化手法を使用します。

- 高速プルーニング
- 制約の除外

パーティションプルーニング技術は、データの検索を、検索対象の値が存在する可能性のあるパーティションのみに制限します。両方のプルーニング手法は、クエリの実行計画からパーティションを削除し、パフォーマンスを向上させます。

高速プルーニングと制約による除外の違いは、高速プルーニングは Oracle パーティション表のパーティション間のリレーションを理解するのに対し、制約による除外除外は理解しないことです。例、クエリーがリストパーティションテーブル内の特定の値を検索する場合、高速プルーニングは特定のパーティションのみがその値を保持できると推論できますが、制約による除外は* each *パーティションに定義されたコンストレインを調べる必要があります。高速プルーニングは計画プロセスの早い段階で行われ、プランナが考慮する必要のあるパーティションの数を減らしますが、制約による除外は計画プロセスの遅い段階で行われます。

制約の除外の使用

constraint_exclusion パラメータは、制約による除外を制御します。constraint_exclusion パラメータの値は、on, off, または partition になります。制約による除外を有効にするには、パラメータを* partition または on に設定する必要があります。デフォルトでは、パラメータは partition. に設定されます

制約による除外の詳細については、以下を参照してください。

< <https://www.postgresql.org/docs/current/ 静的/ddl-partitioning.html> >

制約による除外が有効になっている場合、サーバーは各パーティションに定義されたコンストレインを調べて、そのパーティションがクエリーを満たすことができるかどうかを判断します。

WHERE 句を**含まない** SELECT ステートメントを実行する場合、クエリープランナはテーブル全体を検索する実行計画を推奨する必要があります。WHERE 句を**含む** SELECT ステートメントを実行すると、クエリープランナはどのパーティションにその行を格納するかを決定し、クエリーフラグメントをそのパーティションに送信し、実行計画からその行を含むことができなかったパーティションをブルーニングします。テーブルパーティションを使用していない場合、制約による除外を無効にするとパフォーマンスが向上する場合があります。

高速ブルーニング

制約による除外と同様に、高速ブルーニングは WHERE (または結合) 句を含むクエリのみを最適化し、WHERE 句の修飾子が特定のフォームにマッチする場合にのみ最適化できます。どちらの場合でも、クエリープランナは、クエリーに必要なデータを保持できない可能性があるパーティション内のデータの検索を回避します。

高速ブルーニングは、`edb_enable_pruning`. 名前付けのブール値設定パラメータによって制御されます。`edb_enable_pruning` が ON の場合、Advanced Server は特定のクエリを高速でブルーニングします。`edb_enable_pruning` が OFF の場合、サーバーは高速ブルーニングを無効にします。

!!! Note * 高速ブルーニングでは、サブパーティションテーブルに対するクエリを最適化することも、複数の列でパーティション化された範囲テーブルパーティションに対するクエリを最適化することもできません。

LIST テーブルパーティションの場合、Advanced Server は、パーティショニング列をリテラル値に制約する WHERE 句を含むクエリを高速にブルーニングできます。例、次のような LIST パーティションテーブルがあるとします。

```
CREATE TABLE sales_hist(..., country text, ...)
PARTITION BY LIST(country)
(
  * PARTITION americas VALUES('US', 'CA', 'MX'),
  * PARTITION europe VALUES('BE', 'NL', 'FR'),
  * PARTITION asia VALUES('JP', 'PK', 'CN'),
  * PARTITION others VALUES(DEFAULT)
)
```

高速ブルーニングは、次のような WHERE 句について推論できます。

```
WHERE country = 'US'
```

```
WHERE country IS NULL;
```

最初の WHERE 句が与えられた場合、高速ブルーニングはパーティション `europe`, `asia`, および `others` を削除します。これらのパーティションは修飾子 `WHERE country = 'US'`. を満たす行を保持できないためです。

2 番目の WHERE 句が与えられた場合、高速ブルーニングは `americas`, `europe` および `asia` パーティションを削除します。これらのパーティションは `country IS NULL`. の行を保持できないためです。

WHERE 句で指定される演算子は、パーティショニング列のデータタイプに適した等号 (=) または等価演算

子でなければなりません。

レンジテーブルパーティションの場合、Advanced Server は、パーティショニングカラムをリテラル値に制限する WHERE 句を含むクエリを高速にブルーニングできますが、演算子は次のいずれかになります。

>

>=

=

<=

<

高速ブルーニングは、次のような AND および BETWEEN 演算子を含むより複雑な式についても推論します。

```
WHERE size > 100 AND size <= 200
WHERE size BETWEEN 100 AND 200
```

高速枝刈りは、OR または IN. を含む式に基づいて枝刈りできません。例、次のような RANGE パーティションテーブルを照会する場合

```
CREATE TABLE boxes(id int, size int, color text)
  * PARTITION BY RANGE(size)
(
  * PARTITION small VALUES LESS THAN(100),
  * PARTITION medium VALUES LESS THAN(200),
  * PARTITION large VALUES LESS THAN(300)
)
```

高速ブルーニングは、次のような WHERE 句について推論できます。

```
WHERE size > 100 -- scan partitions 'medium' and 'large'
```

```
WHERE size >= 100 -- scan partitions 'medium' and 'large'
```

```
WHERE size = 100 -- scan partition 'medium'
```

```
WHERE size <= 100 -- scan partitions 'small' and 'medium'
```

```
WHERE size < 100 -- scan partition 'small'
```

```
WHERE size > 100 AND size < 199 -- scan partition 'medium'
```

```
WHERE size BETWEEN 100 AND 199 -- scan partition 'medium'
```

```
WHERE color = 'red' AND size = 100 -- scan 'medium'
```

```
WHERE color = 'red' AND (size > 100 AND size < 199) -- scan 'medium'
```

いずれの場合も、高速ブルーニングでは、修飾子がパーティショニング列とリテラル値（または IS NULL/IS NOT NULL）。

!!! Note * 高速ブルーニングは、上記の形式の WHERE 句を含む DELETE および UPDATE ステートメントも最適化できます。

2.1 Example - Partition Pruning

</ div>

EXPLAIN ステートメントは、ステートメントの実行計画を表示します。EXPLAIN ステートメントを使用して、Advanced Server がクエリーの実行計画からパーティションをプルーニングしていることを確認できます。

パーティションプルーニングの効率を実証するには、最初にシンプルテーブルを作成します。

```
CREATE TABLE sales
(
  * dept_no      number,
  * part_no      varchar2,
  * country      varchar2(20),
  * date         date,
  * amount       number
)
PARTITION BY LIST(country)
(
  * PARTITION europe VALUES('FRANCE', 'ITALY'),
  * PARTITION asia VALUES('INDIA', 'PAKISTAN'),
  * PARTITION americas VALUES('US', 'CANADA')
);
```

次に、EXPLAIN ステートメントを含む制約付きクエリーを実行します。

```
EXPLAIN (COSTS OFF) SELECT * FROM sales WHERE country = 'INDIA';
```

結果のクエリプランは、サーバーが sales_asia テーブル (country 値が INDIA の行が格納されるテーブル) のみをスキャンすることを示しています。

```
edb=# EXPLAIN (COSTS OFF) SELECT * FROM sales WHERE country = 'INDIA';
*
      QUERY PLAN
-----
Append
*  ->  Seq Scan on sales_asia
*      Filter: ((country)::text = 'INDIA'::text)
(3 rows)
```

パーティショニングキーに含まれていない値に一致する行を検索するクエリーを実行する場合：

```
EXPLAIN (COSTS OFF) SELECT * FROM sales WHERE dept_no = '30';
```

結果のクエリプランは、クエリーを満たす行を見つけるために、サーバーがすべてのパーティションを調べる必要があることを示しています。

```
edb=# EXPLAIN (COSTS OFF) SELECT * FROM sales WHERE dept_no = '30';
*
          QUERY PLAN
-----
Append
*  -> Seq Scan on sales_americas
*      Filter: (dept_no = '30'::numeric)
*  -> Seq Scan on sales_europe
*      Filter: (dept_no = '30'::numeric)
*  -> Seq Scan on sales_asia
*      Filter: (dept_no = '30'::numeric)
(7 rows)
```

制約の除外は、サブパーティションテーブルをクエリする場合にも適用されます。

```
CREATE TABLE sales
(
    * dept_no      number,
    * part_no      varchar2,
    * country      varchar2(20),
    * date         date,
    * amount       number
)
PARTITION BY RANGE(date) SUBPARTITION BY LIST (country)
(
    * PARTITION "2011" VALUES LESS THAN('01-JAN-2012')
    * (
    *     SUBPARTITION europe_2011 VALUES ('ITALY', 'FRANCE'),
    *     SUBPARTITION asia_2011 VALUES ('PAKISTAN', 'INDIA'),
    *     SUBPARTITION americas_2011 VALUES ('US', 'CANADA')
    * ),
    * PARTITION "2012" VALUES LESS THAN('01-JAN-2013')
    * (
    *     SUBPARTITION europe_2012 VALUES ('ITALY', 'FRANCE'),
    *     SUBPARTITION asia_2012 VALUES ('PAKISTAN', 'INDIA'),
    *     SUBPARTITION americas_2012 VALUES ('US', 'CANADA')
    * ),
    * PARTITION "2013" VALUES LESS THAN('01-JAN-2015')
    * (
    *     SUBPARTITION europe_2013 VALUES ('ITALY', 'FRANCE'),
    *     SUBPARTITION asia_2013 VALUES ('PAKISTAN', 'INDIA'),
    *     SUBPARTITION americas_2013 VALUES ('US', 'CANADA')
    * )
);
```

テーブルをクエリーすると、クエリープランナは、検索パスから必要な結果セットを含めることができないパーティションまたはサブパーティションを削除します。

```
edb=# EXPLAIN (COSTS OFF) SELECT * FROM sales WHERE country = 'US' AND date = 'Dec 12, 2012';
```

```
  *
```

```
QUERY PLAN
```

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

```
Append
```

```
  * -> Seq Scan on sales_americas_2012
```

```
  *      Filter: (((country)::text = 'US'::text) AND (date = '12-DEC-12
```

```
  *      00:00:00'::timestamp without time zone))
```

```
(3 rows)
```

第3章 Partitioning Commands Compatible with Oracle Databases

</ div>

次のセクションでは、Advanced Server がサポートするオラクルデータベースと互換性のあるテーブルパーティショニング構文の使用に関する情報を提供します。

3.1 CREATE TABLE...PARTITION BY

</ div>

CREATE TABLE コマンドの PARTITION BY 句を使用して、1 つ以上のパーティション（およびサブパーティション）にデータが分散されたパーティションテーブルを作成します。コマンド構文は次の形式で提供されます。

リスト分割構文

最初のフォームを使用して、リストパーティションテーブルを作成します。

```
CREATE TABLE [ schema. ]<table_name>
* <table_definition>
* PARTITION BY LIST(<column>) [ AUTOMATIC ]
* [SUBPARTITION BY {RANGE|LIST|HASH} (<column>[, <column> ]...)]
* [SUBPARTITION TEMPLATE (<subpartition_template_description>)]
* (<list_partition_definition>[, <list_partition_definition>]...);
```

Where list_partition_definition is:

```
* PARTITION [<partition_name>]
* VALUES (<value>[, <value>]...)
* [TABLESPACE <tablespace_name>]
* [(<subpartition>, ...)]
```

Where subpartition_template_description is:

```
* SUBPARTITION [<subpartition_name>]
* VALUES (<value>[, <value>]...)
* [TABLESPACE <tablespace_name>], ...
```

範囲分割構文

2 番目のフォームを使用して、範囲パーティションテーブルを作成します。

```
CREATE TABLE [ schema. ]<table_name>
* <table_definition>
* PARTITION BY RANGE(<column>[, <column> ]...)
* [INTERVAL (<constant> | <expression>)]
* [SUBPARTITION BY {RANGE|LIST|HASH} (<column>[, <column> ]...)
* [SUBPARTITION TEMPLATE (<subpartition_template_description>)]
* (<range_partition_definition>[, <range_partition_definition>]...);
```

Where range_partition_definition is:

```
* PARTITION [<partition_name>]
* VALUES LESS THAN (<value>[, <value>]...)
* [TABLESPACE <tablespace_name>]
* [(<subpartition>, ...)]
```

Where subpartition_template_description is:

```
* SUBPARTITION [<subpartition_name>]
* VALUES LESS THAN (<value>[, <value>]...)
* [TABLESPACE <tablespace_name>], ...
```

ハッシュ分割構文

3 番目のフォームを使用して、ハッシュパーティションテーブルを作成します。

```
CREATE TABLE [ schema. ]<table_name>
* <table_definition>
* PARTITION BY HASH(<column>[, <column> ]...)
* [SUBPARTITION BY {RANGE|LIST|HASH} (<column>[, <column> ]...)
* [SUBPARTITION TEMPLATE (<subpartition_template_description>)]
* [SUBPARTITIONS <num>] [STORE IN ( <tablespace_name> [, <tablespace_name>]... ) ]
→]]
* [PARTITIONS <num>] [STORE IN ( <tablespace_name> [, <tablespace_name>]... ) ] |
* (<hash_partition_definition>[, <hash_partition_definition>]...);
```

Where hash_partition_definition is:

```
* PARTITION [<partition_name>]
* [TABLESPACE <tablespace_name>]
* [(<subpartition>, ...)]
```

Where subpartition_template_description is:

```
* SUBPARTITION [<subpartition_name>]
* [TABLESPACE <tablespace_name>], ...
```

サブパーティションの構文

subpartition は次のいずれかです。

```

{<list_subpartition> | <range_subpartition> | <hash_subpartition>}

where list_subpartition is:
    *      SUBPARTITION [<subpartition_name>]
    *      VALUES (<value>[, <value>]...)
    *      [TABLESPACE <tablespace_name>]

where range_subpartition is:
    *      SUBPARTITION [<subpartition_name>]
    *      VALUES LESS THAN (<value>[, <value>]...)
    *      [TABLESPACE <tablespace_name>]

where hash_subpartition is:
    *      SUBPARTITION [<subpartition_name>]
    *      [TABLESPACE <tablespace_name>] |
    *      SUBPARTITIONS <num> [STORE IN ( <tablespace_name> [, <tablespace_name>]... )]
→ ]

```

説明

CREATE TABLE... PARTITION BY コマンドは、1 つ以上のパーティションを持つテーブルを作成します。各パーティションには 1 つ以上のサブパーティションがあります。定義されたパーティションの数に上限はありませんが、PARTITION BY 句を含める場合は、少なくとも 1 つのパーティショニングルールを指定する必要があります。結果のテーブルは、それを作成したユーザが所有します。

PARTITION BY LIST 句を使用して、指定した列に入力された値に基づいてテーブルをパーティションに分割します。各パーティショニングルールでは、少なくとも 1 つのリテラル値を指定する必要がありますが、指定できる値の数に上限はありません。DEFAULT のマッチング値を指定するルールを含めて、非修飾行を特定のパーティションに向けます。DEFAULT キーワードの使用の詳細については、[Handling Stray Values in a LIST or RANGE Partitioned Table](#) を参照してください。

AUTOMATIC 句を使用して、自動リストパーティショニングを使用するテーブルを指定します。AUTOMATIC 句を指定することにより、既存のパーティションに対して宣言された値に対応しないテーブルに新しいデータが挿入されると、データベースは自動的にパーティションを作成します。

AUTOMATIC LIST PARTITIONING の詳細については、[Automatic List Partitioning](#) を参照してください。

PARTITION BY RANGE 句を使用して、パーティションを作成する境界ルールを指定します。各パーティショニングルールには、2 つの演算子（つまり、演算子以上、および演算子より小さい）を持つデータタイプの列が少なくとも 1 つ含まれている必要があります。範囲の境界は、LESS THAN 句に対して評価され、包括的ではありません。2013 年 1 月 1 日の日付境界には、2012 年 12 月 31 日以前の日付値のみが含まれます。

範囲パーティショニングルールはオーダーで指定する必要があります。パーティショニングルールに MAXVALUE の値を指定する境界ルールが含まれていない限り、値が範囲パーティションテーブルの上部境界を超える行を格納する INSERT コマンドは失敗します。MAXVALUE パーティショニングルールを含めない場合、境界ルールで指定された最大制限を超える行はエラーになります。

MAXVALUE キーワードの使用の詳細については、[Handling Stray Values in a LIST or RANGE Partitioned Table](#) を参照してください。

INTERVAL 句を使用して、インターバルレンジのパーティション表を指定します。INTERVAL 句を指定すると、データベースによってレンジパーティショニングが自動的に拡張され、既存のレンジパーティションを超えるテーブルに新しいデータが挿入されると、指定されたインターバルのパーティションが作成されます。

INTERVAL RANGE PARTITION の詳細については、[Interval Range Partitioning](#) を参照してください。

PARTITION BY HASH 句を使用して、ハッシュパーティションテーブルを作成します。HASH パーティションテーブルでは、パーティショニング構文で指定された列のハッシュ値に基づいて、データが同じサイズのパーティションに分割されます。ヘルプパーティションを指定する場合、できるだけ一意に近い列（または列の組み合わせ）を選択して、パーティション間でデータが均等に分散される保証にします。パーティショニング列（または列の組み合わせ）を選択する場合、最適なパフォーマンスを得るために完全一致を頻繁に検索する列（または列）を選択します。

STORE IN 句を使用して、自動生成されたパーティションまたはサブパーティションが格納されるテーブルスペースリストを指定します。

!!! Note * Advanced Server をアップグレード処理場合、アップグレードされたバージョンのサーバーでハッシュパーティションテーブルを再構築する必要があります。

TABLESPACE キーワードを使用して、パーティションまたはサブパーティションが存在するテーブルスペースの名前を指定します。テーブルスペースを指定しない場合、パーティションまたはサブパーティションはデフォルトテーブルスペースに存在します。

テーブル定義に SUBPARTITION BY 句が含まれている場合、そのテーブル内の各パーティションには少なくとも 1 つのサブパーティションがあります。各サブパーティションは、明示的に定義することもシステムで定義することもできます。

サブパーティションがシステム定義の場合、サーバー生成のサブパーティションはデフォルトテーブルスペースに存在し、サブパーティションの名前はサーバーによって割り当てられます。サーバーは以下を作成します。

- SUBPARTITION BY 句で LIST が指定されている場合、DEFAULT サブパーティション。
- SUBPARTITION BY 句で RANGE が指定されている場合、MAXVALUE サブパーティション。

サーバーは、パーティションテーブル名前と一意の識別子の組み合わせであるサブパーティション名前を生成します。ALL_TAB_SUBPARTITIONS テーブルをクエリーして、サブパーティション名の完全なリストを確認できます。

SUBPARTITION TEMPLATE をリスト、レンジ、またはハッシュサブパーティションで使用して、サブパーティションを明示的に定義していない各テーブルパーティションのデフォルトのサブパーティションを作成します。サブパーティションテンプレートの設定の詳細については、[ALTER TABLE...SET SUBPARTITION TEMPLATE](#) を参照してください。

パラメータ

table_name

作成するテーブルの名前（オプションでスキーマ修飾）。

table_definition

CREATE TABLE ステートメントの PostgreSQL コア文書で説明されている列名、データ型、および制約情報。

< <https://www.postgresql.org/docs/current/sql-createtable.html> >

partition_name

作成するパーティションの名前。パーティション名は、すべてのパーティションとサブパーティション間で一意であり、オブジェクト識別子の命名規則に従う必要があります。

subpartition_name

作成するサブパーティションの名前。サブパーティション名は、すべてのパーティションとサブパーティション間で一意であり、オブジェクト識別子の命名規則に従う必要があります。

num

PARTITIONS num 句を使用して、作成する **HASH** パーティションの数を指定できます。または、パーティション定義を使用して、個々のパーティションとその表領域を指定できます。

注： テーブルの作成時に **PARTITIONS** または **PARTITION DEFINITION** を使用できます。

SUBPARTITIONS num 句は、**HASH** パーティションでのみサポートされており、作成するハッシュサブパーティションの数を指定するために使用できます。または、サブパーティション定義を使用して、個々のサブパーティションとその表領域を指定できます。**SUBPARTITIONS** または **SUBPARTITION DEFINITION** が指定されていない場合、パーティションは **SUBPARTITION TEMPLATE** に基づいてサブパーティションを作成します。

subpartition_template_description

サブパーティション名前、バインドされたサブパーティション、およびテーブルスペース名前を含むサブパーティションの詳細のリスト。

column

パーティショニングルールのベースとなる列の名前。行は、指定された列の **value** に対応するパーティションに保存されます。

constant | expression

constant および **expression** は、**NUMERIC**、**DATE** または **TIME** 値を指定します。

(value[, value]...)

value を使用して、引用されたリテラル値（またはリテラル値のコンマ区切りリスト）を指定します。これにより、テーブルエントリがパーティションにグループ化されます。各パーティショニングルールは少なくとも 1 つの値を指定する必要がありますが、ルール内で指定される値の数に制限はありません。**value** は、**NULL**、**DEFAULT** (**LIST** パーティションを指定する場合) または **MAXVALUE** (**RANGE** パーティションを指定する場合) になります。

リストパーティションテーブルのルールを指定する場合、最後のパーティションルールに **DEFAULT** キーワードを含めて、一致しない行を特定のパーティションに転送します。**DEFAULT** の値を含むルールを含めない場合、少なくとも 1 つのパーティションの指定されたルールにマッチしない行を追加しようとする **INSERT** ステートメントは失敗し、エラーを結果めます。

リストパーティションテーブルのルールを指定する場合、最後のパーティションルールに **MAXVALUE** キーワードを含めて、未分類の行を特定のパーティションに転送します。**MAXVALUE** パーティションを含めない場合、パーティショニングキーが指定された最大値より大きい行を追加しようとする **INSERT** ステートメントは失敗し、エラーが結果れます。

tablespace_name

パーティションまたはサブパーティションが存在するテーブルスペースの名前。

3.1.1 Example - PARTITION BY LIST

</div>

次の例では、PARTITION BY LIST 句を使用してパーティションテーブル (sales) を作成します。sales テーブルは、3 つのパーティション (europe, asia および americas) に情報を格納します。

```
CREATE TABLE sales
(
  * dept_no      number,
  * part_no      varchar2,
  * country      varchar2(20),
  * date         date,
  * amount       number
)
PARTITION BY LIST(country)
(
  * PARTITION europe VALUES('FRANCE', 'ITALY'),
  * PARTITION asia  VALUES('INDIA', 'PAKISTAN'),
  * PARTITION americas VALUES('US', 'CANADA')
);
```

結果のテーブルは、country 列で指定された値によってパーティション分割されます。

```
edb=# SELECT partition_name, high_value from ALL_TAB_PARTITIONS;
partition_name |      high_value
-----+-----
EUROPE         | 'FRANCE', 'ITALY'
ASIA           | 'INDIA', 'PAKISTAN'
AMERICAS       | 'US', 'CANADA'
(3 rows)
```

- country 列の値が FRANCE または ITALY である行は、europe パーティションに格納されます。
- country 列の値が INDIA または PAKISTAN である行は、asia パーティションに格納されます。
- country 列の値が US または CANADA である行は、americas パーティションに格納されます。

サーバーは、パーティショニングルールに対して次のステートメントを評価し、europe パーティションに行を格納します。

```
INSERT INTO sales VALUES (10, '9519a', 'FRANCE', '18-Aug-2012', '650000');
```

3.1.2 Example - AUTOMATIC LIST PARTITION

</div>

次の例は、AUTOMATIC 句を使用して sales_state 列に自動リストパーティションテーブルを作成する (sales) テーブルを示しています。データベースは新しいパーティションを作成し、テーブルにデータを追加します。

```
CREATE TABLE sales
(
  * dept_no      number,
  * part_no      varchar2,
  * sales_state  varchar2(20),
  * date         date,
  * amount       number
)
PARTITION BY LIST(sales_state) AUTOMATIC
(
  * PARTITION P_CAL VALUES('CALIFORNIA'),
  * PARTITION P_FLO VALUES('FLORIDA')
);
```

ALL_TAB_PARTITIONS ビューを照会して、正常に作成された既存のパーティションを確認します。

```
edb=# SELECT table_name, partition_name, high_value from ALL_TAB_PARTITIONS;
table_name | partition_name | high_value
-----+-----+-----
SALES      | P_CAL          | 'CALIFORNIA'
SALES      | P_FLO          | 'FLORIDA'
(2 rows)
```

次に、既存のパーティションに収まらない sales テーブルにデータを挿入します。通常のリストパーティションテーブルの場合、エラーが発生しますが、リストの自動パーティショニングによりデータが作成され、新しいパーティションに挿入されます。

```
edb=# INSERT INTO sales VALUES (1, 'IND', 'INDIANA');
INSERT 0 1
edb=# INSERT INTO sales VALUES (2, 'OHI', 'OHIO');
INSERT 0 1
```

次に、挿入後に ALL_TAB_PARTITIONS ビューを再度クエリーします。パーティションが自動的に作成され、新しい値を保持するためにデータが挿入されます。セッションごとに異なるシステム生成のパーティション名前が作成されます。

```
edb=# SELECT table_name, partition_name, high_value from ALL_TAB_PARTITIONS;
table_name | partition_name | high_value
-----+-----+-----
```

(次のページに続く)

(前のページからの続き)

SALES	P_CAL	'CALIFORNIA'
SALES	P_FLO	'FLORIDA'
SALES	SYS106900103	'INDIANA'
SALES	SYS106900104	'OHIO'
(4 rows)		

3.1.3 Example - PARTITION BY RANGE

</ div>

次の例では、PARTITION BY RANGE 句を使用してパーティションテーブル (sales) を作成します。sales テーブルは、4 つのパーティション (q1_2012, q2_2012, q3_2012 および q4_2012) に情報を格納します。

```
CREATE TABLE sales
(
  * dept_no      number,
  * part_no      varchar2,
  * country      varchar2(20),
  * date         date,
  * amount      number
)
PARTITION BY RANGE(date)
(
  * PARTITION q1_2012
  *   VALUES LESS THAN('2012-Apr-01'),
  * PARTITION q2_2012
  *   VALUES LESS THAN('2012-Jul-01'),
  * PARTITION q3_2012
  *   VALUES LESS THAN('2012-Oct-01'),
  * PARTITION q4_2012
  *   VALUES LESS THAN('2013-Jan-01')
);
```

結果のテーブルは、date 列で指定された値によってパーティション分割されます。

```
edb=# SELECT partition_name, high_value from ALL_TAB_PARTITIONS;
partition_name |      high_value
-----+-----
Q1_2012        | '01-APR-12 00:00:00'
Q2_2012        | '01-JUL-12 00:00:00'
Q3_2012        | '01-OCT-12 00:00:00'
Q4_2012        | '01-JAN-13 00:00:00'
(4 rows)
```

- 2012 年 4 月 1 日より前の date 列に値を持つ行は、q1_2012 名前付けのパーティションに保存されます。
- 2012 年 7 月 1 日より前の date 列の値を持つ行は、q2_2012 名前付けのパーティションに保存されます。
- 2012 年 10 月 1 日より前の date 列の値を持つ行は、q3_2012 名前付けのパーティションに保存されます。
- date 行の値が 2013 年 1 月 1 日より前の行は、q4_2012 名前付けのパーティションに保存されます。

サーバーは、パーティショニングルールに対して次の文を評価し、行を q3_2012 パーティションに格納します。

```
INSERT INTO sales VALUES (10, '9519a', 'FRANCE', '18-Aug-2012', '650000');
```

3.1.4 Example - INTERVAL RANGE PARTITION

</ div>

次の例は、sold_month 列のインターバルでパーティション化された (sales) テーブルを示しています。レンジパーティションは、移行ポイントを確認するために作成され、新しいパーティションはその移行ポイントを超えて作成されます。データベースは新しいインターバルレンジパーティションを作成し、データをテーブルに追加します。

```
CREATE TABLE sales
(
  * prod_id          int,
  * prod_quantity    int,
  * sold_month        date
)
PARTITION BY RANGE(sold_month)
INTERVAL(NUMTOYMINTERVAL(1, 'MONTH'))
(
  * PARTITION p1
  *   VALUES LESS THAN('15-JAN-2019'),
  * PARTITION p2
  *   VALUES LESS THAN('15-FEB-2019')
);
```

最初に、インターバルレンジパーティションがデータベースによって作成される前に、ALL_TAB_PARTITIONS ビューをクエリーします。

```
edb=# SELECT partition_name, high_value from ALL_TAB_PARTITIONS;
partition_name |      high_value
-----+-----
P1              | '15-JAN-19 00:00:00'
P2              | '15-FEB-19 00:00:00'
(2 rows)
```

次に、レンジパーティションの高値を超えるデータを sales テーブルに挿入します。

```
edb=# INSERT INTO sales VALUES (1,200,'10-MAY-2019');
INSERT 0 1
```

次に、挿入後に ALL_TAB_PARTITIONS ビューを再度クエリーします。データが正常に挿入され、セッションごとに異なるインターバルレンジパーティションのシステム生成名前が作成されます。

```
edb=# SELECT partition_name, high_value from ALL_TAB_PARTITIONS;
 partition_name |      high_value
-----+-----
 P1              | '15-JAN-19 00:00:00'
 P2              | '15-FEB-19 00:00:00'
 SYS916340103    | '15-MAY-19 00:00:00'
(3 rows)
```

3.1.5 Example - PARTITION BY HASH

</div>

次の例では、PARTITION BY HASH 句を使用してパーティションテーブル (sales) を作成します。sales テーブルは、3 つのパーティション (p1, p2 および p3) に情報を格納します。

```
CREATE TABLE sales
(
  * dept_no      number,
  * part_no      varchar2,
  * country      varchar2(20),
  * date         date,
  * amount       number
)
PARTITION BY HASH (part_no)
(
  * PARTITION p1,
  * PARTITION p2,
  * PARTITION p3
);
```

テーブルは、part_no 列で指定されたハッシュパーティション値に空の文字列を結果ます。

```
edb=# SELECT partition_name, high_value from ALL_TAB_PARTITIONS;
 partition_name | high_value
-----+-----
 P1              |
 P2              |
```

(次のページに続く)

(前のページからの続き)

```
P3
(3 rows)
```

次のコマンドを使用して、`part_no` 列のハッシュ値をビューします。

```
edb=# \d+ sales
*
Partitioned table "public.sales"
Column |          Type          | Collation | Nullable | Default | Storage |
-----+-----+-----+-----+-----+-----+
dept_no | numeric                |           |          |         | main    |
part_no | character varying      |           |          |         | extended|
country | character varying(20)  |           |          |         | extended|
date    | timestamp without time zone |         |          |         | plain   |
amount  | numeric                |           |          |         | main    |
Partition key: HASH (part_no)
Partitions: sales_p1 FOR VALUES WITH (modulus 3, remainder 0),
*
           sales_p2 FOR VALUES WITH (modulus 3, remainder 1),
*
           sales_p3 FOR VALUES WITH (modulus 3, remainder 2)
```

テーブルは、`part_no` 列で指定された値のハッシュ値によってパーティション分割されます。

```
edb=# SELECT partition_name, partition_position from ALL_TAB_PARTITIONS;
partition_name | partition_position
-----+-----
P1              | 1
P2              | 2
P3              | 3
(3 rows)
```

サーバーは `part_no` 列のハッシュ値を評価し、行をほぼ等しいパーティションに分散します。

3.1.6 Example - PARTITION BY HASH...PARTITIONS num...

</ div>

次の例では、`PARTITION BY HASH` 句を使用してハッシュパーティションテーブル (`sales`) を作成します。パーティショニング化列は `part_no` です。作成するパーティションの数が指定されます。

```
CREATE TABLE sales
(
  * dept_no    number,
  * part_no    varchar2,
  * country    varchar2(20),
  * date       date,
  * amount     number
```

(次のページに続く)

(前のページからの続き)

```
)
PARTITION BY HASH (part_no) PARTITIONS 8;
```

8つのパーティションが作成され、システム生成の名前が割り当てられます。パーティションは、テーブルのデフォルトテーブルスペースのテーブルスペースに保存されます。

```
edb=# SELECT table_name, partition_name FROM ALL_TAB_PARTITIONS WHERE
table_name = 'SALES' ORDER BY 1,2;
 table_name | partition_name
-----+-----
SALES      | SYS0101
SALES      | SYS0102
SALES      | SYS0103
SALES      | SYS0104
SALES      | SYS0105
SALES      | SYS0106
SALES      | SYS0107
SALES      | SYS0108
(8 rows)
```

例-ハッシュによるパーティション…パーティション num …ストアイン

次の例では、(sales) 名前付けのハッシュパーティションテーブルを作成します。作成するパーティションの数と、パーティションが存在するテーブルスペースを指定します。

```
CREATE TABLE sales
(
  * dept_no      number,
  * part_no      varchar2,
  * country      varchar2(20),
  * date         date,
  * amount       number
)
PARTITION BY HASH (part_no) PARTITIONS 5 STORE IN (ts1, ts2, ts3);
```

STORE IN 句は、指定されたテーブルスペース (ts1, ts2, ts3) にパーティションを均等に分散します。

```
edb=# SELECT table_name, partition_name, tablespace_name FROM
ALL_TAB_PARTITIONS WHERE table_name = 'SALES' ORDER BY 1,2;
 table_name | partition_name | tablespace_name
-----+-----+-----
SALES      | SYS0101       | TS1
SALES      | SYS0102       | TS2
SALES      | SYS0103       | TS3
```

(次のページに続く)

(前のページからの続き)

SALES	SYS0104	TS1
SALES	SYS0105	TS2
(5 rows)		

例-HASH / RANGE PARTITIONS num ...

HASH パーティション句を使用すると、パーティショニング戦略を定義できます。PARTITION BY HASH 句を拡張して、SUBPARTITION BY または [RANGE | LIST | HASH] を含めて HASH パーティション表にサブパーティションを作成できます。

次の例では、part_no によってハッシュパーティション化され、dept_no によってレンジを使用してサブパーティション化されたテーブル (sales) を作成します。パーティションの数は、テーブル (sales) の作成時に指定されます。

```
CREATE TABLE sales
(
  * dept_no      number,
  * part_no      varchar2,
  * country      varchar2(20),
  * date         date,
  * amount      number
)
PARTITION BY HASH (part_no) SUBPARTITION BY RANGE (dept_no) PARTITIONS 5;
```

5つのパーティションがデフォルトのサブパーティションで作成され、システム生成の名前が割り当てられます。

```
edb=# SELECT table_name, partition_name, subpartition_name FROM
ALL_TAB_SUBPARTITIONS WHERE table_name = 'SALES' ORDER BY 1,2;
 table_name | partition_name | subpartition_name
-----+-----+-----
SALES      | SYS0101       | SYS0102
SALES      | SYS0103       | SYS0104
SALES      | SYS0105       | SYS0106
SALES      | SYS0107       | SYS0108
SALES      | SYS0109       | SYS0110
(5 rows)
```

例-LIST / HASH SUBPARTITIONS num ...

次の例は、country によってリストパーティション化され、dept_no 列によるハッシュパーティショニングを使用してサブパーティション化されたテーブル (sales) を示しています。サブパーティションの数は、テーブルの作成時に指定されます。

```
CREATE TABLE sales
(
  * dept_no      number,
  * part_no      varchar2,
  * country      varchar2(20),
  * date         date,
  * amount       number
)
PARTITION BY LIST (country) SUBPARTITION BY HASH (dept_no) SUBPARTITIONS 3
(
  * PARTITION p1 VALUES('FRANCE', 'ITALY'),
  * PARTITION p2 VALUES('INDIA', 'PAKISTAN'),
  * PARTITION p3 VALUES('US', 'CANADA')
);
```

3つのパーティション p1, p2 および p3 には、それぞれシステム生成の名前を持つ3つのサブパーティションが含まれています。

```
edb=# SELECT table_name, partition_name, subpartition_name FROM
ALL_TAB_SUBPARTITIONS WHERE table_name = 'SALES' ORDER BY 1,2;
 table_name | partition_name | subpartition_name
-----+-----+-----
SALES      | P1             | SYS0101
SALES      | P1             | SYS0102
SALES      | P1             | SYS0103
SALES      | P2             | SYS0104
SALES      | P2             | SYS0105
SALES      | P2             | SYS0106
SALES      | P3             | SYS0107
SALES      | P3             | SYS0108
SALES      | P3             | SYS0109
(9 rows)
```

例-HASH / HASH PARTITIONS num ... SUBPARTITIONS num ...

次の例では、part_no によってハッシュ分割され、dept_no によってハッシュ分割された (sales) テーブルを作成します。

```
CREATE TABLE sales
(
  * dept_no      number,
  * part_no      varchar2,
  * country      varchar2(20),
  * date         date,
  * amount       number
)
PARTITION BY HASH (part_no) SUBPARTITION BY HASH (dept_no) SUBPARTITIONS 3
PARTITIONS 2;
```

2つのパーティションが作成され、各パーティションには、システム生成の名前が割り当てられた3つのサブパーティションが含まれます。

```
edb=# SELECT table_name, partition_name, subpartition_name FROM
ALL_TAB_SUBPARTITIONS WHERE table_name = 'SALES' ORDER BY 1,2;
 table_name | partition_name | subpartition_name
-----+-----+-----
SALES      | SYS0101       | SYS0102
SALES      | SYS0101       | SYS0103
SALES      | SYS0101       | SYS0104
SALES      | SYS0105       | SYS0106
SALES      | SYS0105       | SYS0107
SALES      | SYS0105       | SYS0108
(6 rows)
```

例-HASH / HASH SUBPARTITIONS num ... STORE IN

次の例では、ハッシュパーティションテーブル (sales) を作成します。作成するパーティションとサブパーティションの数は、ハッシュパーティションテーブルを作成するときにサブパーティションが存在するテーブルスペースとともに指定されます。

```
CREATE TABLE sales
(
  * dept_no      number,
  * part_no      varchar2,
  * country      varchar2(20),
  * date         date,
  * amount       number
)
```

(次のページに続く)

(前のページからの続き)

```
PARTITION BY HASH (part_no) SUBPARTITION BY HASH (dept_no) SUBPARTITIONS 3
PARTITIONS 2 STORE IN (ts1, ts2);
```

2つのパーティションが作成され、システム生成の名前が割り当てられます。パーティションはデフォルトテーブルスペースのテーブルスペースに保存され、サブパーティションはテーブルスペース (ts1) および (ts2) に保存されます。

```
edb=# SELECT table_name, partition_name, tablespace_name FROM
ALL_TAB_PARTITIONS WHERE table_name = 'SALES' ORDER BY 1,2;
```

table_name	partition_name	tablespace_name
SALES	SYS0101	
SALES	SYS0105	

(2 rows)

STORE IN 句は、ハッシュサブパーティションをテーブルスペースに割り当て、2つの名前付けテーブルスペース (ts1, ts2) に保存します。

```
edb=# SELECT table_name, partition_name, subpartition_name, tablespace_name
FROM ALL_TAB_SUBPARTITIONS WHERE table_name = 'SALES' ORDER BY 1,2;
```

table_name	partition_name	subpartition_name	tablespace_name
SALES	SYS0101	SYS0102	TS1
SALES	SYS0101	SYS0103	TS2
SALES	SYS0101	SYS0104	TS1
SALES	SYS0105	SYS0106	TS1
SALES	SYS0105	SYS0107	TS2
SALES	SYS0105	SYS0108	TS1

(6 rows)

例-HASH / HASH PARTITIONS num ... STORE IN SUBPARTITIONS num ... STORE IN

次の例では、ハッシュパーティションテーブル (sales) を作成します。作成するパーティションとサブパーティションの数は、パーティションとサブパーティションが存在する表領域とともに指定されます。

```
CREATE TABLE sales
(
  * dept_no      number,
  * part_no      varchar2,
  * country      varchar2(20),
  * date         date,
  * amount       number
)
```

(次のページに続く)

(前のページからの続き)

```
PARTITION BY HASH (part_no) SUBPARTITION BY HASH (dept_no) SUBPARTITIONS 3
STORE IN (ts3) PARTITIONS 2 STORE IN (ts1, ts2);
```

2つのパーティションはシステム生成の名前で作成され、デフォルトテーブルスペースに保存されます。

```
edb=# SELECT table_name, partition_name, tablespace_name FROM
ALL_TAB_PARTITIONS WHERE table_name = 'SALES' ORDER BY 1,2;
 table_name | partition_name | tablespace_name
-----+-----+-----
SALES      | SYS0101       |
SALES      | SYS0105       |
(2 rows)
```

各パーティションには3つのサブパーティションが含まれます。STORE IN 句は、サブパーティションをテーブルスペース (ts1) および (ts2) に格納します。

```
edb=# SELECT table_name, partition_name, subpartition_name, tablespace_name
FROM ALL_TAB_SUBPARTITIONS WHERE table_name = 'SALES' ORDER BY 1,2;
 table_name | partition_name | subpartition_name | tablespace_name
-----+-----+-----+-----
SALES      | SYS0101       | SYS0102           | TS1
SALES      | SYS0101       | SYS0103           | TS2
SALES      | SYS0101       | SYS0104           | TS1
SALES      | SYS0105       | SYS0106           | TS1
SALES      | SYS0105       | SYS0107           | TS2
SALES      | SYS0105       | SYS0108           | TS1
(6 rows)
```

!!! Note * STORE IN 句がパーティションおよびサブパーティションに指定されている場合、サブパーティションは PARTITIONS...STORE IN 句で定義された表領域に格納され、SUBPARTITIONS...STORE IN 句は無視されます。

例-RANGE / HASH SUBPARTITIONS num ...

次の例では、トランザクションの日付で最初にパーティション化される範囲パーティション表 (sales) を作成します。2つのレンジパーティションが作成され、country 列の値を使用してハッシュサブパーティション化されます。

```
CREATE TABLE sales
(
  * dept_no      number,
  * part_no      varchar2,
  * country      varchar2(20),
  * date         date,
```

(次のページに続く)

(前のページからの続き)

```

    * amount      number
)
PARTITION BY RANGE (date) SUBPARTITION BY HASH (country) SUBPARTITIONS 2
(
    * PARTITION p1 VALUES LESS THAN ('2012-Apr-01') (SUBPARTITION q1_europe),
    * PARTITION p2 VALUES LESS THAN ('2012-Jul-01')
);

```

このステートメントは、2つのパーティションを持つテーブルを作成します。明示的に **q1_europe** 名前付けのサブパーティションがパーティション **p1** 用に作成されます。サブパーティションにはパーティション **p2** の名前が名前付けていないため、サブパーティションはサブパーティション番号に基づいて作成され、システム生成の名前が割り当てられます。

```

edb=# SELECT table_name, partition_name, subpartition_name FROM
ALL_TAB_SUBPARTITIONS WHERE table_name = 'SALES' ORDER BY 1,2;
 table_name | partition_name | subpartition_name
-----+-----+-----
SALES      | P1             | Q1_EUROPE
SALES      | P2             | SYS0101
SALES      | P2             | SYS0102
(3 rows)

```

例-RANGE / HASH SUBPARTITIONS num ... IN PARTITION DESCRIPTION

次の例では、トランザクションの日付で最初にパーティション化される範囲パーティション表 (**sales**) を作成します。2つのレンジパーティションが作成され、**country** 列の値を使用してハッシュサブパーティション化されます。

```

CREATE TABLE sales
(
    * dept_no      number,
    * part_no      varchar2,
    * country      varchar2(20),
    * date         date,
    * amount       number
)
PARTITION BY RANGE (date) SUBPARTITION BY HASH (country) SUBPARTITIONS 2
(
    * PARTITION p1 VALUES LESS THAN ('2012-Apr-01') SUBPARTITIONS 3,
    * PARTITION p2 VALUES LESS THAN ('2012-Jul-01')
);

```

パーティション **p1** は、パーティションの説明でサブパーティションカウントを明示的に定義します。デフォルトでは、パーティション **p2** に2つのサブパーティションが作成されます。サブパーティションには名前が名前付けていないため、システム生成の名前が割り当てられます。

```
edb=# SELECT table_name, partition_name, subpartition_name FROM
ALL_TAB_SUBPARTITIONS WHERE table_name = 'SALES' ORDER BY 1,2;
 table_name | partition_name | subpartition_name
-----+-----+-----
SALES      | P1             | SYS0101
SALES      | P1             | SYS0102
SALES      | P1             | SYS0103
SALES      | P2             | SYS0104
SALES      | P2             | SYS0105
(5 rows)
```

例-LIST / HASH SUBPARTITIONS num STORE IN ... IN PARTITION DESCRIPTION

次の例では、2つのリストパーティションを持つリストパーティションテーブル (sales) を作成します。パーティション p1 は3つのサブパーティションで構成され、パーティション p2 は2つのサブパーティションで構成されます。サブパーティションには名前が名前付けていないため、システム生成の名前が割り当てられます。

```
CREATE TABLE sales
(
  * dept_no      number,
  * part_no      varchar2,
  * country      varchar2(20),
  * date         date,
  * amount       number
)
PARTITION BY LIST (country) SUBPARTITION BY HASH (part_no) SUBPARTITIONS 3
STORE IN (ts1)
(
  * PARTITION p1 VALUES ('FRANCE', 'ITALY'),
  * PARTITION p2 VALUES ('INDIA', 'PAKISTAN') SUBPARTITIONS 2 STORE IN
  * (ts2)
);
```

パーティション p2 は、パーティションの説明でサブパーティションカウントを明示的に定義します。定義に基づいて、2つのサブパーティションが作成され、(ts2) 名前付けのテーブルスペースに格納されます。パーティション p1 のサブパーティションは、(ts1) 名前付けのテーブルスペースに格納されます。

```
edb=# SELECT table_name, partition_name, subpartition_name, tablespace_name
FROM ALL_TAB_SUBPARTITIONS WHERE table_name = 'SALES' ORDER BY 1,2;
 table_name | partition_name | subpartition_name | tablespace_name
-----+-----+-----+-----
SALES      | P1             | SYS0101           | TS1
SALES      | P1             | SYS0102           | TS1
```

(次のページに続く)

(前のページからの続き)

SALES	P1	SYS0103	TS1
SALES	P2	SYS0104	TS2
SALES	P2	SYS0105	TS2

(5 rows)

例-LIST / HASH STORE IN ... TABLESPACES

次の例では、テーブルスペース (ts2) に明示的に格納された 3 つのサブパーティションで構成されるパーティション p1 を使用して、リストパーティションテーブル (sales) を作成します。

```
CREATE TABLE sales
(
  * dept_no      number,
  * part_no      varchar2,
  * country      varchar2(20),
  * date         date,
  * amount       number
)
PARTITION BY LIST (country) SUBPARTITION BY HASH (part_no) SUBPARTITIONS 3
STORE IN (ts1)
(
  * PARTITION p1 VALUES ('FRANCE', 'ITALY') TABLESPACE ts2
);
```

SELECT ステートメントは、テーブルスペース (ts2) に格納されている 3 つのサブパーティションで構成されるパーティション p1 を示しています。

```
edb=# SELECT table_name, partition_name, subpartition_name, tablespace_name
FROM ALL_TAB_SUBPARTITIONS WHERE table_name = 'SALES' ORDER BY 1,2;
 table_name | partition_name | subpartition_name | tablespace_name
-----+-----+-----+-----
SALES      | P1            | SYS0101          | TS2
SALES      | P1            | SYS0102          | TS2
SALES      | P1            | SYS0103          | TS2
(3 rows)
```

次のコマンドは、新しいパーティション p2 を sales テーブルに追加します。5 つのサブパーティションが作成され、STORE IN 句によってリストされたテーブルスペース全体に分散されます。

```
ALTER TABLE sales ADD PARTITION p2 VALUES ('US', 'CANADA') SUBPARTITIONS 5
STORE IN (ts1);
```

ALL_TAB_PARTITIONS ビューのクエリーは、p2 名前付けのパーティションと 5 つのサブパーティションを持つ sales テーブルを示しています。STORE IN 句は、(ts1) 名前付けのテーブルスペースにサブパーティションを分散します。

```
edb=# SELECT table_name, partition_name, subpartition_name, tablespace_name
FROM ALL_TAB_SUBPARTITIONS WHERE table_name = 'SALES' and partition_name =
'P2' ORDER BY 1,2;
 table_name | partition_name | subpartition_name | tablespace_name
-----+-----+-----+-----
SALES      | P2             | SYS0105           | TS1
SALES      | P2             | SYS0106           | TS1
SALES      | P2             | SYS0107           | TS1
SALES      | P2             | SYS0108           | TS1
SALES      | P2             | SYS0109           | TS1
(5 rows)
```

3.1.7 Example - PARTITION BY RANGE, SUBPARTITION BY LIST

</ div>

次の例では、トランザクション日付で最初にパーティション化されるパーティション表 (sales) を作成します。レンジパーティション (q1_2012, q2_2012, q3_2012 および q4_2012) は、country 列の値を使用してリストサブパーティション化されます。

```
CREATE TABLE sales
(
  * dept_no      number,
  * part_no      varchar2,
  * country      varchar2(20),
  * date         date,
  * amount       number
)
PARTITION BY RANGE(date)
  * SUBPARTITION BY LIST(country)
  * (
  *   PARTITION q1_2012
  *     VALUES LESS THAN('2012-Apr-01')
  *     (
  *       SUBPARTITION q1_europe VALUES ('FRANCE', 'ITALY'),
  *       SUBPARTITION q1_asia VALUES ('INDIA', 'PAKISTAN'),
  *       SUBPARTITION q1_americas VALUES ('US', 'CANADA')
  *     ),
  *   PARTITION q2_2012
  *     VALUES LESS THAN('2012-Jul-01')
  *     (
  *       SUBPARTITION q2_europe VALUES ('FRANCE', 'ITALY'),
  *       SUBPARTITION q2_asia VALUES ('INDIA', 'PAKISTAN'),
  *       SUBPARTITION q2_americas VALUES ('US', 'CANADA')
```

(次のページに続く)

(前のページからの続き)

```

*      ),
* PARTITION q3_2012
*   VALUES LESS THAN('2012-Oct-01')
*   (
*       SUBPARTITION q3_europe VALUES ('FRANCE', 'ITALY'),
*       SUBPARTITION q3_asia VALUES ('INDIA', 'PAKISTAN'),
*       SUBPARTITION q3_americas VALUES ('US', 'CANADA')
*   ),
* PARTITION q4_2012
*   VALUES LESS THAN('2013-Jan-01')
*   (
*       SUBPARTITION q4_europe VALUES ('FRANCE', 'ITALY'),
*       SUBPARTITION q4_asia VALUES ('INDIA', 'PAKISTAN'),
*       SUBPARTITION q4_americas VALUES ('US', 'CANADA')
*   )
);

```

このステートメントは、4つのパーティションを持つテーブルを作成します。各パーティションには3つのサブパーティションがあります。

```

edb=# SELECT subpartition_name, high_value, partition_name FROM ALL_TAB_SUBPARTITIONS;
subpartition_name |      high_value      | partition_name
-----+-----+-----
Q1_EUROPE         | 'FRANCE', 'ITALY'    | Q1_2012
Q1_ASIA           | 'INDIA', 'PAKISTAN'  | Q1_2012
Q1_AMERICAS       | 'US', 'CANADA'       | Q1_2012
Q2_EUROPE         | 'FRANCE', 'ITALY'    | Q2_2012
Q2_ASIA           | 'INDIA', 'PAKISTAN'  | Q2_2012
Q2_AMERICAS       | 'US', 'CANADA'       | Q2_2012
Q3_EUROPE         | 'FRANCE', 'ITALY'    | Q3_2012
Q3_ASIA           | 'INDIA', 'PAKISTAN'  | Q3_2012
Q3_AMERICAS       | 'US', 'CANADA'       | Q3_2012
Q4_EUROPE         | 'FRANCE', 'ITALY'    | Q4_2012
Q4_ASIA           | 'INDIA', 'PAKISTAN'  | Q4_2012
Q4_AMERICAS       | 'US', 'CANADA'       | Q4_2012
(12 rows)

```

このテーブルに行が追加されると、**date** 列の値がレンジパーティショニングルールで指定された値と比較され、サーバーは行が存在するパーティションを選択します。**country** 列の値は、リストのサブパーティションルールで指定された値と比較されます。サーバーが値にマッチするものを見つけると、行は対応するサブパーティションに格納されます。

テーブルに追加された行はサブパーティションに保存されるため、パーティションにはデータが含まれません。

サーバーは、パーティショニングおよびサブパーティション化ルールに対して次の文を評価し、行を **q3_europe** パーティションに格納します。

```
INSERT INTO sales VALUES (10, '9519a', 'FRANCE', '18-Aug-2012', '650000');
```

3.1.8 Example - CREATING UNIQUE INDEX on PARTITION TABLE

</div>

一意インデックステーブルパーティションの場合、レンジ、リスト、またはハッシュパーティショニングメソッドを使用できます。パーティション表に一意の ROWID を作成するには、パラメータ `default_with_rowids = true` を設定する必要があります。

次の例では、`PARTITION BY LIST` 句を使用してパーティションテーブル (`sales`) を作成します。`sales` テーブルは、3 つのパーティション (`europe`, `asia` および `americas`) に情報を格納します。

```
CREATE TABLE sales
(
  * dept_no      number,
  * part_no      varchar2,
  * country      varchar2(20),
  * date         date,
  * amount       number
)
PARTITION BY LIST(country)
(
  * PARTITION europe VALUES('FRANCE', 'ITALY'),
  * PARTITION asia  VALUES('INDIA', 'PAKISTAN'),
  * PARTITION americas VALUES('US', 'CANADA')
)
WITH(ROWIDS=TRUE);
```

テーブルは、`country` 列で指定された値によってパーティション分割されます。

```
edb=# SELECT partition_name, high_value from ALL_TAB_PARTITIONS;
partition_name |      high_value
-----+-----
EUROPE         | 'FRANCE', 'ITALY'
ASIA           | 'INDIA', 'PAKISTAN'
AMERICAS       | 'US', 'CANADA'
(3 rows)
```

次に、`sales` テーブルに値を挿入します。

```
INSERT INTO sales VALUES (10, '9519a', 'FRANCE', '18-Aug-2012', '650000');
INSERT INTO sales VALUES (10, '9519b', 'ITALY', '20-Aug-2012', '700000');
```

次のクエリーは、`sales` テーブルに一意インデックスが作成されることを示しています。

```
edb=# SELECT oid, relname, relhasindex from pg_class where relname='sales';
   oid | relname | relhasindex
-----+-----+-----
 16557 | sales   | f
(1 row)
```

sales_europe の内容を照会すると、テーブルに一意インデックスが作成されていることが確認されます。

```
edb=# SELECT oid, relname, relhasindex from pg_class where relname='sales_europe';
   oid |   relname   | relhasindex
-----+-----+-----
 16561 | sales_europe | t
(1 row)
```

```

edb=# SELECT * from pg_index where indrelid='16561';
 indexrelid | indrelid | indnatts | indnkeyatts | indisunique | indisprimary |
indisexclusion | indimmediate | indisclustered | indisvalid | indcheckxmin |
indisready | indislive | indisreplident |
 indkey | indcollation | indclass | indooption | indexprs | indpred
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
↪-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
↪-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
 *          16565 |          16561 |          1 |          1 | t          |          | f          |          | f          |
          | t          |          | f          |          | t          |          | f          |          | t          | ↪
↪t          | f          |          |          |          |          |          |          |          |
 1          | 0          | 3124      | 0          |          |          |          |          |          |
(1 row)

```

3.1.9 Example - CREATING SUBPARTITION TEMPLATE

このセクションの例は、サブパーティションテンプレートを使用して、パーティションテーブルにサブパーティションを作成する方法を示しています。

次の例では、**country** によってリストパーティション化され、**date** によってリストを使用してサブパーティション化されたテーブル (**sales**) を作成します。(sales) テーブルはサブパーティションテンプレートを使用し、サブパーティションとテーブルスペース名前を表示します。

```
CREATE TABLE sales
(
    * dept_no      number,
    * part_no      varchar2,
    * country      varchar2(20),
```

(次のページに続く)

(前のページからの続き)

```

* date      date,
* amount    number
)
PARTITION BY LIST (country)
* SUBPARTITION BY LIST (date)
* SUBPARTITION TEMPLATE
(
* SUBPARTITION europe VALUES('2021-Jan-01') TABLESPACE ts1,
* SUBPARTITION asia VALUES('2021-Apr-01') TABLESPACE ts2,
* SUBPARTITION americas VALUES('2021-Jul-01') TABLESPACE ts3
)
(
* PARTITION q1_2021 VALUES('2021-Jul-01')
);

```

SELECT ステートメントは、テーブルスペース (ts1)、(ts2)、および (ts3) に格納されている 3 つのサブパーティションで構成されるパーティション q1_2021 を示しています。

```

edb=# SELECT partition_name, subpartition_name, high_value, tablespace_name
FROM DBA_TAB_SUBPARTITIONS WHERE table_name = 'SALES' ORDER BY 1,2;
partition_name | subpartition_name |      high_value      | tablespace_name
-----+-----+-----+-----
Q1_2021        | Q1_2021_AMERICAS  | '01-JUL-21 00:00:00' | TS3
Q1_2021        | Q1_2021_ASIA      | '01-APR-21 00:00:00' | TS2
Q1_2021        | Q1_2021_EUROPE    | '01-JAN-21 00:00:00' | TS1
(3 rows)

```

例-LIST / RANGE パーティションテーブルのサブパーティションテンプレートの作成

次の例では、country によってリストパーティション化され、date 列によるレンジパーティショニングを使用してサブパーティション化されたテーブル (sales) を作成します。

```

CREATE TABLE sales
(
* dept_no    number,
* part_no    varchar2,
* country    varchar2(20),
* date       date,
* amount     number
)
PARTITION BY LIST(country)
* SUBPARTITION BY RANGE(date)
* SUBPARTITION TEMPLATE
(

```

(次のページに続く)

(前のページからの続き)

```

* SUBPARTITION europe VALUES LESS THAN('2021-Jan-01') TABLESPACE ts1,
* SUBPARTITION asia VALUES LESS THAN('2021-Apr-01') TABLESPACE ts2,
* SUBPARTITION americas VALUES LESS THAN('2021-Jul-01') TABLESPACE ts3
)
(
* PARTITION q1_2021 VALUES ('2021-Jul-01')
);

```

sales テーブルは、テーブルスペース (ts1)、(ts2)、および (ts3) に格納された 3 つのサブパーティションを含む q1_2021 という名前付けのパーティションを作成します。

```

edb=# SELECT partition_name, subpartition_name, high_value, tablespace_name
FROM DBA_TAB_SUBPARTITIONS WHERE table_name = 'SALES' ORDER BY 1,2;
partition_name | subpartition_name |      high_value      | tablespace_name
-----+-----+-----+-----
Q1_2021        | Q1_2021_AMERICAS | '01-JUL-21 00:00:00' | TS3
Q1_2021        | Q1_2021_ASIA     | '01-APR-21 00:00:00' | TS2
Q1_2021        | Q1_2021_EUROPE   | '01-JAN-21 00:00:00' | TS1
(3 rows)

```

例-LIST / HASH パーティションテーブルのサブパーティションテンプレートの作成

次の例では、最初に country によってパーティション化され、次に dept_no 列の値を使用してハッシュサブパーティション化されるリストパーティションテーブル (sales) を作成します。

```

CREATE TABLE sales
(
* dept_no      number,
* part_no      varchar2,
* country      varchar2(20),
* date         date,
* amount       number
)
PARTITION BY LIST(country)
* SUBPARTITION BY HASH (dept_no)
* SUBPARTITION TEMPLATE
(
* SUBPARTITION europe TABLESPACE ts1,
* SUBPARTITION asia TABLESPACE ts2,
* SUBPARTITION americas TABLESPACE ts3
)
(
* PARTITION q1_2021 VALUES ('2021-Jul-01')
);

```

sales テーブルは、テーブルスペース (ts1)、(ts2)、および (ts3) に格納されている 3 つのサブパーティションを含む q1_2021 パーティションを作成します。

```
edb=# SELECT partition_name, subpartition_name, high_value, tablespace_name
FROM DBA_TAB_SUBPARTITIONS WHERE table_name = 'SALES' ORDER BY 1,2;
partition_name | subpartition_name | high_value | tablespace_name
-----+-----+-----+-----
Q1_2021        | Q1_2021_AMERICAS |           | TS3
Q1_2021        | Q1_2021_ASIA     |           | TS2
Q1_2021        | Q1_2021_EUROPE    |           | TS1
(3 rows)
```

例-INTERVAL / HASH パーティションテーブルのサブパーティションテンプレートの作成

次の例では、(sales) 列を作成し、date 列で毎月の間隔を使用してインターバルをパーティション化し、dept_no 列の値を使用してハッシュサブパーティションを作成します。

```
CREATE TABLE sales
(
  * dept_no      number,
  * part_no      varchar2,
  * country      varchar2(20),
  * date         date,
  * amount       number
)
PARTITION BY RANGE(date) INTERVAL(NUMTOYMINTERVAL(1, 'MONTH'))
  * SUBPARTITION BY HASH (dept_no)
  * SUBPARTITION TEMPLATE
(
  * SUBPARTITION europe TABLESPACE ts1,
  * SUBPARTITION asia TABLESPACE ts2,
  * SUBPARTITION americas TABLESPACE ts3
)
(
  * PARTITION q2_2021 VALUES LESS THAN ('2021-Jul-01')
);
```

sales テーブルは、テーブルスペース (ts1)、(ts2)、および (ts3) に格納されている 3 つのサブパーティションで構成されるパーティション q2_2021 を作成します。

```
edb=# SELECT partition_name, subpartition_name, high_value, tablespace_name
FROM DBA_TAB_SUBPARTITIONS WHERE table_name = 'SALES' ORDER BY 1,2;
partition_name | subpartition_name | high_value | tablespace_name
-----+-----+-----+-----
Q2_2021        | Q2_2021_AMERICAS |           | TS3
```

(次のページに続く)

(前のページからの続き)

Q2_2021	Q2_2021_ASIA		TS2
Q2_2021	Q2_2021_EUROPE		TS1
(3 rows)			

sales テーブルに値を挿入します。

```
INSERT INTO sales VALUES (10, '9519a', 'FRANCE', '05-Jul-2021', '650000');
```

SELECT ステートメントは、テーブルスペース (ts1)、(ts2)、および (ts3) に格納されているパーティションおよびサブパーティションのシステム生成名前を示します。

```
edb=# SELECT partition_name, subpartition_name, high_value, tablespace_name
FROM DBA_TAB_SUBPARTITIONS WHERE table_name = 'SALES' ORDER BY 1,2;
```

partition_name	subpartition_name	high_value	tablespace_name
-----+-----+-----+-----			
Q2_2021	Q2_2021_AMERICAS		TS3
Q2_2021	Q2_2021_ASIA		TS2
Q2_2021	Q2_2021_EUROPE		TS1
SYS368340105	SYS368340106		TS1
SYS368340105	SYS368340107		TS2
SYS368340105	SYS368340108		TS3
(6 rows)			

3.2 ALTER TABLE…ADD PARTITION

</div>

ALTER TABLE… ADD PARTITION コマンドを使用して、既存のパーティションテーブルにパーティションを追加します。構文は次のとおりです。

```
ALTER TABLE <table_name> ADD PARTITION <partition_definition>;

Where partition_definition is:
    *      {<list_partition> | <range_partition>}

and list_partition is:
    *      PARTITION [<partition_name>]
    *      VALUES (<value>[, <value>]...)
    *      [TABLESPACE <tablespace_name>]
    *      [(<subpartition>, ...)]

and range_partition is:
    *      PARTITION [<partition_name>]
    *      VALUES LESS THAN (<value>[, <value>]...)
```

(次のページに続く)

(前のページからの続き)

```

*      [TABLESPACE <tablespace_name>]
*      [( <subpartition>, ...)]

Where subpartition is:
*      {<list_subpartition> | <range_subpartition> | <hash_subpartition>}

and list_subpartition is:
*      SUBPARTITION [<subpartition_name>]
*      VALUES (<value>[, <value>]...)
*      [TABLESPACE <tablespace_name>]

and range_subpartition is:
*      SUBPARTITION [<subpartition_name>]
*      VALUES LESS THAN (<value>[, <value>]...)
*      [TABLESPACE <tablespace_name>]

and hash_subpartition is:
*      SUBPARTITION [<subpartition_name>]
*      [TABLESPACE <tablespace_name>] |
*      SUBPARTITIONS <num> [STORE IN ( <tablespace_name> [, <tablespace_name>]...
→) ]

```

説明

ALTER TABLE... ADD PARTITION コマンドは、既存のパーティションテーブルにパーティションを追加します。パーティションテーブルに定義されているパーティションの数に上限はありません。

新しいパーティションは、既存のパーティションと同じタイプ (**LIST**、**RANGE** または **HASH**) でなければなりません。新しいパーティションルールは、既存のパーティションを定義するパーティショニングルールで指定された同じ列をリファレンスする必要があります。

テーブル内の既存の行と追加されるパーティションの値の間に矛盾する値がない限り、**ALTER TABLE... ADD PARTITION** ステートメントを使用して、**DEFAULT** ルールでテーブルにパーティションを追加できます。

ALTER TABLE... ADD PARTITION ステートメントを使用して、**MAXVALUE** ルールを使用してパーティションをテーブルに追加することはできません。

または、**ALTER TABLE... SPLIT PARTITION** ステートメントを使用して既存のパーティションを分割し、テーブル内のパーティションの数を効果的に増やすことができます。

RANGE パーティションはオーダーで指定する必要があります。**RANGE** パーティションテーブルの既存のパーティションの前に新しいパーティションを追加することはできません。

TABLESPACE 句を含めて、新しいパーティションが存在するテーブルスペースを指定します。テーブルスペースを指定しない場合、パーティションはデフォルトテーブルスペースに存在します。

STORE IN 句を使用して、自動生成されたサブパーティションが格納されるテーブルスペースリストを指定します。

テーブルにインデックスが付けられている場合、インデックスは新しいパーティションに作成されます。

ALTER TABLE... ADD PARTITION コマンドを使用するには、テーブルの所有者であるか、スーパーユーザ（または管理）権限が必要です。

パラメータ

table_name

パーティション化されたテーブルの名前（オプションでスキーマ修飾）。

partition_name

作成するパーティションの名前。パーティション名は、すべてのパーティションとサブパーティション間で一意であり、オブジェクト識別子の命名規則に従う必要があります。

subpartition_name

作成するサブパーティションの名前。サブパーティション名は、すべてのパーティションとサブパーティション間で一意であり、オブジェクト識別子の命名規則に従う必要があります。

(value[, value]...)

value を使用して、引用符で囲まれたリテラル値（またはリテラル値のコンマ区切りリスト）を指定して、行をパーティションに分散します。各パーティショニングルールは、少なくとも 1 つの **value** を指定する必要がありますが、ルール内で指定される値の数に制限はありません。**value** は、NULL、DEFAULT (LIST パーティションを指定する場合)、または MAXVALUE (RANGE パーティションを指定する場合) にもなります。

DEFAULT または MAXVALUE パーティションの作成については、[Handling Stray Values in a LIST or RANGE Partitioned Table](#) を参照してください。

num

SUBPARTITIONS **num** 句は HASH でのみサポートされており、ハッシュサブパーティションの数を指定するために使用できます。または、サブパーティション定義を使用して、個々のサブパーティションとその表領域を指定できます。SUBPARTITIONS または SUBPARTITION DEFINITION が指定されていない場合、パーティションは SUBPARTITION TEMPLATE に基づいてサブパーティションを作成します。

tablespace_name

パーティションまたはサブパーティションが存在するテーブルスペースの名前。

3.2.1 Example - Adding a Partition to a LIST Partitioned Table

</div>

次の例では、リストパーティション **sales** テーブルにパーティションを追加します。次のコマンドを使用してテーブルが作成されました。

```
CREATE TABLE sales
(
  * dept_no      number,
  * part_no      varchar2,
  * country      varchar2(20),
```

(次のページに続く)

(前のページからの続き)

```

* date      date,
* amount    number
)
PARTITION BY LIST(country)
(
  * PARTITION europe VALUES('FRANCE', 'ITALY'),
  * PARTITION asia VALUES('INDIA', 'PAKISTAN'),
  * PARTITION americas VALUES('US', 'CANADA')
);

```

テーブルには3つのパーティション (americas, asia、および europe) が含まれます。

```

edb=# SELECT partition_name, high_value FROM ALL_TAB_PARTITIONS;
partition_name |      high_value
-----+-----
EUROPE         | 'FRANCE', 'ITALY'
ASIA           | 'INDIA', 'PAKISTAN'
AMERICAS       | 'US', 'CANADA'
(3 rows)

```

次のコマンドは、east_asia 名前付けのパーティションを sales テーブルに追加します。

```

ALTER TABLE sales ADD PARTITION east_asia
  * VALUES ('CHINA', 'KOREA');

```

コマンドを呼び出した後、テーブルには east_asia パーティションが含まれます。

```

edb=# SELECT partition_name, high_value FROM ALL_TAB_PARTITIONS;
partition_name |      high_value
-----+-----
EUROPE         | 'FRANCE', 'ITALY'
ASIA           | 'INDIA', 'PAKISTAN'
AMERICAS       | 'US', 'CANADA'
EAST_ASIA      | 'CHINA', 'KOREA'
(4 rows)

```

3.2.2 Example - Adding a Partition to a RANGE Partitioned Table

</div>

次の例では、sales 名前付けの範囲パーティションテーブルにパーティションを追加します。

```

CREATE TABLE sales
(
  * dept_no    number,

```

(次のページに続く)

(前のページからの続き)

```

* part_no      varchar2,
* country      varchar2(20),
* date         date,
* amount       number
)
PARTITION BY RANGE(date)
(
* PARTITION q1_2012
*   VALUES LESS THAN('2012-Apr-01'),
* PARTITION q2_2012
*   VALUES LESS THAN('2012-Jul-01'),
* PARTITION q3_2012
*   VALUES LESS THAN('2012-Oct-01'),
* PARTITION q4_2012
*   VALUES LESS THAN('2013-Jan-01')
);

```

テーブルには4つのパーティション (q1_2012, q2_2012, q3_2012、および q4_2012) が含まれます。

```

edb=# SELECT partition_name, high_value FROM ALL_TAB_PARTITIONS;
partition_name |      high_value
-----+-----
Q1_2012        | '01-APR-12 00:00:00'
Q2_2012        | '01-JUL-12 00:00:00'
Q3_2012        | '01-OCT-12 00:00:00'
Q4_2012        | '01-JAN-13 00:00:00'
(4 rows)

```

次のコマンドは、q1_2013 名前付けのパーティションを sales テーブルに追加します。

```

ALTER TABLE sales ADD PARTITION q1_2013
* VALUES LESS THAN('01-APR-2013');

```

コマンドを呼び出した後、テーブルには q1_2013 パーティションが含まれます。

```

edb=# SELECT partition_name, high_value FROM ALL_TAB_PARTITIONS;
partition_name |      high_value
-----+-----
Q1_2012        | '01-APR-12 00:00:00'
Q2_2012        | '01-JUL-12 00:00:00'
Q3_2012        | '01-OCT-12 00:00:00'
Q4_2012        | '01-JAN-13 00:00:00'
Q1_2013        | '01-APR-13 00:00:00'
(5 rows)

```

3.2.3 Example - Adding a Partition with SUBPARTITIONS num…IN PARTITION DESCRIPTION

</div>

次の例では、リストパーティション sales テーブルにパーティションを追加します。SUBPARTITIONS 句を指定して、指定した数のサブパーティションを追加できます。次のコマンドで sales テーブルが作成されました。

```
CREATE TABLE sales
(
  * dept_no      number,
  * part_no      varchar2,
  * country      varchar2(20),
  * date         date,
  * amount       number
)
PARTITION BY LIST (country) SUBPARTITION BY HASH (part_no) SUBPARTITIONS 2
(
  * PARTITION europe VALUES ('FRANCE', 'ITALY'),
  * PARTITION asia  VALUES ('INDIA', 'PAKISTAN')
);
```

テーブルには、それぞれ2つのサブパーティションを含む2つのパーティション europe および asia が含まれます。サブパーティションには名前が名前付けていないため、システム生成の名前が割り当てられます。

```
edb=# SELECT table_name, partition_name, subpartition_name FROM
ALL_TAB_SUBPARTITIONS WHERE table_name = 'SALES' ORDER BY 1,2;
 table_name | partition_name | subpartition_name
-----+-----+-----
SALES      | ASIA          | SYS0103
SALES      | ASIA          | SYS0104
SALES      | EUROPE        | SYS0101
SALES      | EUROPE        | SYS0102
(4 rows)
```

次のコマンドは、新しいパーティション americas を sales テーブルに追加し、パーティションの説明で指定された数のサブパーティションを作成します。

```
ALTER TABLE sales ADD PARTITION americas
  * VALUES ('US', 'CANADA');
```

コマンドを呼び出した後、テーブルにはパーティション americas と2つの新しく追加されたサブパーティションが含まれます。

```
edb=# SELECT table_name, partition_name, subpartition_name FROM
ALL_TAB_SUBPARTITIONS WHERE table_name = 'SALES' ORDER BY 1,2;
 table_name | partition_name | subpartition_name
-----+-----+-----
SALES      | AMERICAS      | SYS0107
SALES      | AMERICAS      | SYS0108
SALES      | ASIA          | SYS0103
SALES      | ASIA          | SYS0104
SALES      | EUROPE        | SYS0101
SALES      | EUROPE        | SYS0102
(6 rows)
```

例-SUBPARTITIONS num でパーティションを追加…

次の例では、3つのサブパーティションで構成されるリストパーティションテーブル `sales` にパーティションを追加します。次のコマンドで `sales` テーブルが作成されました。

```
CREATE TABLE sales
(
  * dept_no      number,
  * part_no      varchar2,
  * country      varchar2(20),
  * date         date,
  * amount       number
)
PARTITION BY LIST (country) SUBPARTITION BY HASH (part_no) SUBPARTITIONS 3
(
  * PARTITION europe VALUES ('FRANCE', 'ITALY'),
  * PARTITION asia  VALUES ('INDIA', 'PAKISTAN')
);
```

テーブルには、`europe` と `asia` のパーティションが含まれ、それぞれに3つのサブパーティションが含まれます。

```
edb=# SELECT table_name, partition_name, subpartition_name FROM
ALL_TAB_SUBPARTITIONS WHERE table_name = 'SALES' ORDER BY 1,2;
 table_name | partition_name | subpartition_name
-----+-----+-----
SALES      | ASIA          | SYS0104
SALES      | ASIA          | SYS0105
SALES      | ASIA          | SYS0106
SALES      | EUROPE        | SYS0101
SALES      | EUROPE        | SYS0102
SALES      | EUROPE        | SYS0103
(6 rows)
```

次のコマンドは、ADD PARTITION 句で指定された新しいパーティション `americas` と 5 つのサブパーティションを追加します。

```
ALTER TABLE sales ADD PARTITION americas
  * VALUES ('US', 'CANADA') SUBPARTITIONS 5;
```

コマンド `sales` を呼び出すと、テーブル `americas` と 5 つの新しく追加されたサブパーティションが含まれます。

```
edb=# SELECT table_name, partition_name, subpartition_name FROM
ALL_TAB_SUBPARTITIONS WHERE table_name = 'SALES' and partition_name =
'AMERICAS' ORDER BY 1,2;
 table_name | partition_name | subpartition_name
-----+-----+-----
SALES      | AMERICAS      | SYS0109
SALES      | AMERICAS      | SYS0110
SALES      | AMERICAS      | SYS0111
SALES      | AMERICAS      | SYS0112
SALES      | AMERICAS      | SYS0113
(5 rows)
```

例-SUBPARTITIONS num …を含むパーティションの追加… STORE IN

次の例では、3 つのサブパーティションで構成されるリストパーティションテーブル `sales` にパーティションを追加します。次のコマンドを使用してテーブルが作成されました。

```
CREATE TABLE sales
(
  * dept_no      number,
  * part_no      varchar2,
  * country      varchar2(20),
  * date         date,
  * amount       number
)
PARTITION BY LIST (country) SUBPARTITION BY HASH (part_no) SUBPARTITIONS 3
(
  * PARTITION europe VALUES('FRANCE', 'ITALY'),
  * PARTITION asia VALUES('INDIA', 'PAKISTAN'),
  * PARTITION americas VALUES('US', 'CANADA')
);
```

テーブルには、(`americas`, `asia` と `europe`) の 3 つのパーティションが含まれ、それぞれにシステム生成の名前を持つ 3 つのサブパーティションが含まれます。

```
edb=# SELECT table_name, partition_name, subpartition_name FROM
ALL_TAB_SUBPARTITIONS WHERE table_name = 'SALES' ORDER BY 1,2;
```

(次のページに続く)

(前のページからの続き)

table_name	partition_name	subpartition_name
SALES	AMERICAS	SYS0109
SALES	AMERICAS	SYS0107
SALES	AMERICAS	SYS0108
SALES	ASIA	SYS0105
SALES	ASIA	SYS0104
SALES	ASIA	SYS0106
SALES	EUROPE	SYS0101
SALES	EUROPE	SYS0103
SALES	EUROPE	SYS0102

(9 rows)

次のコマンドは、ADD PARTITION 句で指定された 5 つのサブパーティションを持つ新しいパーティション east_asia を追加し、(ts1) 名前付けのテーブルスペースに保存します。

```
ALTER TABLE sales ADD PARTITION east_asia
* VALUES ('CHINA', 'KOREA') SUBPARTITIONS 5 STORE IN (ts1);
```

コマンドテーブルを呼び出すと、パーティション east_asia と、テーブルスペース (ts1) に格納された 5 つの新しく追加されたサブパーティションが含まれます。

```
edb=# SELECT table_name, partition_name, subpartition_name, tablespace_name
FROM ALL_TAB_SUBPARTITIONS WHERE table_name = 'SALES' and partition_name =
'EAST_ASIA' ORDER BY 1,2;
```

table_name	partition_name	subpartition_name	tablespace_name
SALES	EAST_ASIA	SYS0113	TS1
SALES	EAST_ASIA	SYS0114	TS1
SALES	EAST_ASIA	SYS0115	TS1
SALES	EAST_ASIA	SYS0116	TS1
SALES	EAST_ASIA	SYS0117	TS1

(5 rows)

3.3 ALTER TABLE…ADD SUBPARTITION

</div>

ALTER TABLE… ADD SUBPARTITION コマンドは、既存のサブパーティションにサブパーティションを追加します。構文は次のとおりです。

```
ALTER TABLE <table_name> MODIFY PARTITION <partition_name>
* ADD SUBPARTITION <subpartition_definition>;
```

(次のページに続く)

Where subpartition_definition is:

```
{<list_subpartition> | <range_subpartition>}
```

and list_subpartition is:

```
*      SUBPARTITION [<subpartition_name>]
*      VALUES (<value>[, <value>]...)
*      [TABLESPACE <tablespace_name>]
```

and range_subpartition is:

```
*      SUBPARTITION [<subpartition_name>]
*      VALUES LESS THAN (<value>[, <value>]...)
*      [TABLESPACE <tablespace_name>]
```

説明

ALTER TABLE… ADD SUBPARTITION コマンドは、既存のパーティションにサブパーティションを追加します。パーティションはすでにサブパーティション化されている必要があります。定義されたサブパーティションの数に上限はありません。

新しいサブパーティションは、既存のサブパーティションと同じタイプ (LIST, RANGE または HASH) でなければなりません。新しいサブパーティションルールは、既存のサブパーティションを定義するサブパーティションルールで指定された同じ列をリファレンスする必要があります。

テーブル内の既存の行と追加するサブパーティションの値の間に矛盾する値がない限り、ALTER TABLE… ADD SUBPARTITION ステートメントを使用して、DEFAULT ルールを使用してテーブルにサブパーティションを追加できます。

ALTER TABLE… ADD SUBPARTITION ステートメントを使用して、MAXVALUE ルールを使用してテーブルにサブパーティションを追加することはできません。

ALTER TABLE… SPLIT SUBPARTITION ステートメントを使用して既存のサブパーティションを分割し、効果的にテーブルにサブパーティションを追加できます。

レンジのサブパーティション化されたテーブルで、既存のサブパーティションの前に新しいサブパーティションを追加することはできません。レンジのサブパーティションはオーダーで指定する必要があります。

TABLESPACE 句を含めて、サブパーティションが存在するテーブルスペースを指定します。テーブルスペースを指定しない場合、デフォルトテーブルスペースにサブパーティションが作成されます。

テーブルにインデックスが付けられている場合、インデックスは新しいサブパーティションに作成されます。

ALTER TABLE… ADD SUBPARTITION コマンドを使用するには、テーブルの所有者であるか、スーパーユーザ（または管理）権限が必要です。

パラメータ

table_name

サブパーティションが存在するパーティション表の名前（オプションでスキーマ修飾）。

partition_name

新しいサブパーティションが存在するパーティションの名前。

subpartition_name

作成するサブパーティションの名前。サブパーティション名は、すべてのパーティションとサブパーティション間で一意であり、オブジェクト識別子の命名規則に従う必要があります。

(value[, value]...)

value を使用して、引用されたりテラル値（またはリテラル値のコンマ区切りリスト）を指定します。これにより、テーブルエントリがパーティションにグループ化されます。各パーティショニングルールは少なくとも 1 つの値を指定する必要がありますが、ルール内で指定される値の数に制限はありません。value は、NULL、DEFAULT（LIST パーティションを指定する場合）、または MAXVALUE（RANGE パーティションを指定する場合）にもなります。

DEFAULT または MAXVALUE パーティションの作成については、[Handling Stray Values in a LIST or RANGE Partitioned Table](#) を参照してください。

tablespace_name

サブパーティションが存在するテーブルスペースの名前。

3.3.1 Example - Adding a Subpartition to a LIST/RANGE Partitioned Table

</div>

次の例では、RANGE サブパーティションをリストパーティション化された sales テーブルに追加します。次のコマンドで sales テーブルが作成されました。

```
CREATE TABLE sales
(
  * dept_no      number,
  * part_no      varchar2,
  * country      varchar2(20),
  * date         date,
  * amount       number
)
PARTITION BY LIST(country)
  * SUBPARTITION BY RANGE(date)
(
  * PARTITION europe VALUES('FRANCE', 'ITALY')
  * (
  *   SUBPARTITION europe_2011
  *     VALUES LESS THAN('2012-Jan-01'),
  *   SUBPARTITION europe_2012
  *     VALUES LESS THAN('2013-Jan-01')
  * ),
  * PARTITION asia VALUES('INDIA', 'PAKISTAN')
  * (
```

(次のページに続く)

(前のページからの続き)

```

*      SUBPARTITION asia_2011
*      VALUES LESS THAN('2012-Jan-01'),
*      SUBPARTITION asia_2012
*      VALUES LESS THAN('2013-Jan-01')
*    ),
* PARTITION americas VALUES('US', 'CANADA')
*    (
*      SUBPARTITION americas_2011
*      VALUES LESS THAN('2012-Jan-01'),
*      SUBPARTITION americas_2012
*      VALUES LESS THAN('2013-Jan-01')
*    )
);

```

sales テーブルには、europe, asia と americas 名前付けの 3 つのパーティションがあります。各パーティションには、範囲が定義された 2 つのサブパーティションがあります。

```

edb=# SELECT partition_name, subpartition_name, high_value FROM ALL_TAB_SUBPARTITIONS;
partition_name | subpartition_name |      high_value
-----+-----+-----
EUROPE         | EUROPE_2011       | '01-JAN-12 00:00:00'
EUROPE         | EUROPE_2012       | '01-JAN-13 00:00:00'
ASIA           | ASIA_2011         | '01-JAN-12 00:00:00'
ASIA           | ASIA_2012         | '01-JAN-13 00:00:00'
AMERICAS       | AMERICAS_2011     | '01-JAN-12 00:00:00'
AMERICAS       | AMERICAS_2012     | '01-JAN-13 00:00:00'
(6 rows)

```

次のコマンドは、europe_2013 名前付けのサブパーティションを追加します。

```

ALTER TABLE sales MODIFY PARTITION europe
* ADD SUBPARTITION europe_2013
* VALUES LESS THAN('2015-Jan-01');

```

コマンドを呼び出した後、テーブルには europe_2013 名前付けのサブパーティションが含まれます。

```

edb=# SELECT partition_name, subpartition_name, high_value FROM ALL_TAB_SUBPARTITIONS;
partition_name | subpartition_name |      high_value
-----+-----+-----
EUROPE         | EUROPE_2011       | '01-JAN-12 00:00:00'
EUROPE         | EUROPE_2012       | '01-JAN-13 00:00:00'
EUROPE         | EUROPE_2013       | '01-JAN-15 00:00:00'
ASIA           | ASIA_2011         | '01-JAN-12 00:00:00'
ASIA           | ASIA_2012         | '01-JAN-13 00:00:00'
AMERICAS       | AMERICAS_2011     | '01-JAN-12 00:00:00'

```

(次のページに続く)

(前のページからの続き)

AMERICAS	AMERICAS_2012	'01-JAN-13 00:00:00'
----------	---------------	----------------------

(7 rows)

!!! Note * 新しいレンジサブパーティションを追加する場合、サブパーティションルールは、既存のサブパーティションの後に該当するレンジを指定する必要があります。

3.3.2 Example - Adding a Subpartition to a RANGE/LIST Partitioned Table

</ div>

次の例では、LIST サブパーティションを RANGE パーティション sales テーブルに追加します。次のコマンドで sales テーブルが作成されました。

```
CREATE TABLE sales
(
  * dept_no      number,
  * part_no      varchar2,
  * country      varchar2(20),
  * date         date,
  * amount       number
)
PARTITION BY RANGE(date)
  * SUBPARTITION BY LIST (country)
  * (
  *   PARTITION first_half_2012 VALUES LESS THAN('01-JUL-2012')
  *   (
  *     SUBPARTITION europe VALUES ('ITALY', 'FRANCE'),
  *     SUBPARTITION americas VALUES ('US', 'CANADA')
  *   ),
  *   PARTITION second_half_2012 VALUES LESS THAN('01-JAN-2013')
  *   (
  *     SUBPARTITION asia VALUES ('INDIA', 'PAKISTAN')
  *   )
  * );
```

上記のコマンドを実行すると、sales テーブルには、first_half_2012 と second_half_2012 名前付け 2 つのパーティションが作成されます。first_half_2012 partition には europe および americas 名前付けの 2 つのサブパーティションがあり、second_half_2012 パーティションには asia 名前付けの 1 つのパーティションがあります。

```
edb=# SELECT partition_name, subpartition_name, high_value FROM ALL_TAB_SUBPARTITIONS;
  * partition_name | subpartition_name |      high_value
-----+-----+-----
SECOND_HALF_2012 | ASIA              | 'INDIA', 'PAKISTAN'
FIRST_HALF_2012  | AMERICAS          | 'US', 'CANADA'
```

(次のページに続く)

(前のページからの続き)

```
FIRST_HALF_2012 | EUROPE | 'ITALY', 'FRANCE'
(3 rows)
```

次のコマンドは、east_asia 名前付けのサブパーティションを second_half_2012 パーティションに追加します。

```
ALTER TABLE sales MODIFY PARTITION second_half_2012
* ADD SUBPARTITION east_asia VALUES ('CHINA');
```

コマンドを呼び出した後、テーブルには east_asia 名前付けのサブパーティションが含まれます。

```
edb=# SELECT partition_name, subpartition_name, high_value FROM ALL_TAB_SUBPARTITIONS;
      partition_name | subpartition_name | high_value
-----+-----+-----
SECOND_HALF_2012 | ASIA | 'INDIA', 'PAKISTAN'
SECOND_HALF_2012 | EAST_ASIA | 'CHINA'
FIRST_HALF_2012 | AMERICAS | 'US', 'CANADA'
FIRST_HALF_2012 | EUROPE | 'ITALY', 'FRANCE'
(4 rows)
```

3.4 ALTER TABLE…SPLIT PARTITION

</ div>

ALTER TABLE… SPLIT PARTITION コマンドを使用して、単一のパーティションを2つのパーティションに分割し、新しく作成されたパーティションでオリジナルのテーブルのパーティショニングを維持し、パーティションの内容を新しいパーティション間で再配布します。コマンド構文には2つの形式があります。

最初のフォームは、RANGE パーティションを2つのパーティションに分割します。

```
ALTER TABLE <table_name> SPLIT PARTITION <partition_name>
* AT (<range_part_value>)
* INTO
* (
*   PARTITION <new_part1>
*   [TABLESPACE <tablespace_name>]
*   [SUBPARTITIONS <num>] [STORE IN ( <tablespace_name> [, <tablespace_name>]...
→) ],
*   PARTITION <new_part2>
*   [TABLESPACE <tablespace_name>]
*   [SUBPARTITIONS <num>] [STORE IN ( <tablespace_name> [, <tablespace_name>]...
→) ]
* );
```

2 番目のフォームは、LIST パーティションを2つのパーティションに分割します。

```

ALTER TABLE <table_name> SPLIT PARTITION <partition_name>
* VALUES (<value>[, <value>]...)
* INTO
* (
*   PARTITION <new_part1>
*   [TABLESPACE <tablespace_name>]
*   [SUBPARTITIONS <num>] [STORE IN ( <tablespace_name> [, <tablespace_name>]...
→) ],
*   PARTITION <new_part2>
*   [TABLESPACE <tablespace_name>]
*   [SUBPARTITIONS <num>] [STORE IN ( <tablespace_name> [, <tablespace_name>]...
→) ]
* );

```

説明

ALTER TABLE...SPLIT PARTITION コマンドは、既存の LIST または RANGE パーティションテーブルにパーティションを追加します。ALTER TABLE... SPLIT PARTITION コマンドでは、パーティションを HASH パーティションテーブルに追加できないことにノートしてください。テーブルが持つパーティションの数に上限はありません。

ALTER TABLE...SPLIT PARTITION コマンドを実行すると、Advanced Server は2つの新しいパーティションを作成し、新しく作成されたパーティションでオリジナルのテーブルのパーティショニングを維持し、それらの間で古いパーティションのコンテンツを再分配します（パーティショニングルールの制約に従って）。

TABLESPACE 句を含めて、パーティションが存在するテーブルスペースを指定します。テーブルスペースを指定しない場合、パーティションはオリジナルのパーティション表のテーブルスペースに存在します。

STORE IN 句を使用して、自動生成されたサブパーティションが格納されるテーブルスペースリストを指定します。

テーブルにインデックスが付けられている場合、インデックスは新しいパーティションに作成されます。

ALTER TABLE... SPLIT PARTITION コマンドを使用するには、テーブルの所有者であるか、スーパーユーザ（または管理）権限が必要です。

パラメータ

table_name

パーティション化されたテーブルの名前（オプションでスキーマ修飾）。

partition_name

分割されるパーティションの名前。

new_part1

作成される最初の新しいパーティションの名前。パーティション名は、すべてのパーティションとサブパーティション間で一意であり、オブジェクト識別子の命名規則に従う必要があります。

new_part1 は、ALTER TABLE... SPLIT PARTITION コマンドで指定されたパーティショニングコンストレインを満たす行を受け取ります。

new_part2

作成する 2 番目の新しいパーティションの名前。パーティション名は、すべてのパーティションとサブパーティション間で一意であり、オブジェクト識別子の命名規則に従う必要があります。

new_part2 は、ALTER TABLE... SPLIT PARTITION コマンドで指定されたパーティショニングコンストレーンによって行が **new_part1** に送信されないことを受け取ります。

range_part_value

range_part_value を使用して、新しいパーティションを作成する境界ルールを指定します。パーティショニングルールには、2 つの演算子（つまり、演算子以上、および演算子未満）を持つデータタイプの列が少なくとも 1 つ含まれている必要があります。範囲の境界は、LESS THAN 句に対して評価され、包括的ではありません。2010 年 1 月 1 日の日付境界には、2009 年 12 月 31 日以前の日付値のみが含まれます。

(value[, value]...)

value を使用して、引用されたリテラル値（またはリテラル値のコンマ区切りリスト）を指定して、行をパーティションに分散します。各パーティショニングルールは少なくとも 1 つの値を指定する必要がありますが、ルール内で指定される値の数に制限はありません。

DEFAULT または MAXVALUE パーティションの作成については、[Handling Stray Values in a LIST or RANGE Partitioned Table](#) を参照してください。

num

SUBPARTITIONS **num** 句は、HASH サブパーティションでのみサポートされます。句を使用して、ハッシュサブパーティションの数を指定します。または、サブパーティション定義を使用して、個々のサブパーティションとその表領域を指定できます。SUBPARTITIONS または SUBPARTITION DEFINITION が指定されていない場合、パーティションは SUBPARTITION TEMPLATE に基づいてサブパーティションを作成します。

tablespace_name

パーティションまたはサブパーティションが存在するテーブルスペースの名前。

3.4.1 Example - Splitting a LIST Partition

</ div>

この例では、list-partitioned **sales** テーブルのパーティションの 1 つを 2 つの新しいパーティションに分割し、パーティションの内容をそれらの間で再配布します。次のステートメントで **sales** テーブルが作成されます。

```
CREATE TABLE sales
(
  * dept_no      number,
  * part_no      varchar2,
  * country      varchar2(20),
  * date         date,
  * amount       number
)
```

(次のページに続く)

(前のページからの続き)

```
PARTITION BY LIST(country)
(
  * PARTITION europe VALUES('FRANCE', 'ITALY'),
  * PARTITION asia VALUES('INDIA', 'PAKISTAN'),
  * PARTITION americas VALUES('US', 'CANADA')
);
```

テーブル定義は3つのパーティション (europe, asia、および americas) を作成します。次のコマンドは、各パーティションに行を追加します。

```
INSERT INTO sales VALUES
  * (10, '4519b', 'FRANCE', '17-Jan-2012', '45000'),
  * (20, '3788a', 'INDIA', '01-Mar-2012', '75000'),
  * (40, '9519b', 'US', '12-Apr-2012', '145000'),
  * (20, '3788a', 'PAKISTAN', '04-Jun-2012', '37500'),
  * (40, '4577b', 'US', '11-Nov-2012', '25000'),
  * (30, '7588b', 'CANADA', '14-Dec-2012', '50000'),
  * (30, '9519b', 'CANADA', '01-Feb-2012', '75000'),
  * (30, '4519b', 'CANADA', '08-Apr-2012', '120000'),
  * (40, '3788a', 'US', '12-May-2012', '4950'),
  * (10, '9519b', 'ITALY', '07-Jul-2012', '15000'),
  * (10, '9519a', 'FRANCE', '18-Aug-2012', '650000'),
  * (10, '9519b', 'FRANCE', '18-Aug-2012', '650000'),
  * (20, '3788b', 'INDIA', '21-Sept-2012', '5090'),
  * (40, '4788a', 'US', '23-Sept-2012', '4950'),
  * (40, '4788b', 'US', '09-Oct-2012', '15000'),
  * (20, '4519a', 'INDIA', '18-Oct-2012', '650000'),
  * (20, '4519b', 'INDIA', '2-Dec-2012', '5090');
```

行はパーティションに分散されます。

```
edb=# SELECT tableoid::regclass, * FROM sales;
   *   tableoid   | dept_no | part_no | country |          date          | amount
-----+-----+-----+-----+-----+-----+-----
sales_americas |      40 | 9519b  | US      | 12-APR-12 00:00:00 | 145000
sales_americas |      40 | 4577b  | US      | 11-NOV-12 00:00:00 | 25000
sales_americas |      30 | 7588b  | CANADA  | 14-DEC-12 00:00:00 | 50000
sales_americas |      30 | 9519b  | CANADA  | 01-FEB-12 00:00:00 | 75000
sales_americas |      30 | 4519b  | CANADA  | 08-APR-12 00:00:00 | 120000
sales_americas |      40 | 3788a  | US      | 12-MAY-12 00:00:00 | 4950
sales_americas |      40 | 4788a  | US      | 23-SEP-12 00:00:00 | 4950
sales_americas |      40 | 4788b  | US      | 09-OCT-12 00:00:00 | 15000
sales_europe   |      10 | 4519b  | FRANCE  | 17-JAN-12 00:00:00 | 45000
sales_europe   |      10 | 9519b  | ITALY   | 07-JUL-12 00:00:00 | 15000
sales_europe   |      10 | 9519a  | FRANCE  | 18-AUG-12 00:00:00 | 650000
```

(次のページに続く)

(前のページからの続き)

sales_europe		10		9519b		FRANCE		18-AUG-12 00:00:00		650000
sales_asia		20		3788a		INDIA		01-MAR-12 00:00:00		75000
sales_asia		20		3788a		PAKISTAN		04-JUN-12 00:00:00		37500
sales_asia		20		3788b		INDIA		21-SEP-12 00:00:00		5090
sales_asia		20		4519a		INDIA		18-OCT-12 00:00:00		650000
sales_asia		20		4519b		INDIA		02-DEC-12 00:00:00		5090

(17 rows)

次のコマンドは、americas パーティションを us と canada 名前付け 2 つのパーティションに分割します。

```
ALTER TABLE sales SPLIT PARTITION americas
* VALUES ('US')
* INTO (PARTITION us, PARTITION canada);
```

SELECT ステートメントは、行が再配布されたことを確認します。

```
edb=# SELECT tableoid::regclass, * FROM sales;
* tableoid | dept_no | part_no | country | date | amount
-----+-----+-----+-----+-----+-----
sales_canada | 30 | 7588b | CANADA | 14-DEC-12 00:00:00 | 50000
sales_canada | 30 | 9519b | CANADA | 01-FEB-12 00:00:00 | 75000
sales_canada | 30 | 4519b | CANADA | 08-APR-12 00:00:00 | 120000
sales_europe | 10 | 4519b | FRANCE | 17-JAN-12 00:00:00 | 45000
sales_europe | 10 | 9519b | ITALY | 07-JUL-12 00:00:00 | 15000
sales_europe | 10 | 9519a | FRANCE | 18-AUG-12 00:00:00 | 650000
sales_europe | 10 | 9519b | FRANCE | 18-AUG-12 00:00:00 | 650000
sales_asia | 20 | 3788a | INDIA | 01-MAR-12 00:00:00 | 75000
sales_asia | 20 | 3788a | PAKISTAN | 04-JUN-12 00:00:00 | 37500
sales_asia | 20 | 3788b | INDIA | 21-SEP-12 00:00:00 | 5090
sales_asia | 20 | 4519a | INDIA | 18-OCT-12 00:00:00 | 650000
sales_asia | 20 | 4519b | INDIA | 02-DEC-12 00:00:00 | 5090
sales_us | 40 | 9519b | US | 12-APR-12 00:00:00 | 145000
sales_us | 40 | 4577b | US | 11-NOV-12 00:00:00 | 25000
sales_us | 40 | 3788a | US | 12-MAY-12 00:00:00 | 4950
sales_us | 40 | 4788a | US | 23-SEP-12 00:00:00 | 4950
sales_us | 40 | 4788b | US | 09-OCT-12 00:00:00 | 15000
(17 rows)
```

3.4.2 Example - Splitting a RANGE Partition

</ div>

この例では、（範囲でパーティション化された sales テーブルの） q4_2012 パーティションを 2 つのパーティションに分割し、パーティションのコンテンツを再配布します。次のコマンドを使用して、sales テーブルを作成します。

```
CREATE TABLE sales
(
  * dept_no      number,
  * part_no      varchar2,
  * country      varchar2(20),
  * date         date,
  * amount       number
)
PARTITION BY RANGE(date)
(
  * PARTITION q1_2012
  *   VALUES LESS THAN('2012-Apr-01'),
  * PARTITION q2_2012
  *   VALUES LESS THAN('2012-Jul-01'),
  * PARTITION q3_2012
  *   VALUES LESS THAN('2012-Oct-01'),
  * PARTITION q4_2012
  *   VALUES LESS THAN('2013-Jan-01')
);
```

テーブル定義により、4 つのパーティション（q1_2012、q2_2012、q3_2012、および q4_2012）が作成されます。次のコマンドは、各パーティションに行を追加します。

```
INSERT INTO sales VALUES
  * (10, '4519b', 'FRANCE', '17-Jan-2012', '45000'),
  * (20, '3788a', 'INDIA', '01-Mar-2012', '75000'),
  * (40, '9519b', 'US', '12-Apr-2012', '145000'),
  * (20, '3788a', 'PAKISTAN', '04-Jun-2012', '37500'),
  * (40, '4577b', 'US', '11-Nov-2012', '25000'),
  * (30, '7588b', 'CANADA', '14-Dec-2012', '50000'),
  * (30, '9519b', 'CANADA', '01-Feb-2012', '75000'),
  * (30, '4519b', 'CANADA', '08-Apr-2012', '120000'),
  * (40, '3788a', 'US', '12-May-2012', '4950'),
  * (10, '9519b', 'ITALY', '07-Jul-2012', '15000'),
  * (10, '9519a', 'FRANCE', '18-Aug-2012', '650000'),
  * (10, '9519b', 'FRANCE', '18-Aug-2012', '650000'),
  * (20, '3788b', 'INDIA', '21-Sept-2012', '5090'),
  * (40, '4788a', 'US', '23-Sept-2012', '4950'),
  * (40, '4788b', 'US', '09-Oct-2012', '15000'),
```

(次のページに続く)

(前のページからの続き)

```
* (20, '4519a', 'INDIA', '18-Oct-2012', '650000'),
* (20, '4519b', 'INDIA', '2-Dec-2012', '5090');
```

SELECT ステートメントは、行が期待どおりにパーティションに分散されていることを確認します。

```
edb=# SELECT tableoid::regclass, * FROM sales;
   * tableoid   | dept_no | part_no | country |          date          | amount
-----+-----+-----+-----+-----+-----
sales_q1_2012 |      10 | 4519b   | FRANCE  | 17-JAN-12 00:00:00 | 45000
sales_q1_2012 |      20 | 3788a   | INDIA   | 01-MAR-12 00:00:00 | 75000
sales_q1_2012 |      30 | 9519b   | CANADA  | 01-FEB-12 00:00:00 | 75000
sales_q2_2012 |      40 | 9519b   | US      | 12-APR-12 00:00:00 | 145000
sales_q2_2012 |      20 | 3788a   | PAKISTAN | 04-JUN-12 00:00:00 | 37500
sales_q2_2012 |      30 | 4519b   | CANADA  | 08-APR-12 00:00:00 | 120000
sales_q2_2012 |      40 | 3788a   | US      | 12-MAY-12 00:00:00 | 4950
sales_q3_2012 |      10 | 9519b   | ITALY   | 07-JUL-12 00:00:00 | 15000
sales_q3_2012 |      10 | 9519a   | FRANCE  | 18-AUG-12 00:00:00 | 650000
sales_q3_2012 |      10 | 9519b   | FRANCE  | 18-AUG-12 00:00:00 | 650000
sales_q3_2012 |      20 | 3788b   | INDIA   | 21-SEP-12 00:00:00 | 5090
sales_q3_2012 |      40 | 4788a   | US      | 23-SEP-12 00:00:00 | 4950
sales_q4_2012 |      40 | 4577b   | US      | 11-NOV-12 00:00:00 | 25000
sales_q4_2012 |      30 | 7588b   | CANADA  | 14-DEC-12 00:00:00 | 50000
sales_q4_2012 |      40 | 4788b   | US      | 09-OCT-12 00:00:00 | 15000
sales_q4_2012 |      20 | 4519a   | INDIA   | 18-OCT-12 00:00:00 | 650000
sales_q4_2012 |      20 | 4519b   | INDIA   | 02-DEC-12 00:00:00 | 5090
(17 rows)
```

次のコマンドは、q4_2012 パーティションを q4_2012_p1 と q4_2012_p2 名前付け 2 つのパーティションに分割します。

```
ALTER TABLE sales SPLIT PARTITION q4_2012
* AT ('15-Nov-2012')
* INTO
* (
*   PARTITION q4_2012_p1,
*   PARTITION q4_2012_p2
* );
```

SELECT ステートメントは、新しいパーティションに行が再分散されたことを確認します。

```
edb=# SELECT tableoid::regclass, * FROM sales;
   * tableoid   | dept_no | part_no | country |          date          | amount
-----+-----+-----+-----+-----+-----
sales_q1_2012 |      10 | 4519b   | FRANCE  | 17-JAN-12 00:00:00 | 45000
sales_q1_2012 |      20 | 3788a   | INDIA   | 01-MAR-12 00:00:00 | 75000
```

(次のページに続く)

(前のページからの続き)

sales_q1_2012		30		9519b		CANADA		01-FEB-12 00:00:00		75000
sales_q2_2012		40		9519b		US		12-APR-12 00:00:00		145000
sales_q2_2012		20		3788a		PAKISTAN		04-JUN-12 00:00:00		37500
sales_q2_2012		30		4519b		CANADA		08-APR-12 00:00:00		120000
sales_q2_2012		40		3788a		US		12-MAY-12 00:00:00		4950
sales_q3_2012		10		9519b		ITALY		07-JUL-12 00:00:00		15000
sales_q3_2012		10		9519a		FRANCE		18-AUG-12 00:00:00		650000
sales_q3_2012		10		9519b		FRANCE		18-AUG-12 00:00:00		650000
sales_q3_2012		20		3788b		INDIA		21-SEP-12 00:00:00		5090
sales_q3_2012		40		4788a		US		23-SEP-12 00:00:00		4950
sales_q4_2012_p1		40		4577b		US		11-NOV-12 00:00:00		25000
sales_q4_2012_p1		40		4788b		US		09-OCT-12 00:00:00		15000
sales_q4_2012_p1		20		4519a		INDIA		18-OCT-12 00:00:00		650000
sales_q4_2012_p2		30		7588b		CANADA		14-DEC-12 00:00:00		50000
sales_q4_2012_p2		20		4519b		INDIA		02-DEC-12 00:00:00		5090

(17 rows)

3.4.3 Example - Splitting a RANGE/LIST Partition

</ div>

次の例では、範囲パーティション sales テーブルのパーティションの 1 つを新しいパーティションに分割し、新しく作成されたパーティションでオリジナルのテーブルのパーティショニングを維持し、パーティション間でパーティションの内容を再分散します。

```
CREATE TABLE sales
(
  * dept_no      number,
  * part_no      varchar2,
  * country      varchar2(20),
  * date         date,
  * amount       number
)
PARTITION BY RANGE(date)
  * SUBPARTITION BY LIST (country)
  * (
  *   PARTITION q1_2012 VALUES LESS THAN('2012-Apr-01')
  *   (
  *     SUBPARTITION europe VALUES('FRANCE', 'ITALY'),
  *     SUBPARTITION americas VALUES('US', 'CANADA'),
  *     SUBPARTITION asia VALUES('INDIA', 'PAKISTAN')
  *   )
  * );
```

sales テーブルには、パーティション q1_2012 と 3 つのサブパーティション europe, americas および asia が含まれます。

```
edb=# SELECT table_name, partition_name, subpartition_name, high_value FROM
ALL_TAB_SUBPARTITIONS WHERE table_name = 'SALES' ORDER BY 1,2;
 table_name | partition_name | subpartition_name |      high_value
-----+-----+-----+-----
SALES      | Q1_2012       | EUROPE            | 'FRANCE', 'ITALY'
SALES      | Q1_2012       | AMERICAS          | 'US', 'CANADA'
SALES      | Q1_2012       | ASIA              | 'INDIA', 'PAKISTAN'
(3 rows)
```

次のコマンドは、q1_2012 パーティションを q1_2012_p1 と q1_2012_p2 名前付け 2 つのパーティションに分割します。

```
ALTER TABLE sales SPLIT PARTITION q1_2012
* AT ('01-Mar-2012')
* INTO
* (
*   PARTITION q1_2012_p1,
*   PARTITION q1_2012_p2
* );
```

SELECT ステートメントは、システムが生成した名前で、新しく作成されたパーティション q1_2012_p1 と q1_2012_p2 に同じ数のサブパーティションが作成されたことを確認します。

```
edb=# SELECT table_name, partition_name, subpartition_name, high_value FROM
ALL_TAB_SUBPARTITIONS WHERE table_name = 'SALES' ORDER BY 1,2,3;
 table_name | partition_name | subpartition_name |      high_value
-----+-----+-----+-----
SALES      | Q1_2012_P1    | SYS0105           | 'US', 'CANADA'
SALES      | Q1_2012_P1    | SYS0106           | 'FRANCE', 'ITALY'
SALES      | Q1_2012_P1    | SYS0107           | 'INDIA', 'PAKISTAN'
SALES      | Q1_2012_P2    | SYS0108           | 'US', 'CANADA'
SALES      | Q1_2012_P2    | SYS0109           | 'FRANCE', 'ITALY'
SALES      | Q1_2012_P2    | SYS0110           | 'INDIA', 'PAKISTAN'
(6 rows)
```

例-SUBPARTITIONS num でパーティションを分割…

次の例では、リストパーティション `sales` テーブルのパーティションの1つを新しいパーティションに分割し、新しく作成されたパーティションでオリジナルのテーブルのパーティショニングを維持し、パーティションの内容をそれらの間で再分配します。SUBPARTITIONS 句を使用すると、指定した数のサブパーティションを追加できます。SUBPARTITIONS 句が指定されていない場合、新しいパーティションはデフォルトのサブパーティション数を継承します。次のステートメントで `sales` テーブルが作成されます。

```
CREATE TABLE sales
(
  * dept_no      number,
  * part_no      varchar2,
  * country      varchar2(20),
  * date         date,
  * amount       number
)
PARTITION BY LIST(country) SUBPARTITION BY HASH (part_no) SUBPARTITIONS 2
(
  * PARTITION europe VALUES('FRANCE', 'ITALY'),
  * PARTITION asia VALUES('INDIA', 'PAKISTAN'),
  * PARTITION americas VALUES('US', 'CANADA')
);
```

テーブル定義は、それぞれ2つのサブパーティションを含む3つのパーティション (`europe`, `asia`, および `americas`) を作成します。

```
edb=# SELECT table_name, partition_name, subpartition_name FROM
ALL_TAB_SUBPARTITIONS WHERE table_name = 'SALES' ORDER BY 1,2;
 table_name | partition_name | subpartition_name
-----+-----+-----
SALES      | AMERICAS      | SYS0105
SALES      | AMERICAS      | SYS0106
SALES      | ASIA          | SYS0103
SALES      | ASIA          | SYS0104
SALES      | EUROPE        | SYS0101
SALES      | EUROPE        | SYS0102
(6 rows)
```

次のコマンドは、`americas` パーティションを `partition_us` と `partition_canada` 名前付け2つのパーティションに分割します。

```
ALTER TABLE sales SPLIT PARTITION americas VALUES ('US') INTO (PARTITION
partition_us SUBPARTITIONS 5, PARTITION partition_canada);
```

SELECT ステートメントは、`americas` パーティションが `partition_us` と `partition_canada` に分割されていることを確認します。`partition_us` には5つのサブパーティションが含まれ、`partition_canada` にはデフォルトで2つのサブパーティションと割り当てられたシステム生成名が含まれます。

```
edb=# SELECT table_name, partition_name, subpartition_name FROM
ALL_TAB_SUBPARTITIONS WHERE table_name = 'SALES' ORDER BY 1,2,3;
 table_name | partition_name | subpartition_name
-----+-----+-----
SALES      | ASIA          | SYS0103
SALES      | ASIA          | SYS0104
SALES      | EUROPE        | SYS0101
SALES      | EUROPE        | SYS0102
SALES      | PARTITION_CANADA | SYS0115
SALES      | PARTITION_CANADA | SYS0116
SALES      | PARTITION_US   | SYS0110
SALES      | PARTITION_US   | SYS0111
SALES      | PARTITION_US   | SYS0112
SALES      | PARTITION_US   | SYS0113
SALES      | PARTITION_US   | SYS0114
(11 rows)
```

例-SUBPARTITIONS num … STORE IN でパーティションを分割する

次の例では、リストパーティション `sales` テーブルの `europa` パーティションを2つのパーティションに分割し、新しく作成されたパーティションでオリジナルのテーブルのパーティショニングを維持し、パーティションのコンテンツを再配布します。SUBPARTITIONS 句を使用すると、指定した数のサブパーティションを追加できます。次のコマンドを使用して、`sales` テーブルを作成します。

```
CREATE TABLE sales
(
  * dept_no      number,
  * part_no      varchar2,
  * country      varchar2(20),
  * date         date,
  * amount       number
)
PARTITION BY LIST(country) SUBPARTITION BY HASH (part_no) SUBPARTITIONS 4
(
  * PARTITION europe VALUES('FRANCE', 'ITALY'),
  * PARTITION asia VALUES('INDIA', 'PAKISTAN'),
  * PARTITION americas VALUES('US', 'CANADA')
);
```

`sales` テーブルには3つのパーティション (`europe`, `asia`, および `americas`) が含まれ、それぞれにシステム生成の名前が割り当てられた4つのサブパーティションが含まれます。

```
edb=# SELECT table_name, partition_name, subpartition_name FROM
ALL_TAB_SUBPARTITIONS WHERE table_name = 'SALES' ORDER BY 1,2;
```

(次のページに続く)

(前のページからの続き)

table_name	partition_name	subpartition_name
SALES	AMERICAS	SYS0112
SALES	AMERICAS	SYS0111
SALES	AMERICAS	SYS0109
SALES	AMERICAS	SYS0110
SALES	ASIA	SYS0107
SALES	ASIA	SYS0105
SALES	ASIA	SYS0106
SALES	ASIA	SYS0108
SALES	EUROPE	SYS0101
SALES	EUROPE	SYS0104
SALES	EUROPE	SYS0103
SALES	EUROPE	SYS0102

(12 rows)

次のコマンドは、europe パーティションを france と italy 名前付け 2 つのパーティションに分割します。

```
ALTER TABLE sales SPLIT PARTITION europe VALUES ('FRANCE') INTO (PARTITION
france SUBPARTITIONS 10 STORE IN (ts1), PARTITION italy);
```

SELECT ステートメントは、europe パーティションが 2 つのパーティションに分割されていることを確認します。パーティション france には、テーブルスペース ts1 に格納される 10 個のサブパーティションが含まれ、パーティション italy には、オリジナルのパーティション europe と同様に 4 つのサブパーティションが含まれます。

```
edb=# SELECT table_name, partition_name, subpartition_name, tablespace_name
FROM ALL_TAB_SUBPARTITIONS WHERE table_name = 'SALES' ORDER BY 1,2,3;
```

table_name	partition_name	subpartition_name	tablespace_name
SALES	AMERICAS	SYS0109	
SALES	AMERICAS	SYS0110	
SALES	AMERICAS	SYS0111	
SALES	AMERICAS	SYS0112	
SALES	ASIA	SYS0105	
SALES	ASIA	SYS0106	
SALES	ASIA	SYS0107	
SALES	ASIA	SYS0108	
SALES	FRANCE	SYS0116	TS1
SALES	FRANCE	SYS0117	TS1
SALES	FRANCE	SYS0118	TS1
SALES	FRANCE	SYS0119	TS1
SALES	FRANCE	SYS0120	TS1
SALES	FRANCE	SYS0121	TS1
SALES	FRANCE	SYS0122	TS1

(次のページに続く)

(前のページからの続き)

SALES		FRANCE		SYS0123		TS1
SALES		FRANCE		SYS0124		TS1
SALES		FRANCE		SYS0125		TS1
SALES		ITALY		SYS0126		
SALES		ITALY		SYS0127		
SALES		ITALY		SYS0128		
SALES		ITALY		SYS0129		
(22 rows)						

3.5 ALTER TABLE...SPLIT SUBPARTITION

</ div>

ALTER TABLE... SPLIT SUBPARTITION コマンドを使用して、単一のサブパーティションを2つのサブパーティションに分割し、サブパーティションのコンテンツを再配布します。コマンドには2つのバリエーションがあります。

最初のバリエーションは、レンジサブパーティションを2つのサブパーティションに分割します。

```
ALTER TABLE <table_name> SPLIT SUBPARTITION <subpartition_name>
* AT (range_part_value)
* INTO
* (
*   SUBPARTITION <new_subpart1>
*   [TABLESPACE <tablespace_name>],
*   SUBPARTITION <new_subpart2>
*   [TABLESPACE <tablespace_name>]
* );
```

2 番目のバリエーションは、リストサブパーティションを2つのサブパーティションに分割します。

```
ALTER TABLE <table_name> SPLIT SUBPARTITION <subpartition_name>
* VALUES (<value>[, <value>]...)
* INTO
* (
*   SUBPARTITION <new_subpart1>
*   [TABLESPACE <tablespace_name>],
*   SUBPARTITION <new_subpart2>
*   [TABLESPACE <tablespace_name>]
* );
```

説明

ALTER TABLE...SPLIT SUBPARTITION コマンドは、既存のサブパーティションテーブルにサブパーティションを追加します。定義されたサブパーティションの数に上限はありません。ALTER TABLE...SPLIT SUBPARTITION コマンドを実行すると、Advanced Server は2つの新しいサブパーティションを作成し、指

定されたサブパーティションルールによって制約されている値を含む行を **new_subpart1** に、残りの行を **new_subpart2** に移動します。

新しいサブパーティションルールは、既存のサブパーティションを定義するルールで指定された列をリファレンスする必要があります。

TABLESPACE 句を含めて、新しいサブパーティションが存在するテーブルスペースを指定します。テーブルスペースを指定しない場合、デフォルトテーブルスペースにサブパーティションが作成されます。

テーブルにインデックスが付けられている場合、インデックスは新しいサブパーティションに作成されます。

ALTER TABLE... SPLIT SUBPARTITION コマンドを使用するには、テーブルの所有者であるか、スーパーユーザ（または管理）権限が必要です。

パラメータ

table_name

パーティション化されたテーブルの名前（オプションでスキーマ修飾）。

subpartition_name

分割されるサブパーティションの名前。

new_subpart1

作成される最初の新しいサブパーティションの名前。サブパーティション名は、すべてのパーティションとサブパーティション間で一意であり、オブジェクト識別子の命名規則に従う必要があります。

new_subpart1 は、**ALTER TABLE... SPLIT SUBPARTITION** コマンドで指定されたサブパーティションのコンストレインを満たす行を受け取ります。

new_subpart2

作成する 2 番目の新しいサブパーティションの名前。サブパーティション名は、すべてのパーティションとサブパーティション間で一意であり、オブジェクト識別子の命名規則に従う必要があります。

new_subpart2 は、**ALTER TABLE... SPLIT SUBPARTITION** コマンドで指定されたサブパーティションコンストレインによって **new_subpart1** に送信されない行を受け取ります。

(value[, value]...)

value を使用して、引用されたりテラル値（またはリテラル値のコンマ区切りリスト）を指定します。これにより、テーブルエントリがパーティションにグループ化されます。各パーティショニングルールは少なくとも 1 つの値を指定する必要がありますが、ルール内で指定される値の数に制限はありません。**value** は、**NULL**、**DEFAULT**（**LIST** サブパーティションを指定する場合）、または **MAXVALUE**（**RANGE** サブパーティションを指定する場合）にもなります。

DEFAULT または **MAXVALUE** パーティションの作成については、[Handling Stray Values in a LIST or RANGE Partitioned Table](#) を参照してください。

tablespace_name

パーティションまたはサブパーティションが存在するテーブルスペースの名前。

3.5.1 Example - Splitting a LIST Subpartition

</div>

次の例では、リストサブパーティションを分割し、サブパーティションのコンテンツを2つの新しいサブパーティションに再分配します。サンプルテーブル (sales) は、次のコマンドで作成されました。

```
CREATE TABLE sales
(
  * dept_no      number,
  * part_no      varchar2,
  * country      varchar2(20),
  * date         date,
  * amount      number
)
PARTITION BY RANGE(date)
  * SUBPARTITION BY LIST (country)
  * (
  *   PARTITION first_half_2012 VALUES LESS THAN('01-JUL-2012')
  *   (
  *     SUBPARTITION p1_europe VALUES ('ITALY', 'FRANCE'),
  *     SUBPARTITION p1_americas VALUES ('US', 'CANADA')
  *   ),
  *   PARTITION second_half_2012 VALUES LESS THAN('01-JAN-2013')
  *   (
  *     SUBPARTITION p2_europe VALUES ('ITALY', 'FRANCE'),
  *     SUBPARTITION p2_americas VALUES ('US', 'CANADA')
  *   )
  * );
```

sales テーブルには、first_half_2012 と second_half_2012 名前付け2つのパーティションがあります。各パーティションには、パーティションのコンテンツを country 列の値に基づいてサブパーティションに分散する、範囲が定義された2つのサブパーティションがあります。

```
edb=# SELECT partition_name, subpartition_name, high_value FROM ALL_TAB_SUBPARTITIONS;
   * partition_name | subpartition_name |   high_value
-----+-----+-----
SECOND_HALF_2012 | P2_AMERICAS      | 'US', 'CANADA'
SECOND_HALF_2012 | P2_EUROPE        | 'ITALY', 'FRANCE'
FIRST_HALF_2012  | P1_AMERICAS      | 'US', 'CANADA'
FIRST_HALF_2012  | P1_EUROPE        | 'ITALY', 'FRANCE'
(4 rows)
```

次のコマンドは、各サブパーティションに行を追加します。

```
INSERT INTO sales VALUES
  * (10, '4519b', 'FRANCE', '17-Jan-2012', '45000'),
```

(次のページに続く)

(前のページからの続き)

```
* (40, '9519b', 'US', '12-Apr-2012', '145000'),
* (40, '4577b', 'US', '11-Nov-2012', '25000'),
* (30, '7588b', 'CANADA', '14-Dec-2012', '50000'),
* (30, '9519b', 'CANADA', '01-Feb-2012', '75000'),
* (30, '4519b', 'CANADA', '08-Apr-2012', '120000'),
* (40, '3788a', 'US', '12-May-2012', '4950'),
* (10, '9519b', 'ITALY', '07-Jul-2012', '15000'),
* (10, '9519a', 'FRANCE', '18-Aug-2012', '650000'),
* (10, '9519b', 'FRANCE', '18-Aug-2012', '650000'),
* (40, '4788a', 'US', '23-Sept-2012', '4950'),
* (40, '4788b', 'US', '09-Oct-2012', '15000');
```

SELECT ステートメントは、行がサブパーティション間で正しく分散されていることを確認します。

```
edb=# SELECT tableoid::regclass, * FROM sales;
   tableoid   | dept_no | part_no | country |          date          | amount
-----+-----+-----+-----+-----+-----
sales_p1_americas |      40 | 9519b   | US      | 12-APR-12 00:00:00 | 145000
sales_p1_americas |      30 | 9519b   | CANADA  | 01-FEB-12 00:00:00 | 75000
sales_p1_americas |      30 | 4519b   | CANADA  | 08-APR-12 00:00:00 | 120000
sales_p1_americas |      40 | 3788a   | US      | 12-MAY-12 00:00:00 | 4950
sales_p1_europe   |      10 | 4519b   | FRANCE  | 17-JAN-12 00:00:00 | 45000
sales_p2_americas |      40 | 4577b   | US      | 11-NOV-12 00:00:00 | 25000
sales_p2_americas |      30 | 7588b   | CANADA  | 14-DEC-12 00:00:00 | 50000
sales_p2_americas |      40 | 4788a   | US      | 23-SEP-12 00:00:00 | 4950
sales_p2_americas |      40 | 4788b   | US      | 09-OCT-12 00:00:00 | 15000
sales_p2_europe   |      10 | 9519b   | ITALY   | 07-JUL-12 00:00:00 | 15000
sales_p2_europe   |      10 | 9519a   | FRANCE  | 18-AUG-12 00:00:00 | 650000
sales_p2_europe   |      10 | 9519b   | FRANCE  | 18-AUG-12 00:00:00 | 650000
(12 rows)
```

次のコマンドは、p2_americas サブパーティションを 2 つの新しいサブパーティションに分割し、コンテンツを再配布します。

```
ALTER TABLE sales SPLIT SUBPARTITION p2_americas
* VALUES ('US')
* INTO
* (
*   SUBPARTITION p2_us,
*   SUBPARTITION p2_canada
* );
```

コマンドを呼び出した後、p2_americas サブパーティションは削除されました。その代わりに、サーバーは 2 つの新しいサブパーティション (p2_us と p2_canada) を作成しました。

```
edb=# SELECT partition_name, subpartition_name, high_value FROM ALL_TAB_SUBPARTITIONS;
* partition_name | subpartition_name | high_value
-----+-----+-----
FIRST_HALF_2012 | P1_EUROPE         | 'ITALY', 'FRANCE'
FIRST_HALF_2012 | P1_AMERICAS       | 'US', 'CANADA'
SECOND_HALF_2012 | P2_EUROPE         | 'ITALY', 'FRANCE'
SECOND_HALF_2012 | P2_US             | 'US'
SECOND_HALF_2012 | P2_CANADA         | 'CANADA'
(5 rows)
```

sales テーブルのクエリは、p2_americas サブパーティションのコンテンツが再配布されたことを示しています。

```
edb=# SELECT tableoid::regclass, * FROM sales;
* tableoid | dept_no | part_no | country | date | amount
-----+-----+-----+-----+-----+-----
sales_p1_americas | 40 | 9519b | US | 12-APR-12 00:00:00 | 145000
sales_p1_americas | 30 | 9519b | CANADA | 01-FEB-12 00:00:00 | 75000
sales_p1_americas | 30 | 4519b | CANADA | 08-APR-12 00:00:00 | 120000
sales_p1_americas | 40 | 3788a | US | 12-MAY-12 00:00:00 | 4950
sales_p1_europe | 10 | 4519b | FRANCE | 17-JAN-12 00:00:00 | 45000
sales_p2_canada | 30 | 7588b | CANADA | 14-DEC-12 00:00:00 | 50000
sales_p2_europ | 10 | 9519b | ITALY | 07-JUL-12 00:00:00 | 15000
sales_p2_europe | 10 | 9519a | FRANCE | 18-AUG-12 00:00:00 | 650000
sales_p2_europe | 10 | 9519b | FRANCE | 18-AUG-12 00:00:00 | 650000
sales_p2_us | 40 | 4577b | US | 11-NOV-12 00:00:00 | 25000
sales_p2_us | 40 | 4788a | US | 23-SEP-12 00:00:00 | 4950
sales_p2_us | 40 | 4788b | US | 09-OCT-12 00:00:00 | 15000
(12 rows)
```

3.5.2 Example - Splitting a RANGE Subpartition

</ div>

次の例では、レンジサブパーティションを分割し、サブパーティションのコンテンツを2つの新しいサブパーティションに再分配します。サンプルテーブル (sales) は、次のコマンドで作成されました。

```
CREATE TABLE sales
(
  * dept_no      number,
  * part_no      varchar2,
  * country      varchar2(20),
  * date         date,
  * amount       number
)
```

(次のページに続く)

(前のページからの続き)

```

PARTITION BY LIST(country)
  * SUBPARTITION BY RANGE(date)
(
  * PARTITION europe VALUES('FRANCE', 'ITALY')
  * (
  *   SUBPARTITION europe_2011
  *     VALUES LESS THAN('2012-Jan-01'),
  *   SUBPARTITION europe_2012
  *     VALUES LESS THAN('2013-Jan-01')
  * ),
  * PARTITION asia VALUES('INDIA', 'PAKISTAN')
  * (
  *   SUBPARTITION asia_2011
  *     VALUES LESS THAN('2012-Jan-01'),
  *   SUBPARTITION asia_2012
  *     VALUES LESS THAN('2013-Jan-01')
  * ),
  * PARTITION americas VALUES('US', 'CANADA')
  * (
  *   SUBPARTITION americas_2011
  *     VALUES LESS THAN('2012-Jan-01'),
  *   SUBPARTITION americas_2012
  *     VALUES LESS THAN('2013-Jan-01')
  * )
);

```

sales テーブルには3つのパーティション (europe, asia、および americas) があります。各パーティションには、パーティションの内容を date 列の値でサブパーティションにソートする、範囲が定義された2つのサブパーティションがあります。

```

edb=# SELECT partition_name, subpartition_name, high_value FROM ALL_TAB_SUBPARTITIONS;
partition_name | subpartition_name |      high_value
-----+-----+-----
EUROPE         | EUROPE_2011       | '01-JAN-12 00:00:00'
EUROPE         | EUROPE_2012       | '01-JAN-13 00:00:00'
ASIA           | ASIA_2011         | '01-JAN-12 00:00:00'
ASIA           | ASIA_2012         | '01-JAN-13 00:00:00'
AMERICAS       | AMERICAS_2011     | '01-JAN-12 00:00:00'
AMERICAS       | AMERICAS_2012     | '01-JAN-13 00:00:00'
(6 rows)

```

次のコマンドは、各サブパーティションに行を追加します。

```

INSERT INTO sales VALUES
  * (10, '4519b', 'FRANCE', '17-Jan-2012', '45000'),

```

(次のページに続く)

(前のページからの続き)

```
* (20, '3788a', 'INDIA', '01-Mar-2012', '75000'),
* (40, '9519b', 'US', '12-Apr-2012', '145000'),
* (20, '3788a', 'PAKISTAN', '04-Jun-2012', '37500'),
* (40, '4577b', 'US', '11-Nov-2012', '25000'),
* (30, '7588b', 'CANADA', '14-Dec-2012', '50000'),
* (30, '9519b', 'CANADA', '01-Feb-2012', '75000'),
* (30, '4519b', 'CANADA', '08-Apr-2012', '120000'),
* (40, '3788a', 'US', '12-May-2012', '4950'),
* (10, '9519b', 'ITALY', '07-Jul-2012', '15000'),
* (10, '9519a', 'FRANCE', '18-Aug-2012', '650000'),
* (10, '9519b', 'FRANCE', '18-Aug-2012', '650000'),
* (20, '3788b', 'INDIA', '21-Sept-2012', '5090'),
* (40, '4788a', 'US', '23-Sept-2012', '4950'),
* (40, '4788b', 'US', '09-Oct-2012', '15000'),
* (20, '4519a', 'INDIA', '18-Oct-2012', '650000'),
* (20, '4519b', 'INDIA', '2-Dec-2012', '5090');
```

SELECT ステートメントは、行がサブパーティションに分散されていることを確認します。

```
edb=# SELECT tableoid::regclass, * FROM sales;
 tableoid          |dept_no|part_no| country |          date          | amount
-----+-----+-----+-----+-----+-----
sales_americas_2012|  40   | 9519b | US       | 12-APR-12 00:00:00    | 145000
sales_americas_2012|  40   | 4577b | US       | 11-NOV-12 00:00:00    | 25000
sales_americas_2012|  30   | 7588b | CANADA   | 14-DEC-12 00:00:00    | 50000
sales_americas_2012|  30   | 9519b | CANADA   | 01-FEB-12 00:00:00    | 75000
sales_americas_2012|  30   | 4519b | CANADA   | 08-APR-12 00:00:00    | 120000
sales_americas_2012|  40   | 3788a | US       | 12-MAY-12 00:00:00    | 4950
sales_americas_2012|  40   | 4788a | US       | 23-SEP-12 00:00:00    | 4950
sales_americas_2012|  40   | 4788b | US       | 09-OCT-12 00:00:00    | 15000
sales_europe_2012  |  10   | 4519b | FRANCE   | 17-JAN-12 00:00:00    | 45000
sales_europe_2012  |  10   | 9519b | ITALY    | 07-JUL-12 00:00:00    | 15000
sales_europe_2012  |  10   | 9519a | FRANCE   | 18-AUG-12 00:00:00    | 650000
sales_europe_2012  |  10   | 9519b | FRANCE   | 18-AUG-12 00:00:00    | 650000
sales_asia_2012    |  20   | 3788a | INDIA    | 01-MAR-12 00:00:00    | 75000
sales_asia_2012    |  20   | 3788a | PAKISTAN | 04-JUN-12 00:00:00    | 37500
sales_asia_2012    |  20   | 3788b | INDIA    | 21-SEP-12 00:00:00    | 5090
sales_asia_2012    |  20   | 4519a | INDIA    | 18-OCT-12 00:00:00    | 650000
sales_asia_2012    |  20   | 4519b | INDIA    | 02-DEC-12 00:00:00    | 5090
(17 rows)
```

次のコマンドは、americas_2012 サブパーティションを2つの新しいサブパーティションに分割し、コンテンツを再配布します。

```
ALTER TABLE sales
```

(次のページに続く)

(前のページからの続き)

```
* SPLIT SUBPARTITION americas_2012
* AT('2012-Jun-01')
* INTO
* (
*     SUBPARTITION americas_p1_2012,
*     SUBPARTITION americas_p2_2012
* );
```

コマンドを呼び出した後、americas_2012 サブパーティションは削除されました。その代わりに、サーバーは2つの新しいサブパーティション (americas_p1_2012 と americas_p2_2012) を作成しました。

```
edb=# SELECT partition_name, subpartition_name, high_value FROM ALL_TAB_SUBPARTITIONS;
partition_name | subpartition_name |      high_value
-----+-----+-----
EUROPE         | EUROPE_2011       | '01-JAN-12 00:00:00'
EUROPE         | EUROPE_2012       | '01-JAN-13 00:00:00'
ASIA           | ASIA_2011         | '01-JAN-12 00:00:00'
ASIA           | ASIA_2012         | '01-JAN-13 00:00:00'
AMERICAS       | AMERICAS_2011     | '01-JAN-12 00:00:00'
AMERICAS       | AMERICAS_P1_2012  | '01-JUN-12 00:00:00'
AMERICAS       | AMERICAS_P2_2012  | '01-JAN-13 00:00:00'
(7 rows)
```

sales テーブルのクエリは、サブパーティションのコンテンツが再配布されることを示しています。

```
edb=# SELECT tableoid::regclass, * FROM sales;
tableoid      | dept_no | part_no | country |      date      | amount
-----+-----+-----+-----+-----+-----
sales_americas_p1_2012 | 40      | 9519b  | US      | 12-APR-12 00:00:00 | 145000
sales_americas_p1_2012 | 30      | 9519b  | CANADA  | 01-FEB-12 00:00:00 | 75000
sales_americas_p1_2012 | 30      | 4519b  | CANADA  | 08-APR-12 00:00:00 | 120000
sales_americas_p1_2012 | 40      | 3788a  | US      | 12-MAY-12 00:00:00 | 4950
sales_americas_p2_2012 | 40      | 4577b  | US      | 11-NOV-12 00:00:00 | 25000
sales_americas_p2_2012 | 30      | 7588b  | CANADA  | 14-DEC-12 00:00:00 | 50000
sales_americas_p2_2012 | 40      | 4788a  | US      | 23-SEP-12 00:00:00 | 4950
sales_americas_p2_2012 | 40      | 4788b  | US      | 09-OCT-12 00:00:00 | 15000
sales_europe_2012     | 10      | 4519b  | FRANCE  | 17-JAN-12 00:00:00 | 45000
sales_europe_2012     | 10      | 9519b  | ITALY   | 07-JUL-12 00:00:00 | 15000
sales_europe_2012     | 10      | 9519a  | FRANCE  | 18-AUG-12 00:00:00 | 650000
sales_europe_2012     | 10      | 9519b  | FRANCE  | 18-AUG-12 00:00:00 | 650000
sales_asia_2012       | 20      | 3788a  | INDIA   | 01-MAR-12 00:00:00 | 75000
sales_asia_2012       | 20      | 3788a  | PAKISTAN | 04-JUN-12 00:00:00 | 37500
sales_asia_2012       | 20      | 3788b  | INDIA   | 21-SEP-12 00:00:00 | 5090
sales_asia_2012       | 20      | 4519a  | INDIA   | 18-OCT-12 00:00:00 | 650000
sales_asia_2012       | 20      | 4519b  | INDIA   | 02-DEC-12 00:00:00 | 5090
```

(次のページに続く)

(前のページからの続き)

(17 rows)

3.6 ALTER TABLE...EXCHANGE PARTITION

</div>

ALTER TABLE...EXCHANGE PARTITION コマンドは、既存のテーブルをパーティションと交換します。パーティションテーブルにラージのデータを追加する場合は、ALTER TABLE... EXCHANGE PARTITION コマンドを使用してバルクロードを実装できます。ALTER TABLE... EXCHANGE PARTITION コマンドを使用して、ストレージの古いデータまたは不要なデータを削除することもできます。

コマンド構文には2つの形式があります。

最初のフォームは、パーティションのテーブルを交換します：

```
ALTER TABLE <target_table>
  * EXCHANGE PARTITION <target_partition>
  * WITH TABLE <source_table>
  * [(INCLUDING | EXCLUDING) INDEXES]
  * [(WITH | WITHOUT) VALIDATION];
```

2 番目のフォームは、サブパーティションのテーブルを交換します。

```
ALTER TABLE <target_table>
  * EXCHANGE SUBPARTITION <target_subpartition>
  * WITH TABLE <source_table>
  * [(INCLUDING | EXCLUDING) INDEXES]
  * [(WITH | WITHOUT) VALIDATION];
```

説明

ALTER TABLE... EXCHANGE PARTITION コマンドが完了すると、target_partition に元々あったデータは source_table にあり、source_table に元々あったデータは target_partition にあります。

ALTER TABLE... EXCHANGE PARTITION コマンドは、LIST、RANGE または HASH パーティションテーブルのパーティションを交換できます。source_table の構造は target_table の構造とマッチする必要がある（両方のテーブルにマッチング列とデータ型が必要）、テーブル内に含まれるデータはパーティショニングのコンストレインに従う必要があります。

INCLUDING INDEXES 句が EXCHANGE PARTITION で指定されている場合、target_partition と source_table のマッチングインデックスが交換されます。source_table にマッチしない target_partition のインデックスが再構築され、その逆も同様です（つまり、target_partition にマッチしない source_table のインデックスも再構築されます）。

EXCLUDING INDEXES 句が EXCHANGE PARTITION で指定されている場合、target_partition および source_table のマッチングインデックスはスワップされますが、source_table でマッチしないマッチインデックスは無効としてマークされます（逆も同様です） target_partition も無効としてマークされます）。

以前に使用されたマッピングインデックス用語は、順序付けコマンドによって決定される照合順序オーダー、昇順または降順、最初の NULL または最後の NULL などの同じ属性を持つインデックスを指します。

INCLUDING INDEXES と EXCLUDING INDEXES の両方が省略された場合、デフォルトのアクションは EXCLUDING INDEXES の動作です。

前述の動作と同じ動作が、EXCHANGE SUBPARTITION 句で使用される target_subpartition に適用されます。

そのテーブルに対して ALTER TABLE... EXCHANGE PARTITION または ALTER TABLE... EXCHANGE SUBPARTITION を呼び出すには、テーブルを所有する必要があります。

パラメータ

target_table

パーティションまたはサブパーティションが存在するテーブルの名前（オプションでスキーマ修飾）。

target_partition

置き換えられるパーティションの名前。

target_subpartition

置き換えられるサブパーティションの名前。

source_table

target_partition または target_subpartition を置き換えるテーブルの名前。

3.6.1 Example - Exchanging a Table for a Partition

</div>

次の例は、sales テーブルのパーティション (americas) のテーブルのスワッピングを示しています。次のコマンドで sales テーブルを作成できます。

```
CREATE TABLE sales
(
  * dept_no      number,
  * part_no      varchar2,
  * country      varchar2(20),
  * date         date,
  * amount       number
)
PARTITION BY LIST(country)
(
  * PARTITION europe VALUES('FRANCE', 'ITALY'),
  * PARTITION asia VALUES('INDIA', 'PAKISTAN'),
  * PARTITION americas VALUES('US', 'CANADA')
);
```

次のコマンドを使用して、サンプルデータを sales テーブルに追加します。

```
INSERT INTO sales VALUES
  * (40, '9519b', 'US', '12-Apr-2012', '145000'),
  * (10, '4519b', 'FRANCE', '17-Jan-2012', '45000'),
  * (20, '3788a', 'INDIA', '01-Mar-2012', '75000'),
  * (20, '3788a', 'PAKISTAN', '04-Jun-2012', '37500'),
  * (10, '9519b', 'ITALY', '07-Jul-2012', '15000'),
  * (10, '9519a', 'FRANCE', '18-Aug-2012', '650000'),
  * (10, '9519b', 'FRANCE', '18-Aug-2012', '650000'),
  * (20, '3788b', 'INDIA', '21-Sept-2012', '5090'),
  * (20, '4519a', 'INDIA', '18-Oct-2012', '650000'),
  * (20, '4519b', 'INDIA', '2-Dec-2012', '5090');
```

sales テーブルを照会すると、americas パーティションに存在する行は 1 つだけであることがわかります。

```
edb=# SELECT tableoid::regclass, * FROM sales;
  *   tableoid   | dept_no | part_no | country |          date          | amount
-----+-----+-----+-----+-----+-----
sales_americas |      40 | 9519b   | US      | 12-APR-12 00:00:00 | 145000
sales_europe   |      10 | 4519b   | FRANCE  | 17-JAN-12 00:00:00 |  45000
sales_europe   |      10 | 9519b   | ITALY   | 07-JUL-12 00:00:00 |  15000
sales_europe   |      10 | 9519a   | FRANCE  | 18-AUG-12 00:00:00 | 650000
sales_europe   |      10 | 9519b   | FRANCE  | 18-AUG-12 00:00:00 | 650000
sales_asia     |      20 | 3788a   | INDIA   | 01-MAR-12 00:00:00 |  75000
sales_asia     |      20 | 3788a   | PAKISTAN | 04-JUN-12 00:00:00 |  37500
sales_asia     |      20 | 3788b   | INDIA   | 21-SEP-12 00:00:00 |   5090
sales_asia     |      20 | 4519a   | INDIA   | 18-OCT-12 00:00:00 | 650000
sales_asia     |      20 | 4519b   | INDIA   | 02-DEC-12 00:00:00 |   5090
(10 rows)
```

次のコマンドは、sales テーブルの定義に一致するテーブル (n_america) を作成します。

```
CREATE TABLE n_america
(
  * dept_no      number,
  * part_no      varchar2,
  * country      varchar2(20),
  * date         date,
  * amount       number
);
```

次のコマンドは、n_america テーブルにデータを追加します。データは americas パーティションのパーティショニングルールに準拠しています。

```
INSERT INTO n_america VALUES
  * (40, '9519b', 'US', '12-Apr-2012', '145000'),
```

(次のページに続く)

(前のページからの続き)

```
* (40, '4577b', 'US', '11-Nov-2012', '25000'),
* (30, '7588b', 'CANADA', '14-Dec-2012', '50000'),
* (30, '9519b', 'CANADA', '01-Feb-2012', '75000'),
* (30, '4519b', 'CANADA', '08-Apr-2012', '120000'),
* (40, '3788a', 'US', '12-May-2012', '4950'),
* (40, '4788a', 'US', '23-Sept-2012', '4950'),
* (40, '4788b', 'US', '09-Oct-2012', '15000');
```

次のコマンドは、テーブルをパーティションテーブルにスワップします。

```
ALTER TABLE sales
* EXCHANGE PARTITION americas
* WITH TABLE n_america;
```

sales テーブルのクエリは、n_america テーブルのコンテンツが americas パーティションのコンテンツと交換されたことを示しています。

```
edb=# SELECT tableoid::regclass, * FROM sales;
*   tableoid   | dept_no | part_no | country |          date          | amount
-----+-----+-----+-----+-----+-----
sales_americas |      40 | 9519b   | US      | 12-APR-12 00:00:00    | 145000
sales_americas |      40 | 4577b   | US      | 11-NOV-12 00:00:00    | 25000
sales_americas |      30 | 7588b   | CANADA  | 14-DEC-12 00:00:00    | 50000
sales_americas |      30 | 9519b   | CANADA  | 01-FEB-12 00:00:00    | 75000
sales_americas |      30 | 4519b   | CANADA  | 08-APR-12 00:00:00    | 120000
sales_americas |      40 | 3788a   | US      | 12-MAY-12 00:00:00    | 4950
sales_americas |      40 | 4788a   | US      | 23-SEP-12 00:00:00    | 4950
sales_americas |      40 | 4788b   | US      | 09-OCT-12 00:00:00    | 15000
sales_europe   |      10 | 4519b   | FRANCE  | 17-JAN-12 00:00:00    | 45000
sales_europe   |      10 | 9519b   | ITALY   | 07-JUL-12 00:00:00    | 15000
sales_europe   |      10 | 9519a   | FRANCE  | 18-AUG-12 00:00:00    | 650000
sales_europe   |      10 | 9519b   | FRANCE  | 18-AUG-12 00:00:00    | 650000
sales_asia     |      20 | 3788a   | INDIA   | 01-MAR-12 00:00:00    | 75000
sales_asia     |      20 | 3788a   | PAKISTAN| 04-JUN-12 00:00:00    | 37500
sales_asia     |      20 | 3788b   | INDIA   | 21-SEP-12 00:00:00    | 5090
sales_asia     |      20 | 4519a   | INDIA   | 18-OCT-12 00:00:00    | 650000
sales_asia     |      20 | 4519b   | INDIA   | 02-DEC-12 00:00:00    | 5090
(17 rows)
```

n_america テーブルを照会すると、以前に americas パーティションに格納されていた行が n_america テーブルに移動されたことがわかります。

```
edb=# SELECT tableoid::regclass, * FROM n_america;
tableoid | dept_no | part_no | country |          date          | amount
-----+-----+-----+-----+-----+-----
```

(次のページに続く)

(前のページからの続き)

n_america	40	9519b	US	12-APR-12 00:00:00	145000
-----------	----	-------	----	--------------------	--------

(1 row)

3.7 ALTER TABLE...MOVE PARTITION

</div>

ALTER TABLE... MOVE PARTITION コマンドを使用して、パーティションを別のテーブルスペースに移動します。コマンドには2つの形式があります。

最初のフォームはパーティションを新しいテーブルスペースに移動します：

```
ALTER TABLE <table_name>
  * MOVE PARTITION <partition_name>
  * TABLESPACE <tablespace_name>;
```

2番目のフォームは、サブパーティションを新しいテーブルスペースに移動します。

```
ALTER TABLE <table_name>
  * MOVE SUBPARTITION <subpartition_name>
  * TABLESPACE <tablespace_name>;
```

説明

ALTER TABLE...MOVE PARTITION コマンドは、パーティションを現在のテーブルスペースから別のテーブルスペース領域に移動します。ALTER TABLE... MOVE PARTITION コマンドは、LIST、RANGE または HASH パーティションテーブルのパーティションを移動できます。

前述の動作と同じ動作が、MOVE SUBPARTITION 句で使用される subpartition_name に適用されます。

ALTER TABLE... MOVE PARTITION または ALTER TABLE... MOVE SUBPARTITION を呼び出すには、テーブルを所有する必要があります。

パラメータ

table_name

パーティションまたはサブパーティションが存在するテーブルの名前（オプションでスキーマ修飾）。

partition_name

移動するパーティションの名前。

subpartition_name

移動するサブパーティションの名前。

tablespace_name

パーティションまたはサブパーティションの移動先のテーブルスペースの名前。

3.7.1 Example - Moving a Partition to a Different Tablespace

</div>

次の例では、sales テーブルのパーティションをあるテーブルスペースから別のテーブルスペースに移動します。まず、次のコマンドで sales テーブルを作成します。

```
CREATE TABLE sales
(
  * dept_no      number,
  * part_no      varchar2,
  * country      varchar2(20),
  * date         date,
  * amount       number
)
PARTITION BY RANGE(date)
(
  * PARTITION q1_2012 VALUES LESS THAN ('2012-Apr-01'),
  * PARTITION q2_2012 VALUES LESS THAN ('2012-Jul-01'),
  * PARTITION q3_2012 VALUES LESS THAN ('2012-Oct-01'),
  * PARTITION q4_2012 VALUES LESS THAN ('2013-Jan-01') TABLESPACE ts_1,
  * PARTITION q1_2013 VALUES LESS THAN ('2013-Mar-01') TABLESPACE ts_2
);
```

ALL_TAB_PARTITIONS ビューをクエリすると、パーティションが予想されるサーバーとテーブルスペースに存在することが確認されます。

```
edb=# SELECT partition_name, tablespace_name FROM ALL_TAB_PARTITIONS;
partition_name | tablespace_name
-----+-----
Q1_2012        |
Q2_2012        |
Q3_2012        |
Q4_2012        | TS_1
Q1_2013        | TS_2
(5 rows)
```

ターゲットテーブルスペースを準備した後、ALTER TABLE... MOVE PARTITION コマンドを起動して、q1_2013 パーティションを ts_2 名前付けのテーブルスペースから ts_3 という名前付けのテーブルスペース領域に移動します。

```
ALTER TABLE sales MOVE PARTITION q1_2013 TABLESPACE ts_3;
```

ALL_TAB_PARTITIONS ビューを照会すると、移動が成功したことがわかります。

```
edb=# SELECT partition_name, tablespace_name FROM ALL_TAB_PARTITIONS;
partition_name | tablespace_name
```

(次のページに続く)

(前のページからの続き)

-----+-----
Q1_2012
Q2_2012
Q3_2012
Q4_2012 TS_1
Q1_2013 TS_3
(5 rows)

3.8 ALTER TABLE...RENAME PARTITION

</ div>

ALTER TABLE... RENAME PARTITION コマンドを使用して、テーブルパーティションの名前を変更します。構文は 2 つの形式を取ります。

最初のフォームはパーティションの名前を変更します：

```
ALTER TABLE <table_name>
  RENAME PARTITION <partition_name>
  TO <new_name>;
```

2 番目のフォームは、サブパーティションの名前を変更します。

```
ALTER TABLE <table_name>
  RENAME SUBPARTITION <subpartition_name>
  TO <new_name>;
```

説明

ALTER TABLE... RENAME PARTITION コマンドは、パーティションの名前を変更します。

前述の動作と同じ動作が、RENAME SUBPARTITION 句で使用される subpartition_name に適用されます。

ALTER TABLE... RENAME PARTITION または ALTER TABLE... RENAME SUBPARTITION を呼び出すには、指定されたテーブルを所有する必要があります。

パラメータ

table_name

パーティションまたはサブパーティションが存在するテーブルの名前（オプションでスキーマ修飾）。

partition_name

名前を変更するパーティションの名前。

subpartition_name

名前を変更するサブパーティションの名前。

new_name

パーティションまたはサブパーティションの新しい名前。

3.8.1 Example - Renaming a Partition

</div>

次のコマンドは、sales 名前付けのリストパーティションテーブルを作成します。

```
CREATE TABLE sales
(
  * dept_no      number,
  * part_no      varchar2,
  * country      varchar2(20),
  * date         date,
  * amount       number
)
PARTITION BY LIST(country)
(
  * PARTITION europe VALUES('FRANCE', 'ITALY'),
  * PARTITION asia VALUES('INDIA', 'PAKISTAN'),
  * PARTITION americas VALUES('US', 'CANADA')
);
```

ALL_TAB_PARTITIONS ビューを照会して、パーティション名を表示します。

```
edb=# SELECT partition_name, high_value FROM ALL_TAB_PARTITIONS;
partition_name |      high_value
-----+-----
EUROPE         | 'FRANCE', 'ITALY'
ASIA           | 'INDIA', 'PAKISTAN'
AMERICAS       | 'US', 'CANADA'
(3 rows)
```

次のコマンドは、americas パーティションの名前を n_america に変更します。

```
ALTER TABLE sales
  RENAME PARTITION americas TO n_america;
```

ALL_TAB_PARTITIONS ビューを照会すると、パーティションの名前が正常に変更されたことがわかります。

```
edb=# SELECT partition_name, high_value FROM ALL_TAB_PARTITIONS;
partition_name |      high_value
-----+-----
EUROPE         | 'FRANCE', 'ITALY'
ASIA           | 'INDIA', 'PAKISTAN'
N_AMERICA      | 'US', 'CANADA'
(3 rows)
```

3.9 ALTER TABLE...SET INTERVAL

</div>

ALTER TABLE... SET INTERVAL コマンドを使用して、既存の範囲パーティション表をインターバルレンジパーティション表に変換します。INTERVAL が設定されている場合、データベースはパーティションテーブルに指定されたレンジまたはインターバルの新しいパーティションを自動的に作成します。構文は次のとおりです。

```
ALTER TABLE <table_name> SET INTERVAL (<constant> | <expression>);
```

時インターバルレンジパーティション表を無効にして範囲パーティション表に変換するための構文は次のとおりです。

```
ALTER TABLE <table_name> SET INTERVAL ();
```

パラメータ

table_name

範囲パーティションテーブルの名前（オプションでスキーマ修飾）。

constant | expression

NUMERIC, DATE または TIME の値を指定します。

説明

ALTER TABLE... SET INTERVAL コマンドを使用して、範囲パーティション表を変換して、インターバルレンジパーティショニングを使用できます。指定したインターバルの新しいパーティションが作成され、データを新しいパーティションに挿入できます。

SET INTERVAL () コマンドを使用して、インターバルレンジのパーティショニング化を無効にできます。データベースは、インターバルレンジパーティションテーブルを範囲パーティションに変換し、インターバルレンジパーティションの境界をレンジパーティションの境界に設定します。

3.9.1 Example - Setting an Interval Range Partition

</div>

次の例では、sales 表のインターバルレンジパーティションをレンジパーティショニングから設定して、毎月のインターバルレンジパーティショニングの使用をスタートします。次のコマンドを使用して、sales テーブルを作成します。

```
CREATE TABLE sales
(
  * prod_id          int,
  * prod_quantity    int,
  * sold_month        date
)
```

(次のページに続く)

(前のページからの続き)

```

PARTITION BY RANGE(sold_month)
(
  * PARTITION p1
  *   VALUES LESS THAN('15-JAN-2019'),
  * PARTITION p2
  *   VALUES LESS THAN('15-FEB-2019')
);

```

sales テーブルからインターバルレンジパーティショニングを設定するには、次のコマンドを呼び出します。

```
ALTER TABLE sales SET INTERVAL (NUMTOYMINTERVAL(1, 'MONTH'));
```

データベースがインターバルレンジパーティションを作成する前に、ALL_TAB_PARTITIONS ビューを照会します。

```

edb=# SELECT partition_name, high_value from ALL_TAB_PARTITIONS;
partition_name |      high_value
-----+-----
P1              | '15-JAN-19 00:00:00'
P2              | '15-FEB-19 00:00:00'
(2 rows)

```

次に、レンジパーティションの高値を超えるデータを sales テーブルに追加します。

```

edb=# INSERT INTO sales VALUES (1,100,'05-APR-2019');
INSERT 0 1

```

次に、INSERT ステートメントの後に ALL_TAB_PARTITIONS ビューを再度クエリーします。インターバルレンジパーティションが正常に作成され、データが挿入されます。セッションごとに異なるインターバルレンジパーティションのシステム生成名前が作成されます。

```

edb=# SELECT partition_name, high_value from ALL_TAB_PARTITIONS;
partition_name |      high_value
-----+-----
P1              | '15-JAN-19 00:00:00'
P2              | '15-FEB-19 00:00:00'
SYS916340103    | '15-APR-19 00:00:00'
(3 rows)

```

3.10 ALTER TABLE...SET [PARTITIONING] AUTOMATIC

</div>

ALTER TABLE... SET [PARTITIONING] AUTOMATIC コマンドを使用して、既存のリストパーティションテーブルを自動リストパーティショニングに変換します。データベースは、AUTOMATIC が設定されたときにテーブルに挿入された新しい値に基づいてパーティションを自動的に作成します。構文は次のとおりです。

```
ALTER TABLE <table_name> SET [ PARTITIONING ] AUTOMATIC;
```

AUTOMATIC LIST PARTITIONING を無効にし、通常の LIST パーティションに変換するための構文は次のとおりです。

```
ALTER TABLE <table_name> SET [ PARTITIONING ] MANUAL;
```

パラメータ

table_name

リストパーティションテーブルの名前（オプションでスキーマ修飾）。

説明

ALTER TABLE... SET [PARTITIONING] AUTOMATIC コマンドを使用して、通常のリストパーティションテーブルを自動パーティショニングを使用するように変換できます。パーティションが作成され、新しいパーティションにデータを挿入できます。

ALTER TABLE... SET [PARTITIONING] MANUAL コマンドを使用して、自動リストパーティショニングを無効にすることができます。データベースは、自動リストパーティションテーブルを通常のリストパーティションテーブルに変換します。

3.10.1 Example - Setting an AUTOMATIC List Partition

</div>

次の例では、テーブル sales を変更して、通常のリストパーティショニング分割の代わりに自動リストパーティションを使用しています。次のコマンドを使用して、sales テーブルを作成します。

```
CREATE TABLE sales
(
  * dept_no      number,
  * part_no      varchar2,
  * sales_state  varchar2(20),
  * date         date,
  * amount       number
)
PARTITION BY LIST(sales_state)
(
  * PARTITION P_KAN VALUES('KANSAS'),
```

(次のページに続く)

(前のページからの続き)

```
* PARTITION P_TEX VALUES('TEXAS')
);
```

sales テーブルに自動リストパーティショニングを実装するには、次のコマンドを呼び出します。

```
ALTER TABLE sales SET AUTOMATIC;
```

ALL_TAB_PARTITIONS ビューを照会して、既存のパーティションが正常に作成されたことを確認します。

```
edb=# SELECT table_name, partition_name, high_value FROM ALL_TAB_PARTITIONS;
table_name | partition_name | high_value
-----+-----+-----
SALES      | P_KAN          | 'KANSAS'
SALES      | P_TEX          | 'TEXAS'
(2 rows)
```

次に、データを sales テーブルに挿入して、新しいパーティションを作成し、新しい値を追加します。

```
edb=# INSERT INTO sales VALUES (1, 'VIR', 'VIRGINIA');
INSERT 0 1
```

次に、挿入後に ALL_TAB_PARTITIONS ビューを再度クエリーします。自動リストパーティションが正常に作成され、データが挿入されます。セッションごとに異なるシステム生成のパーティション名前が作成されます。

```
edb=# SELECT table_name, partition_name, high_value FROM ALL_TAB_PARTITIONS;
table_name | partition_name | high_value
-----+-----+-----
SALES      | P_KAN          | 'KANSAS'
SALES      | P_TEX          | 'TEXAS'
SALES      | SYS106900103   | 'VIRGINIA'
(3 rows)
```

3.11 ALTER TABLE...SET SUBPARTITION TEMPLATE

</div>

ALTER TABLE... SET SUBPARTITION TEMPLATE コマンドを使用して、テーブルのサブパーティションテンプレートを更新します。コマンド構文は次の形式で提供されます。

HASH サブパーティションのみにサブパーティション番号を設定するには：

```
ALTER TABLE <table_name> SET SUBPARTITION TEMPLATE <num>;
```

既存のパーティションテーブルにサブパーティションの説明を設定するには：

```
ALTER TABLE <table_name>
[SET SUBPARTITION TEMPLATE (<subpartition_template_description>)]
```

サブパーティションテンプレートをを使用してサブパーティション番号をデフォルトにリセットするには：

```
ALTER TABLE <table_name> SET SUBPARTITION TEMPLATE ();
```

パラメータ

table_name

パーティション化されたテーブルの名前（オプションでスキーマ修飾）。

num

パーティションに追加されるサブパーティションの数。

subpartition_template_description

サブパーティション名前、バインドされたサブパーティション、およびテーブルスペース名前を含むサブパーティションの詳細のリスト。

説明

ALTER TABLE... SET SUBPARTITION TEMPLATE コマンドを使用して、レンジ、リストまたはハッシュパーティションテーブルのサブパーティションテンプレートを更新できます。パーティションにサブパーティションディスクリプタを指定する場合、サブパーティションテンプレートの代わりにサブパーティションディスクリプタが使用されます。サブパーティションディスクリプタは、パーティションに指定されていない場合はいつでも使用できます。サブパーティションディスクリプタまたはサブパーティションテンプレートのいずれかが指定されていない場合、デフォルトで単一のサブパーティションが作成されます。サブパーティションテンプレートの作成の詳細については、[CREATE TABLE...PARTITION BY](#) を参照してください。

!!! Note * ALTER TABLE... SET SUBPARTITION TEMPLATE コマンドの呼び出し後にテーブルに追加されたパーティションは、新しい SUBPARTITION TEMPLATE を使用します。

ALTER TABLE... SET SUBPARTITION TEMPLATE () コマンドを使用して、サブパーティション番号をデフォルトの 1 にリセットできます。

3.11.1 Example - Setting a SUBPARTITION TEMPLATE

</div>

次の例では、**date** でレンジパーティション化され、**country** でハッシュサブパーティション化されたテーブル **sales** を作成します。次のコマンドを使用して、**sales** テーブルを作成します。

```
CREATE TABLE sales
(
  * dept_no      number,
  * part_no      varchar2,
```

(次のページに続く)

(前のページからの続き)

```

* country      varchar2(20),
* date         date,
* amount       number
)
PARTITION BY RANGE (date) SUBPARTITION BY HASH (country) SUBPARTITIONS 2
(
* PARTITION q1_2012
*   VALUES LESS THAN('2012-Apr-01'),
* PARTITION q2_2012
*   VALUES LESS THAN('2012-Jul-01'),
* PARTITION q3_2012
*   VALUES LESS THAN('2012-Oct-01'),
* PARTITION q4_2012
*   VALUES LESS THAN('2013-Jan-01')
);

```

テーブル定義は、4つのパーティション (q1_2012, q2_2012, q3_2012 および q4_2012) を作成します。各パーティションは、システム生成の名前を持つ2つのサブパーティションで構成されます。

```

edb=# SELECT table_name, partition_name, subpartition_name FROM
ALL_TAB_SUBPARTITIONS WHERE table_name = 'SALES' ORDER BY 1,2;

```

table_name	partition_name	subpartition_name
SALES	Q1_2012	SYS0101
SALES	Q1_2012	SYS0102
SALES	Q2_2012	SYS0103
SALES	Q2_2012	SYS0104
SALES	Q3_2012	SYS0105
SALES	Q3_2012	SYS0106
SALES	Q4_2012	SYS0107
SALES	Q4_2012	SYS0108

(8 rows)

sales テーブルにサブパーティションテンプレートを設定するには、次のコマンドを呼び出します。

```
ALTER TABLE sales SET SUBPARTITION TEMPLATE 8;
```

sales テーブルが変更され、サブパーティションテンプレートが8に設定されます。ここで、新しいパーティション q1_2013 を追加しようとする、サブパーティションテンプレートで説明されているように、8つのサブパーティションで構成される新しいパーティションが作成されます。

```
ALTER TABLE sales ADD PARTITION q1_2013 VALUES LESS THAN ('2013-Apr-01');
```

ALL_TAB_PARTITIONS ビューを照会すると、システム生成の名前が割り当てられた8つのサブパーティションで構成される q1_2013 パーティションが正常に追加されます。

```
edb=# SELECT table_name, partition_name, subpartition_name FROM
ALL_TAB_SUBPARTITIONS WHERE table_name = 'SALES' and partition_name =
'Q1_2013' ORDER BY 1,2;
 table_name | partition_name | subpartition_name
-----+-----+-----
SALES      | Q1_2013       | SYS0113
SALES      | Q1_2013       | SYS0114
SALES      | Q1_2013       | SYS0115
SALES      | Q1_2013       | SYS0116
SALES      | Q1_2013       | SYS0117
SALES      | Q1_2013       | SYS0118
SALES      | Q1_2013       | SYS0119
SALES      | Q1_2013       | SYS0120
(8 rows)
```

例-LIST / LIST パーティションテーブルのサブパーティションテンプレートの追加

次の例では、country によってリストパーティション化され、date 列によってリストを使用してサブパーティション化されたテーブル sales を作成します。

```
CREATE TABLE sales
(
  * dept_no      number,
  * part_no      varchar2,
  * country      varchar2(20),
  * date         date,
  * amount       number
)
PARTITION BY LIST (country)
  * SUBPARTITION BY LIST (date)
  * SUBPARTITION TEMPLATE
(
  * SUBPARTITION europe VALUES('2021-Jan-01') TABLESPACE ts1,
  * SUBPARTITION asia VALUES('2021-Apr-01') TABLESPACE ts2,
  * SUBPARTITION americas VALUES('2021-Jul-01') TABLESPACE ts3
)
(
  * PARTITION q1_2021 VALUES('2021-Jul-01')
);
```

SELECT ステートメントは、テーブルスペース (ts1)、(ts2)、および (ts3) に格納されている 3 つのサブパーティションで構成されるパーティション q1_2021 を示しています。

```
edb=# SELECT partition_name, subpartition_name, high_value, tablespace_name FROM
DBA_TAB_SUBPARTITIONS WHERE table_name = 'SALES' ORDER BY 1,2;
 partition_name | subpartition_name |      high_value      | tablespace_name
-----+-----+-----+-----
 Q1_2021        | Q1_2021_AMERICAS  | '01-JUL-21 00:00:00' | TS3
 Q1_2021        | Q1_2021_ASIA      | '01-APR-21 00:00:00' | TS2
 Q1_2021        | Q1_2021_EUROPE    | '01-JAN-21 00:00:00' | TS1
(3 rows)
```

次のコマンドは、q2_2021 名前付けのパーティションを sales テーブルに追加します。

```
ALTER TABLE sales ADD PARTITION q2_2021 VALUES('US', 'CANADA');
```

次のコマンドを呼び出します。sales テーブルには q2_2021 パーティションが含まれます。

```
edb=# SELECT partition_name, subpartition_name, high_value, tablespace_name FROM
DBA_TAB_SUBPARTITIONS WHERE table_name = 'SALES' ORDER BY 1,2;
 partition_name | subpartition_name |      high_value      | tablespace_name
-----+-----+-----+-----
 Q1_2021        | Q1_2021_AMERICAS  | '01-JUL-21 00:00:00' | TS3
 Q1_2021        | Q1_2021_ASIA      | '01-APR-21 00:00:00' | TS2
 Q1_2021        | Q1_2021_EUROPE    | '01-JAN-21 00:00:00' | TS1
 Q2_2021        | Q2_2021_AMERICAS  | '01-JUL-21 00:00:00' | TS3
 Q2_2021        | Q2_2021_ASIA      | '01-APR-21 00:00:00' | TS2
 Q2_2021        | Q2_2021_EUROPE    | '01-JAN-21 00:00:00' | TS1
(6 rows)
```

例-LIST / RANGE パーティションテーブルのサブパーティションテンプレートを追加する

次の例では、country によってリストパーティション化され、date 列によるレンジパーティショニングを使用してサブパーティション化されたテーブル sales を作成します。

```
CREATE TABLE sales
(
  * * dept_no      number,
  * * part_no      varchar2,
  * * country      varchar2(20),
  * * date         date,
  * * amount       number
)
PARTITION BY LIST(country)
  * * SUBPARTITION BY RANGE(date)
  * * SUBPARTITION TEMPLATE
(
  * * SUBPARTITION europe VALUES LESS THAN('2021-Jan-01') TABLESPACE ts1,
```

(次のページに続く)

(前のページからの続き)

```

* * SUBPARTITION asia VALUES LESS THAN('2021-Apr-01') TABLESPACE ts2,
* * SUBPARTITION americas VALUES LESS THAN('2021-Jul-01') TABLESPACE ts3
)
(
* * PARTITION q1_2021 VALUES ('2021-Jul-01')
);

```

sales テーブルは、テーブルスペース (ts1)、(ts2)、および (ts3) に格納された 3 つのサブパーティションを含む q1_2021 という名前付けのパーティションを作成します。

```

edb=# SELECT partition_name, subpartition_name, high_value, tablespace_name FROM
DBA_TAB_SUBPARTITIONS WHERE table_name = 'SALES' ORDER BY 1,2;

```

partition_name	subpartition_name	high_value	tablespace_name
Q1_2021	Q1_2021_AMERICAS	'01-JUL-21 00:00:00'	TS3
Q1_2021	Q1_2021_ASIA	'01-APR-21 00:00:00'	TS2
Q1_2021	Q1_2021_EUROPE	'01-JAN-21 00:00:00'	TS1

(3 rows)

次のコマンドは、q2_2021 名前付けのパーティションを sales テーブルに追加します。

```
ALTER TABLE sales ADD PARTITION q2_2021 VALUES('INDIA', 'PAKISTAN');
```

次のコマンドを呼び出します。sales テーブルには q2_2021 パーティションが含まれます。

```

edb=# SELECT partition_name, subpartition_name, high_value, tablespace_name FROM
DBA_TAB_SUBPARTITIONS WHERE table_name = 'SALES' ORDER BY 1,2;

```

partition_name	subpartition_name	high_value	tablespace_name
Q1_2021	Q1_2021_AMERICAS	'01-JUL-21 00:00:00'	TS3
Q1_2021	Q1_2021_ASIA	'01-APR-21 00:00:00'	TS2
Q1_2021	Q1_2021_EUROPE	'01-JAN-21 00:00:00'	TS1
Q2_2021	Q2_2021_AMERICAS	'01-JUL-21 00:00:00'	TS3
Q2_2021	Q2_2021_ASIA	'01-APR-21 00:00:00'	TS2
Q2_2021	Q2_2021_EUROPE	'01-JAN-21 00:00:00'	TS1

(6 rows)

例-LIST / HASH パーティションテーブルのサブパーティションテンプレートを追加する

次の例では、最初に `country` によってパーティション化され、次に `dept_no` 列の値を使用してハッシュサブパーティション化されるリストパーティションテーブル `sales` を作成します。

```
CREATE TABLE sales
(
  * dept_no      number,
  * part_no      varchar2,
  * country      varchar2(20),
  * date         date,
  * amount       number
)
PARTITION BY LIST(country)
  * SUBPARTITION BY HASH (dept_no)
  * SUBPARTITION TEMPLATE
(
  * SUBPARTITION europe TABLESPACE ts1,
  * SUBPARTITION asia TABLESPACE ts2,
  * SUBPARTITION americas TABLESPACE ts3
)
(
  * PARTITION q1_2021 VALUES ('2021-Jul-01')
);
```

`sales` テーブルは、テーブルスペース (`ts1`)、(`ts2`)、および (`ts3`) に格納されている 3 つのサブパーティションを含む `q1_2021` パーティションを作成します。

```
edb=# SELECT partition_name, subpartition_name, high_value, tablespace_name
FROM DBA_TAB_SUBPARTITIONS WHERE table_name = 'SALES' ORDER BY 1,2;
partition_name | subpartition_name | high_value | tablespace_name
-----+-----+-----+-----
Q1_2021        | Q1_2021_AMERICAS |           | TS3
Q1_2021        | Q1_2021_ASIA     |           | TS2
Q1_2021        | Q1_2021_EUROPE   |           | TS1
(3 rows)
```

次のコマンドは、`q2_2021` 名前付けのパーティションを `sales` テーブルに追加します。

```
ALTER TABLE sales ADD PARTITION q2_2021 VALUES('FRANCE', 'ITALY');
```

次のコマンドを呼び出します。 `sales` テーブルには `q2_2021` パーティションが含まれます。

```
edb=# SELECT partition_name, subpartition_name, high_value, tablespace_name
FROM DBA_TAB_SUBPARTITIONS WHERE table_name = 'SALES' ORDER BY 1,2;
partition_name | subpartition_name | high_value | tablespace_name
-----+-----+-----+-----
```

(次のページに続く)

(前のページからの続き)

Q1_2021	Q1_2021_AMERICAS		TS3
Q1_2021	Q1_2021_ASIA		TS2
Q1_2021	Q1_2021_EUROPE		TS1
Q2_2021	Q2_2021_AMERICAS		TS3
Q2_2021	Q2_2021_ASIA		TS2
Q2_2021	Q2_2021_EUROPE		TS1

(6 rows)

例-サブパーティションテンプレートのリセット

例 1

次の例では、`country` でリストパーティション化され、`part_no` でハッシュサブパーティション化されたリストパーティションテーブル `sales` を作成します。次のコマンドを使用して、`sales` テーブルを作成します。

```
CREATE TABLE sales
(
  * dept_no      number,
  * part_no      varchar2,
  * country      varchar2(20),
  * date         date,
  * amount       number
)
PARTITION BY LIST (country) SUBPARTITION BY HASH (part_no) SUBPARTITIONS 3
(
  * PARTITION europe VALUES('FRANCE', 'ITALY'),
  * PARTITION asia VALUES('INDIA', 'PAKISTAN'),
  * PARTITION americas VALUES('US', 'CANADA')
);
```

テーブルには3つのパーティション (americas、asia、および europe) が含まれ、各パーティションはシステム生成の名前を持つ3つのサブパーティションで構成されます。

```
edb=# SELECT table_name, partition_name, subpartition_name FROM
ALL_TAB_SUBPARTITIONS WHERE table_name = 'SALES' ORDER BY 1,2;
 table_name | partition_name | subpartition_name
-----+-----+-----
SALES      | AMERICAS      | SYS0109
SALES      | AMERICAS      | SYS0107
SALES      | AMERICAS      | SYS0108
SALES      | ASIA          | SYS0105
SALES      | ASIA          | SYS0104
SALES      | ASIA          | SYS0106
```

(次のページに続く)

(前のページからの続き)

SALES	EUROPE	SYS0101
SALES	EUROPE	SYS0103
SALES	EUROPE	SYS0102

(9 rows)

次のコマンドは、sales テーブルのサブパーティションテンプレートをリセットします。

```
ALTER TABLE sales SET SUBPARTITION TEMPLATE ();
```

sales テーブルが変更され、サブパーティションテンプレートがデフォルトの 1 にリセットされます。ここで、次のコマンドを使用して、新しいパーティション east_asia を追加してみてください。

```
ALTER TABLE sales ADD PARTITION east_asia VALUES ('CHINA', 'KOREA');
```

ALL_TAB_PARTITIONS ビューを照会すると、システムによって生成された名前が割り当てられた 1 つのサブパーティションで構成される新しいパーティション east_asia が作成されます。

```
edb=# SELECT table_name, partition_name, subpartition_name FROM
ALL_TAB_SUBPARTITIONS WHERE table_name = 'SALES' and partition_name =
'EAST_ASIA' ORDER BY 1,2;
 table_name | partition_name | subpartition_name
-----+-----+-----
SALES      | EAST_ASIA      | SYS0113
(1 row)
```

例 2

次の例では、country によってリストパーティション化され、date 列によるレンジパーティショニングを使用してサブパーティション化されたテーブル sales を作成します。

```
CREATE TABLE sales
(
  * * dept_no      number,
  * * part_no      varchar2,
  * * country      varchar2(20),
  * * date         date,
  * * amount       number
)
PARTITION BY LIST(country)
  * * SUBPARTITION BY RANGE(date)
  * * SUBPARTITION TEMPLATE
(
  * * SUBPARTITION europe VALUES LESS THAN('2021-Jan-01') TABLESPACE ts1,
  * * SUBPARTITION asia VALUES LESS THAN('2021-Apr-01') TABLESPACE ts2,
  * * SUBPARTITION americas VALUES LESS THAN('2021-Jul-01') TABLESPACE ts3
)
```

(次のページに続く)

(前のページからの続き)

```
(
  * * PARTITION q1_2021 VALUES ('2021-Jul-01')
);
```

sales テーブルには、(ts1)、(ts2)、および (ts3) のテーブルスペースに格納された3つのサブパーティションを含む q1_2021 という名前付けのパーティションが含まれます。

```
edb=# SELECT partition_name, subpartition_name, high_value, tablespace_name
FROM DBA_TAB_SUBPARTITIONS WHERE table_name = 'SALES' ORDER BY 1,2;
partition_name | subpartition_name | high_value | tablespace_name
-----+-----+-----+-----
Q1_2021        | Q1_2021_AMERICAS | '01-JUL-21 00:00:00' | TS3
Q1_2021        | Q1_2021_ASIA     | '01-APR-21 00:00:00' | TS2
Q1_2021        | Q1_2021_EUROPE   | '01-JAN-21 00:00:00' | TS1
(3 rows)
```

次のコマンドは、q2_2021 名前付けのパーティションを sales テーブルに追加します。

```
ALTER TABLE sales ADD PARTITION q2_2021 VALUES('INDIA', 'PAKISTAN');
```

次のコマンドを呼び出します。sales テーブルには q2_2021 パーティションが含まれます。

```
edb=# SELECT partition_name, subpartition_name, high_value, tablespace_name FROM
DBA_TAB_SUBPARTITIONS WHERE table_name = 'SALES' ORDER BY 1,2;
partition_name | subpartition_name | high_value | tablespace_name
-----+-----+-----+-----
Q1_2021        | Q1_2021_AMERICAS | '01-JUL-21 00:00:00' | TS3
Q1_2021        | Q1_2021_ASIA     | '01-APR-21 00:00:00' | TS2
Q1_2021        | Q1_2021_EUROPE   | '01-JAN-21 00:00:00' | TS1
Q2_2021        | Q2_2021_AMERICAS | '01-JUL-21 00:00:00' | TS3
Q2_2021        | Q2_2021_ASIA     | '01-APR-21 00:00:00' | TS2
Q2_2021        | Q2_2021_EUROPE   | '01-JAN-21 00:00:00' | TS1
(6 rows)
```

ALTER TABLE... SET SUBPARTITION TEMPLATE コマンドを使用して、新しいサブパーティションテンプレートを指定します。

```
ALTER TABLE sales SET SUBPARTITION TEMPLATE
(
  * SUBPARTITION europe VALUES LESS THAN('2021-Jan-01'),
  * SUBPARTITION asia VALUES LESS THAN('2021-Apr-01') TABLESPACE ts2
);
```

次のコマンドは、q3_2021 名前付けのパーティションを sales テーブルに追加します。

```
ALTER TABLE sales ADD PARTITION q3_2021 VALUES('US', 'CANADA');
```

次のコマンドを呼び出します。sales テーブルには q3_2021 パーティションが含まれます。

```
edb=# SELECT partition_name, subpartition_name, high_value, tablespace_name FROM
DBA_TAB_SUBPARTITIONS WHERE table_name = 'SALES' ORDER BY 1,2;
```

partition_name	subpartition_name	high_value	tablespace_name
Q1_2021	Q1_2021_AMERICAS	'01-JUL-21 00:00:00'	TS3
Q1_2021	Q1_2021_ASIA	'01-APR-21 00:00:00'	TS2
Q1_2021	Q1_2021_EUROPE	'01-JAN-21 00:00:00'	TS1
Q2_2021	Q2_2021_AMERICAS	'01-JUL-21 00:00:00'	TS3
Q2_2021	Q2_2021_ASIA	'01-APR-21 00:00:00'	TS2
Q2_2021	Q2_2021_EUROPE	'01-JAN-21 00:00:00'	TS1
Q3_2021	Q3_2021_ASIA	'01-APR-21 00:00:00'	TS2
Q3_2021	Q3_2021_EUROPE	'01-JAN-21 00:00:00'	PG_DEFAULT

(8 rows)

次のコマンドは、sales テーブルのサブパーティションテンプレートをリセットまたは削除します。

```
ALTER TABLE sales SET SUBPARTITION TEMPLATE ();
```

次のコマンドは、q4_2021 名前付けのパーティションを sales テーブルに追加します。

```
ALTER TABLE sales ADD PARTITION q4_2021 VALUES('FRANCE', 'ITALY');
```

SELECT ステートメントは、システム生成名前のサブパーティションで構成されるパーティション q4_2021 を示しています。

```
edb=# SELECT partition_name, subpartition_name, high_value, tablespace_name FROM
DBA_TAB_SUBPARTITIONS WHERE table_name = 'SALES' ORDER BY 1,2;
```

partition_name	subpartition_name	high_value	tablespace_name
Q1_2021	Q1_2021_AMERICAS	'01-JUL-21 00:00:00'	TS3
Q1_2021	Q1_2021_ASIA	'01-APR-21 00:00:00'	TS2
Q1_2021	Q1_2021_EUROPE	'01-JAN-21 00:00:00'	TS1
Q2_2021	Q2_2021_AMERICAS	'01-JUL-21 00:00:00'	TS3
Q2_2021	Q2_2021_ASIA	'01-APR-21 00:00:00'	TS2
Q2_2021	Q2_2021_EUROPE	'01-JAN-21 00:00:00'	TS1
Q3_2021	Q3_2021_ASIA	'01-APR-21 00:00:00'	TS2
Q3_2021	Q3_2021_EUROPE	'01-JAN-21 00:00:00'	PG_DEFAULT
Q4_2021	SYS0112	MAXVALUE	PG_DEFAULT

(9 rows)

3.12 DROP TABLE

</ div>

PostgreSQL DROP TABLE コマンドを使用して、パーティションテーブル定義、パーティションおよびサブパーティションを削除し、テーブルの内容を削除します。構文は次のとおりです。

```
DROP TABLE <table_name>
```

パラメータ

`table_name`

パーティション化されたテーブルの名前（オプションでスキーマ修飾）。

説明

DROP TABLE コマンドは、テーブル全体と、そのテーブルにあるデータを削除します。テーブルを削除すると、そのテーブルのパーティションまたはサブパーティションも削除されます。

DROP TABLE コマンドを使用するには、パーティショニングルート所有者、テーブルを所有するグループのメンバ、スキーマ所有者、またはデータベーススーパーユーザである必要があります。

例

テーブルを削除するには、コントローラノード（パーティショニングルートのホスト）に接続し、DROP TABLE コマンドを呼び出します。例、sales テーブルを削除するには、次のコマンドを呼び出します。

```
DROP TABLE sales;
```

サーバーは、テーブルが削除されたことを確認します。

```
edb=# drop table sales;
DROP TABLE
edb=#
```

DROP TABLE コマンドの詳細については、次の PostgreSQL コア文書を参照してください。

< <https://www.postgresql.org/docs/current/ 静的/sql-droptable.html> >

3.13 ALTER TABLE...DROP PARTITION

</ div>

ALTER TABLE... DROP PARTITION コマンドを使用して、パーティション定義、およびそのパーティションに保存されているデータを削除します。構文は次のとおりです。

```
ALTER TABLE <table_name> DROP PARTITION <partition_name>;
```

パラメータ

`table_name`

パーティション化されたテーブルの名前（オプションでスキーマ修飾）。

`partition_name`

削除するパーティションの名前。

説明

`ALTER TABLE... DROP PARTITION` コマンドは、パーティションとそのパーティションに保存されているデータを削除します。`ALTER TABLE... DROP PARTITION` コマンドは、`LIST` または `RANGE` パーティションテーブルのパーティションを削除できます。このコマンドは、`HASH` パーティションテーブルでは機能しないことにノートしてください。パーティションを削除すると、そのパーティションのサブパーティションもすべて削除されます。

`DROP PARTITION` 句を使用するには、パーティショニングルートの所有者、テーブルを所有するグループのメンバであるか、データベーススーパーユーザまたは管理者権限が必要です。

3.13.1 Example - Deleting a Partition

</ div>

次の例では、`sales` テーブルのパーティションを削除します。次のコマンドを使用して、`sales` テーブルを作成します。

```

CREATE TABLE sales
(
  * dept_no      number,
  * part_no      varchar2,
  * country      varchar2(20),
  * date         date,
  * amount       number
)
PARTITION BY LIST(country)
(
  * PARTITION europe VALUES('FRANCE', 'ITALY'),
  * PARTITION asia  VALUES('INDIA', 'PAKISTAN'),
  * PARTITION americas VALUES('US', 'CANADA')
);

```

`ALL_TAB_PARTITIONS` ビューを照会すると、パーティション名が表示されます。

```

edb=# SELECT partition_name, high_value FROM ALL_TAB_PARTITIONS;
partition_name |      high_value
-----+-----
EUROPE         | 'FRANCE', 'ITALY'

```

(次のページに続く)

(前のページからの続き)

ASIA	'INDIA', 'PAKISTAN'
AMERICAS	'US', 'CANADA'

(3 rows)

sales テーブルから americas パーティションを削除するには、次のコマンドを呼び出します。

```
ALTER TABLE sales DROP PARTITION americas;
```

ALL_TAB_PARTITIONS ビューを照会すると、パーティションが正常に削除されたことがわかります。

```
edb=# SELECT partition_name, high_value FROM ALL_TAB_PARTITIONS;
```

partition_name	high_value
EUROPE	'FRANCE', 'ITALY'
ASIA	'INDIA', 'PAKISTAN'

(2 rows)

3.14 ALTER TABLE…DROP SUBPARTITION

</div>

ALTER TABLE… DROP SUBPARTITION コマンドを使用して、サブパーティション定義、およびそのサブパーティションに保存されているデータを削除します。構文は次のとおりです。

```
ALTER TABLE <table_name> DROP SUBPARTITION <subpartition_name>;
```

パラメータ

table_name

パーティション化されたテーブルの名前（オプションでスキーマ修飾）。

subpartition_name

削除するサブパーティションの名前。

説明

ALTER TABLE… DROP SUBPARTITION コマンドは、サブパーティションと、そのサブパーティションに保存されているデータを削除します。DROP SUBPARTITION 句を使用するには、パーティショニングルート所有者、テーブルを所有するグループのメンバ、またはスーパーユーザまたは管理者特権が必要です。

3.14.1 Example - Deleting a Subpartition

</ div>

次の例では、sales テーブルのサブパーティションを削除します。次のコマンドを使用して、sales テーブルを作成します。

```
CREATE TABLE sales
(
  * dept_no      number,
  * part_no      varchar2,
  * country      varchar2(20),
  * date         date,
  * amount       number
)
PARTITION BY RANGE(date)
  * SUBPARTITION BY LIST (country)
  * (
  *   PARTITION first_half_2012 VALUES LESS THAN('01-JUL-2012')
  *   (
  *     SUBPARTITION europe VALUES ('ITALY', 'FRANCE'),
  *     SUBPARTITION americas VALUES ('CANADA', 'US'),
  *     SUBPARTITION asia VALUES ('PAKISTAN', 'INDIA')
  *   ),
  *   PARTITION second_half_2012 VALUES LESS THAN('01-JAN-2013')
  * );
```

ALL_TAB_SUBPARTITIONS ビューを照会すると、サブパーティション名が表示されます。

```
edb=# SELECT subpartition_name, high_value FROM ALL_TAB_SUBPARTITIONS;
 subpartition_name |      high_value
-----+-----
EUROPE             | 'ITALY', 'FRANCE'
AMERICAS           | 'CANADA', 'US'
ASIA                | 'PAKISTAN', 'INDIA'
SYS0101            | DEFAULT
(4 rows)
```

sales テーブルから americas サブパーティションを削除するには、次のコマンドを呼び出します。

```
ALTER TABLE sales DROP SUBPARTITION americas;
```

ALL_TAB_SUBPARTITIONS ビューのクエリは、サブパーティションが正常に削除されたことを示しています。

```
edb=# SELECT subpartition_name, high_value FROM ALL_TAB_SUBPARTITIONS;
 subpartition_name |      high_value
-----+-----
```

(次のページに続く)

(前のページからの続き)

EUROPE	'ITALY', 'FRANCE'
ASIA	'PAKISTAN', 'INDIA'
SYS0101	DEFAULT

(3 rows)

3.15 TRUNCATE TABLE

</div>

TRUNCATE TABLE コマンドを使用して、テーブル定義を保持しながら、テーブルの内容を削除します。テーブルを切り捨てると、そのテーブルのパーティションまたはサブパーティションも切り捨てられます。構文は次のとおりです。

```
TRUNCATE TABLE <table_name>
```

パラメータ

table_name

パーティション化されたテーブルの名前（オプションでスキーマ修飾）。

説明

TRUNCATE TABLE コマンドは、テーブル全体と、そのテーブルにあるデータを削除します。テーブルを削除すると、そのテーブルのパーティションまたはサブパーティションも削除されます。

TRUNCATE TABLE コマンドを使用するには、パーティショニングルート所有者、テーブルを所有するグループのメンバ、スキーマ所有者、またはデータベーススーパーユーザである必要があります。

3.15.1 Example - Emptying a Table

</div>

次の例では、sales テーブルからデータを削除します。次のコマンドを使用して、sales テーブルを作成します。

```
CREATE TABLE sales
(
  * dept_no      number,
  * part_no      varchar2,
  * country      varchar2(20),
  * date         date,
  * amount       number
)
PARTITION BY LIST(country)
(
```

(次のページに続く)

(前のページからの続き)

```

* PARTITION europe VALUES('FRANCE', 'ITALY'),
* PARTITION asia VALUES('INDIA', 'PAKISTAN'),
* PARTITION americas VALUES('US', 'CANADA')
);

```

sales テーブルに次のコマンドを入力します。

```

INSERT INTO sales VALUES
  * (10, '4519b', 'FRANCE', '17-Jan-2012', '45000'),
  * (20, '3788a', 'INDIA', '01-Mar-2012', '75000'),
  * (40, '9519b', 'US', '12-Apr-2012', '145000'),
  * (20, '3788a', 'PAKISTAN', '04-Jun-2012', '37500'),
  * (40, '4577b', 'US', '11-Nov-2012', '25000'),
  * (30, '7588b', 'CANADA', '14-Dec-2012', '50000'),
  * (30, '9519b', 'CANADA', '01-Feb-2012', '75000'),
  * (30, '4519b', 'CANADA', '08-Apr-2012', '120000'),
  * (40, '3788a', 'US', '12-May-2012', '4950'),
  * (10, '9519b', 'ITALY', '07-Jul-2012', '15000'),
  * (10, '9519a', 'FRANCE', '18-Aug-2012', '650000'),
  * (10, '9519b', 'FRANCE', '18-Aug-2012', '650000'),
  * (20, '3788b', 'INDIA', '21-Sept-2012', '5090'),
  * (40, '4788a', 'US', '23-Sept-2012', '4950'),
  * (40, '4788b', 'US', '09-Oct-2012', '15000'),
  * (20, '4519a', 'INDIA', '18-Oct-2012', '650000'),
  * (20, '4519b', 'INDIA', '2-Dec-2012', '5090');

```

sales テーブルを照会すると、パーティションにデータが入力されていることがわかります。

```

edb=# SELECT tableoid::regclass, * FROM sales;

```

* tableoid	dept_no	part_no	country	date	amount
sales_americas	40	9519b	US	12-APR-12 00:00:00	145000
sales_americas	40	4577b	US	11-NOV-12 00:00:00	25000
sales_americas	30	7588b	CANADA	14-DEC-12 00:00:00	50000
sales_americas	30	9519b	CANADA	01-FEB-12 00:00:00	75000
sales_americas	30	4519b	CANADA	08-APR-12 00:00:00	120000
sales_americas	40	3788a	US	12-MAY-12 00:00:00	4950
sales_americas	40	4788a	US	23-SEP-12 00:00:00	4950
sales_americas	40	4788b	US	09-OCT-12 00:00:00	15000
sales_europe	10	4519b	FRANCE	17-JAN-12 00:00:00	45000
sales_europe	10	9519b	ITALY	07-JUL-12 00:00:00	15000
sales_europe	10	9519a	FRANCE	18-AUG-12 00:00:00	650000
sales_europe	10	9519b	FRANCE	18-AUG-12 00:00:00	650000
sales_asia	20	3788a	INDIA	01-MAR-12 00:00:00	75000
sales_asia	20	3788a	PAKISTAN	04-JUN-12 00:00:00	37500

(次のページに続く)

(前のページからの続き)

sales_asia		20		3788b		INDIA		21-SEP-12 00:00:00		5090
sales_asia		20		4519a		INDIA		18-OCT-12 00:00:00		650000
sales_asia		20		4519b		INDIA		02-DEC-12 00:00:00		5090

(17 rows)

sales テーブルの内容を削除するには、次のコマンドを呼び出します。

```
TRUNCATE TABLE sales;
```

これで、sales テーブルを照会すると、データは削除されましたが、構造はそのままです。

```
edb=# SELECT tableoid::regclass, * FROM sales;
 tableoid | dept_no | part_no | country | date | amount
-----+-----+-----+-----+-----+-----
(0 rows)
```

TRUNCATE TABLE コマンドの詳細については、次の PostgreSQL 文書を参照してください。

< <https://www.postgresql.org/docs/current/ 静的/sql-truncate.html> >

3.16 ALTER TABLE…TRUNCATE PARTITION

</ div>

ALTER TABLE… TRUNCATE PARTITION コマンドを使用して、指定されたパーティションからデータを削除し、パーティション構造をそのまま残します。構文は次のとおりです。

```
ALTER TABLE <table_name> TRUNCATE PARTITION <partition_name>
* [{DROP|REUSE} STORAGE]
```

パラメータ

table_name

パーティション化されたテーブルの名前（オプションでスキーマ修飾）。

partition_name

削除するパーティションの名前。

説明

ALTER TABLE… TRUNCATE PARTITION コマンドを使用して、指定されたパーティションからデータを削除し、パーティション構造をそのまま残します。パーティションを切り捨てると、そのパーティションのサブパーティションも切り捨てられます。

ALTER TABLE… TRUNCATE PARTITION は、テーブルに存在する可能性のある ON DELETE トリガーを起動しませんが、ON TRUNCATE トリガーを起動します。ON TRUNCATE トリガーがパーティションに定義され

ている場合、すべての BEFORE TRUNCATE トリガーは切り捨てが発生する前に起動され、すべての AFTER TRUNCATE トリガーは最後の切り捨てが発生した後に起動されます。

ALTER TABLE... TRUNCATE PARTITION を呼び出すには、テーブルに対する TRUNCATE 権限が必要です。

DROP STORAGE と REUSE STORAGE は互換性のためにのみ含まれています。句は解析され、無視されます。

3.16.1 Example - Emptying a Partition

</ div>

次の例では、sales テーブルのパーティションからデータを削除します。次のコマンドを使用して、sales テーブルを作成します。

```
CREATE TABLE sales
(
  * dept_no      number,
  * part_no      varchar2,
  * country      varchar2(20),
  * date         date,
  * amount       number
)
PARTITION BY LIST(country)
(
  * PARTITION europe VALUES('FRANCE', 'ITALY'),
  * PARTITION asia VALUES('INDIA', 'PAKISTAN'),
  * PARTITION americas VALUES('US', 'CANADA')
);
```

sales テーブルに次のコマンドを入力します。

```
INSERT INTO sales VALUES
  * (10, '4519b', 'FRANCE', '17-Jan-2012', '45000'),
  * (20, '3788a', 'INDIA', '01-Mar-2012', '75000'),
  * (40, '9519b', 'US', '12-Apr-2012', '145000'),
  * (20, '3788a', 'PAKISTAN', '04-Jun-2012', '37500'),
  * (40, '4577b', 'US', '11-Nov-2012', '25000'),
  * (30, '7588b', 'CANADA', '14-Dec-2012', '50000'),
  * (30, '9519b', 'CANADA', '01-Feb-2012', '75000'),
  * (30, '4519b', 'CANADA', '08-Apr-2012', '120000'),
  * (40, '3788a', 'US', '12-May-2012', '4950'),
  * (10, '9519b', 'ITALY', '07-Jul-2012', '15000'),
  * (10, '9519a', 'FRANCE', '18-Aug-2012', '650000'),
  * (10, '9519b', 'FRANCE', '18-Aug-2012', '650000'),
  * (20, '3788b', 'INDIA', '21-Sept-2012', '5090'),
  * (40, '4788a', 'US', '23-Sept-2012', '4950'),
  * (40, '4788b', 'US', '09-Oct-2012', '15000'),
```

(次のページに続く)

(前のページからの続き)

```
* (20, '4519a', 'INDIA', '18-Oct-2012', '650000'),
* (20, '4519b', 'INDIA', '2-Dec-2012', '5090');
```

sales テーブルを照会すると、パーティションにデータが入力されていることがわかります。

```
edb=# SELECT tableoid::regclass, * FROM sales;
*   tableoid | dept_no | part_no | country |          date          | amount
-----+-----+-----+-----+-----+-----
sales_americas |      40 | 9519b | US      | 12-APR-12 00:00:00 | 145000
sales_americas |      40 | 4577b | US      | 11-NOV-12 00:00:00 | 25000
sales_americas |      30 | 7588b | CANADA  | 14-DEC-12 00:00:00 | 50000
sales_americas |      30 | 9519b | CANADA  | 01-FEB-12 00:00:00 | 75000
sales_americas |      30 | 4519b | CANADA  | 08-APR-12 00:00:00 | 120000
sales_americas |      40 | 3788a | US      | 12-MAY-12 00:00:00 | 4950
sales_americas |      40 | 4788a | US      | 23-SEP-12 00:00:00 | 4950
sales_americas |      40 | 4788b | US      | 09-OCT-12 00:00:00 | 15000
sales_europe   |      10 | 4519b | FRANCE  | 17-JAN-12 00:00:00 | 45000
sales_europe   |      10 | 9519b | ITALY   | 07-JUL-12 00:00:00 | 15000
sales_europe   |      10 | 9519a | FRANCE  | 18-AUG-12 00:00:00 | 650000
sales_europe   |      10 | 9519b | FRANCE  | 18-AUG-12 00:00:00 | 650000
sales_asia     |      20 | 3788a | INDIA   | 01-MAR-12 00:00:00 | 75000
sales_asia     |      20 | 3788a | PAKISTAN | 04-JUN-12 00:00:00 | 37500
sales_asia     |      20 | 3788b | INDIA   | 21-SEP-12 00:00:00 | 5090
sales_asia     |      20 | 4519a | INDIA   | 18-OCT-12 00:00:00 | 650000
sales_asia     |      20 | 4519b | INDIA   | 02-DEC-12 00:00:00 | 5090
(17 rows)
```

americas パーティションの内容を削除するには、次のコマンドを呼び出します。

```
ALTER TABLE sales TRUNCATE PARTITION americas;
```

sales テーブルを照会すると、americas パーティションのコンテンツが削除されていることがわかります。

```
edb=# SELECT tableoid::regclass, * FROM sales;
*   tableoid | dept_no | part_no | country |          date          | amount
-----+-----+-----+-----+-----+-----
sales_europe   |      10 | 4519b | FRANCE  | 17-JAN-12 00:00:00 | 45000
sales_europe   |      10 | 9519b | ITALY   | 07-JUL-12 00:00:00 | 15000
sales_europe   |      10 | 9519a | FRANCE  | 18-AUG-12 00:00:00 | 650000
sales_europe   |      10 | 9519b | FRANCE  | 18-AUG-12 00:00:00 | 650000
sales_asia     |      20 | 3788a | INDIA   | 01-MAR-12 00:00:00 | 75000
sales_asia     |      20 | 3788a | PAKISTAN | 04-JUN-12 00:00:00 | 37500
sales_asia     |      20 | 3788b | INDIA   | 21-SEP-12 00:00:00 | 5090
sales_asia     |      20 | 4519a | INDIA   | 18-OCT-12 00:00:00 | 650000
sales_asia     |      20 | 4519b | INDIA   | 02-DEC-12 00:00:00 | 5090
(9 rows)
```

行が削除されている間、americas パーティションの構造はそのままです。

```
edb=# SELECT partition_name, high_value FROM ALL_TAB_PARTITIONS;
partition_name |      high_value
-----+-----
EUROPE         | 'FRANCE', 'ITALY'
ASIA           | 'INDIA', 'PAKISTAN'
AMERICAS       | 'US', 'CANADA'
(3 rows)
```

3.17 ALTER TABLE…TRUNCATE SUBPARTITION

</div>

ALTER TABLE… TRUNCATE SUBPARTITION コマンドを使用して、指定されたサブパーティションからすべてのデータを削除し、サブパーティション構造をそのまま残します。構文は次のとおりです。

```
ALTER TABLE <table_name>
  * TRUNCATE SUBPARTITION <subpartition_name>
  * [{DROP|REUSE} STORAGE]
```

パラメータ

table_name

パーティション化されたテーブルの名前（オプションでスキーマ修飾）。

subpartition_name

切り捨てられるサブパーティションの名前。

説明

ALTER TABLE… TRUNCATE SUBPARTITION コマンドは、指定されたサブパーティションからすべてのデータを削除し、サブパーティション構造をそのまま残します。

ALTER TABLE… TRUNCATE SUBPARTITION は、テーブルに存在する可能性のある ON DELETE トリガーを起動しませんが、ON TRUNCATE トリガーを起動します。ON TRUNCATE トリガーがサブパーティションに定義されている場合、すべての BEFORE TRUNCATE トリガーは切り捨てが発生する前に起動され、すべての AFTER TRUNCATE トリガーは最後の切り捨てが発生した後に起動されます。

ALTER TABLE… TRUNCATE SUBPARTITION を呼び出すには、テーブルに対する TRUNCATE 権限が必要です。

DROP STORAGE および REUSE STORAGE 句は、互換性のためにのみ含まれています。句は解析され、無視されます。

3.17.1 Example - Emptying a Subpartition

</div>

次の例では、sales テーブルのサブパーティションからデータを削除します。次のコマンドを使用して、sales テーブルを作成します。

```
CREATE TABLE sales
(
  * dept_no      number,
  * part_no      varchar2,
  * country      varchar2(20),
  * date         date,
  * amount       number
)
PARTITION BY RANGE(date) SUBPARTITION BY LIST (country)
(
  * PARTITION "2011" VALUES LESS THAN('01-JAN-2012')
  * (
  *   SUBPARTITION europe_2011 VALUES ('ITALY', 'FRANCE'),
  *   SUBPARTITION asia_2011 VALUES ('PAKISTAN', 'INDIA'),
  *   SUBPARTITION americas_2011 VALUES ('US', 'CANADA')
  * ),
  * PARTITION "2012" VALUES LESS THAN('01-JAN-2013')
  * (
  *   SUBPARTITION europe_2012 VALUES ('ITALY', 'FRANCE'),
  *   SUBPARTITION asia_2012 VALUES ('PAKISTAN', 'INDIA'),
  *   SUBPARTITION americas_2012 VALUES ('US', 'CANADA')
  * ),
  * PARTITION "2013" VALUES LESS THAN('01-JAN-2015')
  * (
  *   SUBPARTITION europe_2013 VALUES ('ITALY', 'FRANCE'),
  *   SUBPARTITION asia_2013 VALUES ('PAKISTAN', 'INDIA'),
  *   SUBPARTITION americas_2013 VALUES ('US', 'CANADA')
  * )
);
```

sales テーブルに次のコマンドを入力します。

```
INSERT INTO sales VALUES
  * (10, '4519b', 'FRANCE', '17-Jan-2011', '45000'),
  * (20, '3788a', 'INDIA', '01-Mar-2012', '75000'),
  * (40, '9519b', 'US', '12-Apr-2012', '145000'),
  * (20, '3788a', 'PAKISTAN', '04-Jun-2012', '37500'),
  * (40, '4577b', 'US', '11-Nov-2012', '25000'),
  * (30, '7588b', 'CANADA', '14-Dec-2011', '50000'),
  * (30, '4519b', 'CANADA', '08-Apr-2012', '120000'),
```

(次のページに続く)

(前のページからの続き)

```
* (40, '3788a', 'US', '12-May-2011', '4950'),
* (20, '3788a', 'US', '04-Apr-2012', '37500'),
* (40, '4577b', 'INDIA', '11-Jun-2011', '25000'),
* (10, '9519b', 'ITALY', '07-Jul-2012', '15000'),
* (20, '4519b', 'INDIA', '2-Dec-2012', '5090');
```

sales テーブルを照会すると、行がサブパーティションに分散されていることがわかります。

```
edb=# SELECT tableoid::regclass, * FROM sales;
   *   tableoid           | dept_no | part_no | country |          date          | amount
-----+-----+-----+-----+-----+-----
sales_americas_2011|      30 | 7588b  | CANADA  | 14-DEC-11 00:00:00 | 50000
sales_americas_2011|      40 | 3788a  | US      | 12-MAY-11 00:00:00 | 4950
sales_europe_2011  |      10 | 4519b  | FRANCE  | 17-JAN-11 00:00:00 | 45000
sales_asia_2011    |      40 | 4577b  | INDIA   | 11-JUN-11 00:00:00 | 25000
sales_americas_2012|      40 | 9519b  | US      | 12-APR-12 00:00:00 | 145000
sales_americas_2012|      40 | 4577b  | US      | 11-NOV-12 00:00:00 | 25000
sales_americas_2012|      30 | 4519b  | CANADA  | 08-APR-12 00:00:00 | 120000
sales_americas_2012|      20 | 3788a  | US      | 04-APR-12 00:00:00 | 37500
sales_europe_2012  |      10 | 9519b  | ITALY   | 07-JUL-12 00:00:00 | 15000
sales_asia_2012    |      20 | 3788a  | INDIA   | 01-MAR-12 00:00:00 | 75000
sales_asia_2012    |      20 | 3788a  | PAKISTAN| 04-JUN-12 00:00:00 | 37500
sales_asia_2012    |      20 | 4519b  | INDIA   | 02-DEC-12 00:00:00 | 5090
(12 rows)
```

2012_americas パーティションの内容を削除するには、次のコマンドを呼び出します。

```
ALTER TABLE sales TRUNCATE SUBPARTITION "americas_2012";
```

sales テーブルを照会すると、americas_2012 パーティションのコンテンツが削除されていることがわかります。

```
edb=# SELECT tableoid::regclass, * FROM sales;
   tableoid           | dept_no | part_no | country |          date          | amount
-----+-----+-----+-----+-----+-----
sales_americas_2011|      30 | 7588b  | CANADA  | 14-DEC-11 00:00:00 | 50000
sales_americas_2011|      40 | 3788a  | US      | 12-MAY-11 00:00:00 | 4950
sales_europe_2011  |      10 | 4519b  | FRANCE  | 17-JAN-11 00:00:00 | 45000
sales_asia_2011    |      40 | 4577b  | INDIA   | 11-JUN-11 00:00:00 | 25000
sales_europe_2012  |      10 | 9519b  | ITALY   | 07-JUL-12 00:00:00 | 15000
sales_asia_2012    |      20 | 3788a  | INDIA   | 01-MAR-12 00:00:00 | 75000
sales_asia_2012    |      20 | 3788a  | PAKISTAN| 04-JUN-12 00:00:00 | 37500
sales_asia_2012    |      20 | 4519b  | INDIA   | 02-DEC-12 00:00:00 | 5090
(8 rows)
```

行が削除されている間、2012_americas パーティションの構造はそのままです。

```
edb=# SELECT subpartition_name, high_value FROM ALL_TAB_SUBPARTITIONS;
subpartition_name |      high_value
-----+-----
EUROPE_2011       | 'ITALY', 'FRANCE'
ASIA_2011         | 'PAKISTAN', 'INDIA'
AMERICAS_2011     | 'US', 'CANADA'
EUROPE_2012       | 'ITALY', 'FRANCE'
ASIA_2012         | 'PAKISTAN', 'INDIA'
AMERICAS_2012     | 'US', 'CANADA'
EUROPE_2013       | 'ITALY', 'FRANCE'
ASIA_2013         | 'PAKISTAN', 'INDIA'
AMERICAS_2013     | 'US', 'CANADA'
(9 rows)
```

3.18 Accessing a PARTITION or SUBPARTITION

</ div>

パーティション表の場合、PARTITION part_name または SUBPARTITION subpart_name を使用してパーティションまたはサブパーティションにアクセスできます。次の例では、date によって範囲パーティション化され、country 列によるリストパーティショニングを使用してサブパーティション化されたパーティションテーブル sales を作成します。

```
CREATE TABLE sales
(
  * * dept_no      number,
  * * part_no      varchar2,
  * * country      varchar2(20),
  * * date         date,
  * * amount       number
)
PARTITION BY RANGE(date)
  * * SUBPARTITION BY LIST(country)
(
  * * PARTITION q1_2020
  * *   VALUES LESS THAN('2020-Apr-01')
  * *   (
  * *     SUBPARTITION q1_europe VALUES ('FRANCE', 'ITALY'),
  * *     SUBPARTITION q1_asia VALUES ('INDIA', 'PAKISTAN'),
  * *     SUBPARTITION q1_americas VALUES ('US', 'CANADA')
  * *   ),
  * * PARTITION q2_2020
  * *   VALUES LESS THAN('2020-Jul-01')
  * *   (
```

(次のページに続く)

(前のページからの続き)

```

* *      SUBPARTITION q2_europe VALUES ('FRANCE', 'ITALY'),
* *      SUBPARTITION q2_asia VALUES ('INDIA', 'PAKISTAN'),
* *      SUBPARTITION q2_americas VALUES ('US', 'CANADA')
* *      )
);

```

SELECT ステートメントは2つのパーティションを示しています。各パーティションには3つのサブパーティションがあります。

```

edb=# SELECT subpartition_name, high_value, partition_name FROM ALL_TAB_SUBPARTITIONS;
subpartition_name |      high_value      | partition_name
-----+-----+-----
Q1_EUROPE         | 'FRANCE', 'ITALY'    | Q1_2020
Q1_ASIA           | 'INDIA', 'PAKISTAN'  | Q1_2020
Q1_AMERICAS       | 'US', 'CANADA'       | Q1_2020
Q2_EUROPE         | 'FRANCE', 'ITALY'    | Q2_2020
Q2_ASIA           | 'INDIA', 'PAKISTAN'  | Q2_2020
Q2_AMERICAS       | 'US', 'CANADA'       | Q2_2020
(6 rows)

```

次の INSERT ステートメントは、特定の PARTITION part_name または SUBPARTITION subpart_name を使用して、sales テーブルに行を挿入します。

```

INSERT INTO sales PARTITION (q1_2020) VALUES (10, 'q1_2020', 'FRANCE', '2020-Feb-01',
→ '500000');
INSERT INTO sales PARTITION (q2_2020) VALUES (10, 'q2_2020', 'ITALY', '2020-Apr-01',
→ '550000');
INSERT INTO sales SUBPARTITION (q1_europe) VALUES (10, 'q1_europe', 'FRANCE', '2020-
→ Feb-01', '600000');
INSERT INTO sales SUBPARTITION (q2_asia) VALUES (10, 'q2_asia', 'INDIA', '2020-Apr-01
→ ', '650000');

edb=# SELECT tableoid::regclass, * FROM sales ORDER BY date;
*   tableoid   | dept_no | part_no | country |      date      | amount
-----+-----+-----+-----+-----+-----
sales_q1_europe |      10 | q1_2020 | FRANCE  | 01-FEB-20 00:00:00 | 500000
sales_q1_europe |      10 | q1_europe | FRANCE  | 01-FEB-20 00:00:00 | 600000
sales_q2_europe |      10 | q2_2020 | ITALY   | 01-APR-20 00:00:00 | 550000
sales_q2_asia   |      10 | q2_asia  | INDIA   | 01-APR-20 00:00:00 | 650000
(4 rows)

```

次のクエリーを使用して、特定のパーティション q1_2020 またはサブパーティション q1_europe から値をフェッチします。

```

edb=# SELECT tableoid::regclass, * FROM sales PARTITION (q1_2020) ORDER BY date;

```

(次のページに続く)

(前のページからの続き)

```

*   tableoid      | dept_no | part_no | country |      date      | amount
-----+-----+-----+-----+-----+-----
sales_q1_europe |      10 | q1_2020 | FRANCE | 01-FEB-20 00:00:00 | 500000
sales_q1_europe |      10 | q1_europe | FRANCE | 01-FEB-20 00:00:00 | 600000
(2 rows)

edb=# SELECT tableoid::regclass, country FROM sales SUBPARTITION (q1_europe) ORDER BY
↪date;
*   tableoid      | country
-----+-----
sales_q1_europe | FRANCE
sales_q1_europe | FRANCE
(2 rows)

```

次の SELECT ステートメントは、キーワード PARTITION または SUBPARTITION を指定して、特定のパーティションまたはパーティションテーブルのサブパーティションから行を選択します。

```

edb=# SELECT tableoid::regclass, * FROM sales PARTITION (q2_2020) ORDER BY country;
*   tableoid      | dept_no | part_no | country |      date      | amount
-----+-----+-----+-----+-----+-----
sales_q2_asia    |      10 | q2_asia | INDIA   | 01-APR-20 00:00:00 | 650000
sales_q2_europe  |      10 | q2_2020 | ITALY   | 01-APR-20 00:00:00 | 550000
(2 rows)

edb=# SELECT tableoid::regclass, date FROM sales SUBPARTITION (q2_asia) ORDER BY
↪country;
*   tableoid      |      date
-----+-----
sales_q2_asia    | 01-APR-20 00:00:00
(1 row)

```

次の UPDATE ステートメントは、sales テーブルのパーティションまたはサブパーティションの値を更新します。

```

edb=# UPDATE sales PARTITION (q1_2020) SET amount = 10000 WHERE amount = 500000
↪RETURNING tableoid::regclass, *;
*   tableoid      | dept_no | part_no | country |      date      | amount
-----+-----+-----+-----+-----+-----
sales_q1_europe |      10 | q1_2020 | FRANCE | 01-FEB-20 00:00:00 | 10000
(1 row)

UPDATE 1

edb=# UPDATE sales SUBPARTITION (q1_europe) SET amount = 5000 WHERE amount = 600000
↪RETURNING tableoid::regclass, *;

```

(次のページに続く)

(前のページからの続き)

* tableoid	dept_no	part_no	country	date	amount
-----+-----+-----+-----+-----+-----					
sales_q1_europe	10	q1_europe	FRANCE	01-FEB-20 00:00:00	5000
(1 row)					
UPDATE 1					

次の DELETE ステートメントは、sales テーブルのパーティション q2_2020 またはサブパーティション q2_asia から行を削除します。

```
edb=# DELETE FROM sales PARTITION (q2_2020) WHERE amount = 550000 RETURNING ↵
↵tableoid::regclass, *;
```

* tableoid	dept_no	part_no	country	date	amount
-----+-----+-----+-----+-----+-----					
sales_q2_europe	10	q2_2020	ITALY	01-APR-20 00:00:00	550000
(1 row)					

DELETE 1

```
edb=# DELETE FROM sales SUBPARTITION (q2_asia) RETURNING tableoid::regclass, *;
```

* tableoid	dept_no	part_no	country	date	amount
-----+-----+-----+-----+-----+-----					
sales_q2_asia	10	q2_asia	INDIA	01-APR-20 00:00:00	650000
(1 row)					

DELETE 1

3.18.1 PARTITION または SUBPARTITION へのアクセスにエイリアスを使用する

ALIASES を SELECT、INSERT、UPDATE、または DELETE ステートメントとともに使用して、パーティションまたはサブパーティションにアクセスできます。次の例では、date によって範囲パーティション化され、country 列によるリストパーティショニングを使用してサブパーティション化されたパーティションテーブル sales を作成します。

```
CREATE TABLE sales
(
  * * dept_no      number,
  * * part_no      varchar2,
  * * country      varchar2(20),
  * * date         date,
  * * amount       number
)
PARTITION BY RANGE(date)
  * * SUBPARTITION BY LIST(country)
```

(次のページに続く)

(前のページからの続き)

```
(
* * PARTITION q1_2020
* *   VALUES LESS THAN('2020-Apr-01')
* *   (
* *     SUBPARTITION q1_europe VALUES ('FRANCE', 'ITALY'),
* *     SUBPARTITION q1_asia VALUES ('INDIA', 'PAKISTAN'),
* *     SUBPARTITION q1_americas VALUES ('US', 'CANADA')
* *   ),
* * PARTITION q2_2020
* *   VALUES LESS THAN('2020-Jul-01')
* *   (
* *     SUBPARTITION q2_europe VALUES ('FRANCE', 'ITALY'),
* *     SUBPARTITION q2_asia VALUES ('INDIA', 'PAKISTAN'),
* *     SUBPARTITION q2_americas VALUES ('US', 'CANADA')
* *   )
);
```

SELECT ステートメントは2つのパーティションを示しています。各パーティションには3つのサブパーティションがあります。

```
edb=# SELECT subpartition_name, high_value, partition_name FROM ALL_TAB_SUBPARTITIONS;
subpartition_name |      high_value      | partition_name
-----+-----+-----
Q1_EUROPE         | 'FRANCE', 'ITALY'    | Q1_2020
Q1_ASIA           | 'INDIA', 'PAKISTAN'  | Q1_2020
Q1_AMERICAS       | 'US', 'CANADA'       | Q1_2020
Q2_EUROPE         | 'FRANCE', 'ITALY'    | Q2_2020
Q2_ASIA           | 'INDIA', 'PAKISTAN'  | Q2_2020
Q2_AMERICAS       | 'US', 'CANADA'       | Q2_2020
(6 rows)
```

次の INSERT ステートメントは、sales テーブルの別名を作成し、パーティション q1_2020 と q2_2020 またはサブパーティション q1_europe と q1_asia に行を挿入します。

```
INSERT INTO sales PARTITION (q1_2020) AS q1_sales VALUES (10, 'q1_2020', 'FRANCE',
→ '2020-Feb-01', '500000');
INSERT INTO sales PARTITION (q2_2020) q2_sales (q2_sales.dept_no, q2_sales.part_no,
→ q2_sales.country, q2_sales.date, q2_sales.amount) VALUES (20, 'q2_2020', 'ITALY',
→ '2020-Apr-01', '550000');
INSERT INTO sales SUBPARTITION (q1_europe) AS sales_q1_europe VALUES (10, 'q1_europe',
'FRANCE', '2020-Feb-01', '600000');
INSERT INTO sales SUBPARTITION (q1_asia) sales_q1_asia (sales_q1_asia.dept_no, sales_
→ q1_asia.part_no, sales_q1_asia.country, sales_q1_asia.date, sales_q1_asia.amount)
→ VALUES (20, 'q1_asia', 'INDIA', '2020-Mar-01', '650000');
```

次の SELECT ステートメントは、sales テーブルの特定のパーティションまたはサブパーティションから

行を選択します。

```
edb=# SELECT tableoid::regclass, * FROM sales PARTITION (q1_2020) AS q1_sales ORDER_
↳BY country;
   *   tableoid      | dept_no | part_no | country |          date          | amount
-----+-----+-----+-----+-----+-----+-----
sales_q1_europe |      10 | q1_2020 | FRANCE | 01-FEB-20 00:00:00 | 500000
sales_q1_europe |      10 | q1_europe | FRANCE | 01-FEB-20 00:00:00 | 600000
sales_q1_asia   |      20 | q1_asia | INDIA  | 01-MAR-20 00:00:00 | 650000
(3 rows)
```

```
edb=# SELECT tableoid::regclass, * FROM sales SUBPARTITION (q1_europe) sales_q1_
↳europe ORDER BY country;
   *   tableoid      | dept_no | part_no | country |          date          | amount
-----+-----+-----+-----+-----+-----+-----
sales_q1_europe |      10 | q1_2020 | FRANCE | 01-FEB-20 00:00:00 | 500000
sales_q1_europe |      10 | q1_europe | FRANCE | 01-FEB-20 00:00:00 | 600000
(2 rows)
```

次の UPDATE ステートメントは、sales テーブルのパーティションまたはサブパーティションの値を更新します。

```
edb=# UPDATE sales PARTITION (q1_2020) AS q1_sales SET q1_sales.amount = 10000 WHERE_
↳q1_sales.amount = 500000;
UPDATE 1

edb=# UPDATE sales SUBPARTITION (q1_europe) sales_q1_europe SET sales_q1_europe.
↳amount = 5000 WHERE sales_q1_europe.amount = 600000;
UPDATE 1
```

次の DELETE ステートメントは、sales テーブルのパーティション q1_2020 またはサブパーティション q1_europe から行を削除します。

```
edb=# DELETE FROM sales PARTITION (q1_2020) q1_sales WHERE q1_sales.amount = 10000;
DELETE 1

edb=# DELETE FROM sales SUBPARTITION (q1_europe) AS sales_q1_europe WHERE sales_q1_
↳europe.amount = 5000;
DELETE 1
```

第4章 Handling Stray Values in a LIST or RANGE Partitioned Table

</div>

DEFAULT または MAXVALUE のパーティションまたはサブパーティションは、テーブルに定義された他のパーティショニングルールを満たさない行をキャプチャします。

デフォルトパーティションの定義

DEFAULT パーティションは、LIST パーティション（またはサブパーティション）テーブル内の他のパーティションに収まらない行をキャプチャします。DEFAULT ルールを含めない場合、パーティショニングコンストレインの値のいずれともマッチしない行はエラーになります。各 LIST パーティションまたはサブパーティションには、独自の DEFAULT ルールがあります。

DEFAULT ルールの構文は次のとおりです。

```
PARTITION [<partition_name>] VALUES (DEFAULT)
```

partition_name は、他のパーティションに指定されたルールにマッチしない行を格納するパーティションまたはサブパーティションの名前を指定します。

最後の例では、サーバーが country 列の値に基づいてデータを保存するパーティションを決定するリストパーティションテーブルを作成しました。country 列の値にルールにリストされていない値が含まれる行を追加しようとすると、Advanced Server はエラーを報告します。

```
edb=# INSERT INTO sales VALUES
edb-# (40, '3000x', 'IRELAND', '01-Mar-2012', '45000');
ERROR: no partition of relation "sales_2012" found for row
DETAIL: Partition key of the failing row contains (country) = (IRELAND).
```

次の例では、同じテーブルを作成しますが、DEFAULT パーティションを追加します。サーバーは、europe, asia、または others パーティションの americas パーティションのパーティショニングルールで指定された値とマッチしない行を格納します。

```
CREATE TABLE sales
(
  * dept_no      number,
  * part_no      varchar2,
  * country      varchar2(20),
  * date         date,
  * amount       number
```

(次のページに続く)

(前のページからの続き)

```

)
PARTITION BY LIST(country)
(
  * PARTITION europe VALUES('FRANCE', 'ITALY'),
  * PARTITION asia VALUES('INDIA', 'PAKISTAN'),
  * PARTITION americas VALUES('US', 'CANADA'),
  * PARTITION others VALUES (DEFAULT)
);

```

DEFAULT パーティションをテストするには、パーティショニングコンストレインで指定された国のいずれともマッチしない値を `country` 行に追加します。

```

INSERT INTO sales VALUES
  * (40, '3000x', 'IRELAND', '01-Mar-2012', '45000');

```

`sales` テーブルの内容を照会すると、以前に拒否された行が `sales_others` パーティションに保存されていることが確認されます。

```

edb=# SELECT tableoid::regclass, * FROM sales;

```

* tableoid	dept_no	part_no	country	date	amount
sales_americas	40	9519b	US	12-APR-12 00:00:00	145000
sales_americas	40	4577b	US	11-NOV-12 00:00:00	25000
sales_americas	30	7588b	CANADA	14-DEC-12 00:00:00	50000
sales_americas	30	9519b	CANADA	01-FEB-12 00:00:00	75000
sales_americas	30	4519b	CANADA	08-APR-12 00:00:00	120000
sales_americas	40	3788a	US	12-MAY-12 00:00:00	4950
sales_americas	40	4788a	US	23-SEP-12 00:00:00	4950
sales_americas	40	4788b	US	09-OCT-12 00:00:00	15000
sales_europe	10	4519b	FRANCE	17-JAN-12 00:00:00	45000
sales_europe	10	9519b	ITALY	07-JUL-12 00:00:00	15000
sales_europe	10	9519a	FRANCE	18-AUG-12 00:00:00	650000
sales_europe	10	9519b	FRANCE	18-AUG-12 00:00:00	650000
sales_asia	20	3788a	INDIA	01-MAR-12 00:00:00	75000
sales_asia	20	3788a	PAKISTAN	04-JUN-12 00:00:00	37500
sales_asia	20	3788b	INDIA	21-SEP-12 00:00:00	5090
sales_asia	20	4519a	INDIA	18-OCT-12 00:00:00	650000
sales_asia	20	4519b	INDIA	02-DEC-12 00:00:00	5090
sales_others	40	3000x	IRELAND	01-MAR-12 00:00:00	45000

(18 rows)

Advanced Server は、DEFAULT パーティションまたはサブパーティションのコンテンツを再割り当てするために次のメソッドを提供します。

- `ALTER TABLE... ADD PARTITION` コマンドを使用して、テーブル内の既存の行と追加するパーティションの値の間に矛盾する値がない限り、DEFAULT ルールを使用してパーティションをテーブルに追

加できます。または、ALTER TABLE… SPLIT PARTITION コマンドを使用して、既存のパーティションを分割できます。この簡条書きリストの後に例を示します。

- ALTER TABLE… ADD SUBPARTITION コマンドを使用して、テーブル内の既存の行と追加するサブパーティションの値の間に矛盾する値がない限り、DEFAULT ルールを使用してテーブルにサブパーティションを追加できます。または、ALTER TABLE… SPLIT SUBPARTITION コマンドを使用して、既存のサブパーティションを分割できます。

デフォルトパーティションを持つテーブルへのパーティションの追加

このセクションの冒頭に示した CREATE TABLE sales コマンドで作成されたテーブルを使用して、以下は、テーブル内の既存の行と追加されるパーティションの値の間に値の競合がないと仮定した ALTER TABLE… ADD PARTITION コマンドの使用を示しています。

```
edb=# ALTER TABLE sales ADD PARTITION africa values ('SOUTH AFRICA', 'KENYA');
ALTER TABLE
```

ただし、次の行がテーブルに挿入されたときに値が競合する場合、次のエラーが表示されます。

```
edb=# INSERT INTO sales (dept_no, country) VALUES
(1, 'FRANCE'), (2, 'INDIA'), (3, 'US'), (4, 'SOUTH AFRICA'), (5, 'NEPAL');
INSERT 0 5
```

行 (4, 'SOUTH AFRICA') は、ALTER TABLE… ADD PARTITION ステートメントの VALUES リストと競合するため、エラーが発生します。

```
edb=# ALTER TABLE sales ADD PARTITION africa values ('SOUTH AFRICA', 'KENYA');
ERROR:  updated partition constraint for default partition "sales_others"
would be violated by some row
```

デフォルトパーティションの分割

次の例では、DEFAULT パーティションを分割し、2つの新しいパーティション間でパーティションのコンテンツを再配布します。テーブルは、このセクションの冒頭に示した CREATE TABLE sales コマンドで作成されました。

以下は、DEFAULT パーティションへの行を含む行をテーブルに挿入します。

```
INSERT INTO sales VALUES
* (10, '4519b', 'FRANCE', '17-Jan-2012', '45000'),
* (10, '9519b', 'ITALY', '07-Jul-2012', '15000'),
* (20, '3788a', 'INDIA', '01-Mar-2012', '75000'),
* (20, '3788a', 'PAKISTAN', '04-Jun-2012', '37500'),
* (30, '9519b', 'US', '12-Apr-2012', '145000'),
* (30, '7588b', 'CANADA', '14-Dec-2012', '50000'),
* (40, '4519b', 'SOUTH AFRICA', '08-Apr-2012', '120000'),
* (40, '4519b', 'KENYA', '08-Apr-2012', '120000'),
* (50, '3788a', 'CHINA', '12-May-2012', '4950');
```

パーティションには DEFAULT others パーティションが含まれます。

```
edb=# SELECT partition_name, high_value FROM ALL_TAB_PARTITIONS;
partition_name |      high_value
-----+-----
EUROPE         | 'FRANCE', 'ITALY'
ASIA           | 'INDIA', 'PAKISTAN'
AMERICAS       | 'US', 'CANADA'
OTHERS         | DEFAULT
(4 rows)
```

以下は、パーティションに分散された行を示しています。

```
edb=# SELECT tableoid::regclass, * FROM sales;
tableoid      | dept_no | part_no | country      |      date      | amount
-----+-----+-----+-----+-----+-----
sales_americas |      30 | 9519b   | US           | 12-APR-12 00:00:00 | 145000
sales_americas |      30 | 7588b   | CANADA       | 14-DEC-12 00:00:00 | 50000
sales_europe   |      10 | 4519b   | FRANCE       | 17-JAN-12 00:00:00 | 45000
sales_europe   |      10 | 9519b   | ITALY        | 07-JUL-12 00:00:00 | 15000
sales_asia     |      20 | 3788a   | INDIA        | 01-MAR-12 00:00:00 | 75000
sales_asia     |      20 | 3788a   | PAKISTAN     | 04-JUN-12 00:00:00 | 37500
sales_others   |      40 | 4519b   | SOUTH AFRICA | 08-APR-12 00:00:00 | 120000
sales_others   |      40 | 4519b   | KENYA        | 08-APR-12 00:00:00 | 120000
sales_others   |      50 | 3788a   | CHINA        | 12-MAY-12 00:00:00 | 4950
(9 rows)
```

次のコマンドは、DEFAULT others パーティションを africa と others 名前付け 2 つのパーティションに分割します。

```
ALTER TABLE sales SPLIT PARTITION others VALUES
* ('SOUTH AFRICA', 'KENYA')
* INTO (PARTITION africa, PARTITION others);
```

パーティションには、DEFAULT others パーティションとともに africa パーティションが含まれるようになりました。

```
edb=# SELECT partition_name, high_value FROM ALL_TAB_PARTITIONS;
partition_name |      high_value
-----+-----
EUROPE         | 'FRANCE', 'ITALY'
ASIA           | 'INDIA', 'PAKISTAN'
AMERICAS       | 'US', 'CANADA'
AFRICA         | 'SOUTH AFRICA', 'KENYA'
OTHERS         | DEFAULT
(5 rows)
```

以下は、新しいパーティションに行が再配布されたことを示しています。

```
edb=# SELECT tableoid::regclass, * FROM sales;
 tableoid      |dept_no | part_no |   country   |      date      | amount
-----+-----+-----+-----+-----+-----
sales_americas |      30 | 9519b   |    US       | 12-APR-12 00:00:00 | 145000
sales_americas |      30 | 7588b   |   CANADA    | 14-DEC-12 00:00:00 |  50000
sales_europe   |      10 | 4519b   |   FRANCE    | 17-JAN-12 00:00:00 |  45000
sales_europe   |      10 | 9519b   |    ITALY    | 07-JUL-12 00:00:00 |  15000
sales_asia     |      20 | 3788a   |    INDIA    | 01-MAR-12 00:00:00 |  75000
sales_asia     |      20 | 3788a   |  PAKISTAN   | 04-JUN-12 00:00:00 |  37500
sales_africa   |      40 | 4519b   | SOUTH AFRICA| 08-APR-12 00:00:00 | 120000
sales_africa   |      40 | 4519b   |    KENYA    | 08-APR-12 00:00:00 | 120000
sales_others_1 |      50 | 3788a   |    CHINA    | 12-MAY-12 00:00:00 |   4950
(9 rows)
```

**** MAXVALUE パーティションの定義****

MAXVALUE パーティション（またはサブパーティション）は、範囲パーティション（またはサブパーティション）テーブル内の他のパーティションに適合しない行をキャプチャします。MAXVALUE ルールを含めない場合、パーティショニングルールで指定された最大制限を超える行はエラーになります。各パーティションまたはサブパーティションには、独自の MAXVALUE パーティションがあります。

MAXVALUE ルールの構文は次のとおりです。

```
PARTITION [<partition_name>] VALUES LESS THAN (MAXVALUE)
```

partition_name は、他のパーティションに指定されたルールにマッチしない行を格納するパーティションの名前を指定します。

最後の例では、date 列の値に基づいてデータがパーティション化された範囲パーティションテーブルを作成しました。パーティショニングコンストレインにリストされている日付を超える date を持つ行を追加しようとすると、Advanced Server はエラーを報告します。

```
edb=# INSERT INTO sales VALUES
edb-#   (40, '3000x', 'IRELAND', '01-Mar-2013', '45000');
ERROR:  no partition of relation "sales" found for row
DETAIL:  Partition key of the failing row contains (date) = (01-MAR-13 00:00:00).
```

次の CREATE TABLE コマンドは、同じテーブルを作成しますが、MAXVALUE パーティションを使用します。サーバーはエラーをスローする代わりに、others パーティションの以前のパーティショニングコンストレインにマッチしない行を保存します。

```
CREATE TABLE sales
(
  * dept_no      number,
  * part_no      varchar2,
  * country      varchar2(20),
  * date         date,
  * amount       number
```

(次のページに続く)

(前のページからの続き)

```

)
PARTITION BY RANGE(date)
(
  * PARTITION q1_2012 VALUES LESS THAN('2012-Apr-01'),
  * PARTITION q2_2012 VALUES LESS THAN('2012-Jul-01'),
  * PARTITION q3_2012 VALUES LESS THAN('2012-Oct-01'),
  * PARTITION q4_2012 VALUES LESS THAN('2013-Jan-01'),
  * PARTITION others VALUES LESS THAN (MAXVALUE)
);

```

MAXVALUE パーティションをテストするには、パーティショニングルールにリストされている最後の日付値を超える値を持つ行を `date` 列に追加します。サーバーは、行を `others` パーティションに格納します。

```

INSERT INTO sales VALUES
  * (40, '3000x', 'IRELAND', '01-Mar-2013', '45000');

```

`sales` テーブルの内容を照会すると、以前に拒否された行が `sales_others` パーティションに保存されていることが確認されます。

```

edb=# SELECT tableoid::regclass, * FROM sales;

```

* tableoid	dept_no	part_no	country	date	amount
sales_q1_2012	10	4519b	FRANCE	17-JAN-12 00:00:00	45000
sales_q1_2012	20	3788a	INDIA	01-MAR-12 00:00:00	75000
sales_q1_2012	30	9519b	CANADA	01-FEB-12 00:00:00	75000
sales_q2_2012	40	9519b	US	12-APR-12 00:00:00	145000
sales_q2_2012	20	3788a	PAKISTAN	04-JUN-12 00:00:00	37500
sales_q2_2012	30	4519b	CANADA	08-APR-12 00:00:00	120000
sales_q2_2012	40	3788a	US	12-MAY-12 00:00:00	4950
sales_q3_2012	10	9519b	ITALY	07-JUL-12 00:00:00	15000
sales_q3_2012	10	9519a	FRANCE	18-AUG-12 00:00:00	650000
sales_q3_2012	10	9519b	FRANCE	18-AUG-12 00:00:00	650000
sales_q3_2012	20	3788b	INDIA	21-SEP-12 00:00:00	5090
sales_q3_2012	40	4788a	US	23-SEP-12 00:00:00	4950
sales_q4_2012	40	4577b	US	11-NOV-12 00:00:00	25000
sales_q4_2012	30	7588b	CANADA	14-DEC-12 00:00:00	50000
sales_q4_2012	40	4788b	US	09-OCT-12 00:00:00	15000
sales_q4_2012	20	4519a	INDIA	18-OCT-12 00:00:00	650000
sales_q4_2012	20	4519b	INDIA	02-DEC-12 00:00:00	5090
sales_others	40	3000x	IRELAND	01-MAR-13 00:00:00	45000

(18 rows)

Advanced Server には、MAXVALUE パーティションまたはサブパーティションのコンテンツを再割り当てする方法がないことにノートしてください。

- ALTER TABLE... ADD PARTITION ステートメントを使用して MAXVALUE ルールでパーティションを

テーブルに追加することはできませんが、ALTER TABLE… SPLIT PARTITION ステートメントを使用して既存のパーティションを分割することはできます。

- ALTER TABLE… ADD SUBPARTITION ステートメントを使用して MAXVALUE ルールでサブパーティションをテーブルに追加することはできませんが、既存のサブパーティションを ALTER TABLE… SPLIT SUBPARTITION ステートメントで分割できます。

第5章 Specifying Multiple Partitioning Keys in a RANGE Partitioned Table

</div>

多くの場合、RANGE パーティションテーブルにマルチプルのキー列を指定することにより、パフォーマンスを改善できます。少数の列セットで比較演算子を使用して（より大きい値またはより小さい値に基づいて）行を選択することが多い場合は、RANGE パーティショニングルールでそれらの列を使用することを検討してください。

範囲パーティションテーブルで複数のキーを指定する

範囲パーティションテーブル定義には、パーティショニングキーにマルチプルの列が含まれる場合があります。範囲パーティションテーブルにマルチプルのパーティショニングキーを指定するには、PARTITION BY RANGE 句の後にカンマ区切りのリストで列名を含めます。

```
CREATE TABLE sales
(
  * dept_no      number,
  * part_no      varchar2,
  * country      varchar2(20),
  * sale_year     number,
  * sale_month    number,
  * sale_day      number,
  * amount       number
)
PARTITION BY RANGE(sale_year, sale_month)
(
  * PARTITION q1_2012
  *   VALUES LESS THAN(2012, 4),
  * PARTITION q2_2012
  *   VALUES LESS THAN(2012, 7),
  * PARTITION q3_2012
  *   VALUES LESS THAN(2012, 10),
  * PARTITION q4_2012
  *   VALUES LESS THAN(2013, 1)
);
```

テーブルがマルチプルのパーティショニングキーで作成されている場合、パーティションのプルーニングを最大限に活用するには、テーブルを照会するときにマルチプルのキー値を指定する必要があります。

```
edb=# EXPLAIN SELECT * FROM sales WHERE sale_year = 2012 AND sale_month = 8;
*
                                QUERY PLAN
-----
Append  (cost=0.00..14.35 rows=1 width=250)
*  ->  Seq Scan on sales_q3_2012  (cost=0.00..14.35 rows=1 width=250)
*      Filter: ((sale_year = '2012'::numeric) AND (sale_month = '8'::numeric))
(3 rows)
```

sale_month 列の値が 8 で、sale_year 列の値が 2012 であるすべての行が q3_2012 パーティションに格納されるため、Advanced Server はそのパーティションのみを検索します。

第6章 Retrieving Information about a Partitioned Table

</div>

Advanced Server は、テーブルパーティションの構造に関する情報をビューするために使用できる 5 つのシステムカタログビューを提供します。

パーティション Views のクエリ

次のビューをクエリーして、パーティションテーブルおよびサブパーティションテーブルに関する情報を取得できます。

- ALL_PART_TABLES
- ALL_TAB_PARTITIONS
- ALL_TAB_SUBPARTITIONS
- ALL_PART_KEY_COLUMNS
- ALL_SUBPART_KEY_COLUMNS

各ビューの構造は [Table Partitioning Views - Reference](#) で説明されています。EDB-PSQL クライアントを使用している場合、次のように入力してビューの構造を検出することもできます。

\d <view_name>

view_name は、テーブルパーティショニングビューの名前を指定します。

ビューのクエリは、パーティションテーブルまたはサブパーティションテーブルの構造に関する情報を提供できます。例、次のコードスニペットは、サブパーティションテーブルの名前を表示します。

```
edb=# SELECT subpartition_name, partition_name FROM ALL_TAB_SUBPARTITIONS;
 subpartition_name | partition_name
-----+-----
EUROPE_2011       | EUROPE
EUROPE_2012       | EUROPE
ASIA_2011          | ASIA
ASIA_2012          | ASIA
AMERICAS_2011     | AMERICAS
AMERICAS_2012     | AMERICAS
(6 rows)
```

6.1 Table Partitioning Views - Reference

</ div>

次のカタログビュー（オラクルデータベースと互換性あり）を照会して、テーブルパーティションに関する詳細情報を確認します。

6.1.1 ALL_PART_TABLES

</ div>

次の表に、ALL_PART_TABLES ビューで使用可能な情報を示します。

Column	Type	Description
<i>owner</i>	<i>name</i>	テーブルの所有者。
<i>schema_name</i>	<i>name</i>	テーブルが存在するスキーマ。
<i>table_name</i>	<i>name</i>	テーブルの名前。
<i>partitioning_type</i>	<i>text</i>	RANGE、LIST、または HASH。
<i>subpartitioning_type</i>	<i>text</i>	RANGE、LIST、HASH、または NONE。
<i>partition_count</i>	<i>bigint</i>	パーティションの数。
<i>def_subpartition_count</i>	<i>integer</i>	デフォルトのサブパーティション数-これは常に 0 になります。
<i>partitioning_key_count</i>	<i>integer</i>	パーティション句にリストされている列の数。
<i>subpartitioning_key_count</i>	<i>integer</i>	subpartitionby 句の列数。
<i>status</i>	<i>character varying(8)</i>	この列は常に有効です。
<i>def_tablespace_name</i>	<i>character varying(30)</i>	この列は常に NULL になります。
<i>def_pct_free</i>	<i>numeric</i>	この列は常に NULL になります。
<i>def_pct_used</i>	<i>numeric</i>	この列は常に NULL になります。
<i>def_ini_trans</i>	<i>numeric</i>	この列は常に NULL になります。
<i>def_max_trans</i>	<i>numeric</i>	この列は常に NULL になります。
<i>def_initial_extent</i>	<i>character varying(40)</i>	この列は常に NULL になります。
<i>def_next_extent</i>	<i>character varying(40)</i>	この列は常に NULL になります。
<i>def_min_extents</i>	<i>character varying(40)</i>	この列は常に NULL になります。
<i>def_max_extents</i>	<i>character varying(40)</i>	この列は常に NULL になります。
<i>def_pct_increase</i>	<i>character varying(40)</i>	この列は常に NULL になります。
<i>def_freelists</i>	<i>numeric</i>	この列は常に NULL になります。
<i>def_freelist_groups</i>	<i>numeric</i>	この列は常に NULL になります。
<i>def_logging</i>	<i>character varying(7)</i>	この列は常に YES になります。
<i>def_compression</i>	<i>character varying(8)</i>	この列は常に NONE になります。
<i>def_buffer_pool</i>	<i>character varying(7)</i>	この列は常に DEFAULT になります。
<i>ref_ptn_constraint_name</i>	<i>character varying(30)</i>	この列は常に NULL になります。
<i>interval</i>	<i>character varying(1000)</i>	この列は常に NULL になります。

6.1.2 ALL_TAB_PARTITIONS

</ div>

次の表に、ALL_TAB_PARTITIONS ビューで使用可能な情報を示します。

Column	Type	Description
<i>table_owner</i>	<i>name</i>	テーブルの所有者。
<i>schema_name</i>	<i>name</i>	テーブルが存在するスキーマ。
<i>table_name</i>	<i>name</i>	テーブルの名前。
<i>composite</i>	<i>text</i>	テーブルがサブパーティション化されている場合は YES。サブパーティション化されていない場合は NO。
<i>partition_name</i>	<i>name</i>	パーティションの名前。
<i>subpartition_count</i>	<i>bigint</i>	このパーティションのサブパーティションの数。
<i>high_value</i>	<i>text</i>	'CREATE TABLE' ステートメントで指定された高いパーティショニング値。
<i>high_value_length</i>	<i>integer</i>	高いパーティショニング値の長さ。
<i>partition_position</i>	<i>integer</i>	このパーティションの通常的位置。
<i>tablespace_name</i>	<i>name</i>	このパーティションが存在するテーブルスペース。
<i>pct_free</i>	<i>numeric</i>	この列は常に 0 になります。
<i>pct_used</i>	<i>numeric</i>	この列は常に 0 になります。
<i>ini_trans</i>	<i>numeric</i>	この列は常に 0 になります。
<i>max_trans</i>	<i>numeric</i>	この列は常に 0 になります。
<i>initial_extent</i>	<i>numeric</i>	この列は常に NULL になります。
<i>next_extent</i>	<i>numeric</i>	この列は常に NULL になります。
<i>min_extent</i>	<i>numeric</i>	この列は常に 0 になります。
<i>max_extent</i>	<i>numeric</i>	この列は常に 0 になります。
<i>pct_increase</i>	<i>numeric</i>	この列は常に 0 になります。
<i>freelists</i>	<i>numeric</i>	この列は常に NULL になります。
<i>freelist_groups</i>	<i>numeric</i>	この列は常に NULL になります。
<i>logging</i>	<i>character varying(7)</i>	この列は常に YES になります。
<i>compression</i>	<i>character varying(8)</i>	この列は常に NONE になります。
<i>num_rows</i>	<i>numeric</i>	約。このパーティションの行数。
<i>blocks</i>	<i>integer</i>	約。このパーティション内のブロック数。
<i>empty_blocks</i>	<i>numeric</i>	この列は常に NULL になります。
<i>avg_space</i>	<i>numeric</i>	この列は常に NULL になります。
<i>chain_cnt</i>	<i>numeric</i>	この列は常に NULL になります。
<i>avg_row_len</i>	<i>numeric</i>	この列は常に NULL になります。
<i>sample_size</i>	<i>numeric</i>	この列は常に NULL になります。
<i>last_analyzed</i>	<i>timestamp without time zone</i>	この列は常に NULL になります。
<i>buffer_pool</i>	<i>character varying(7)</i>	この列は常に NULL になります。
<i>global_stats</i>	<i>character varying(3)</i>	この列は常に YES になります。
<i>user_stats</i>	<i>character varying(3)</i>	この列は常に NO になります。

次のページに続く

表 2 – 前のページからの続き

Column	Type	Description
<i>backing_table</i>	<i>regclass</i>	このパーティションのバックアップテーブルの OID。

6.1.3 ALL_TAB_SUBPARTITIONS

</ div>

次の表に、ALL_TAB_SUBPARTITIONS ビューで使用可能な情報を示します。

Column	Type	Description
<i>table_owner</i>	<i>name</i>	テーブルの所有者の名前。
<i>schema_name</i>	<i>name</i>	テーブルが存在するスキーマの名前。
<i>table_name</i>	<i>name</i>	テーブルの名前。
<i>partition_name</i>	<i>name</i>	パーティションの名前。
<i>subpartition_name</i>	<i>name</i>	サブパーティションの名前。
<i>high_value</i>	<i>text</i>	'CREATE TABLE' ステートメントで指定された高いパーティショニング値。
<i>high_value_length</i>	<i>integer</i>	高いパーティショニング値の長さ。
<i>subpartition_position</i>	<i>integer</i>	このサブパーティションの通常の位置。
<i>tablespace_name</i>	<i>name</i>	このサブパーティションが存在するテーブルスペース。
<i>pct_free</i>	<i>numeric</i>	この列は常に 0 になります。
<i>pct_used</i>	<i>numeric</i>	この列は常に 0 になります。
<i>ini_trans</i>	<i>numeric</i>	この列は常に 0 になります。
<i>max_trans</i>	<i>numeric</i>	この列は常に 0 になります。
<i>initial_extent</i>	<i>numeric</i>	この列は常に NULL になります。
<i>next_extent</i>	<i>numeric</i>	この列は常に NULL になります。
<i>min_extent</i>	<i>numeric</i>	この列は常に 0 になります。
<i>max_extent</i>	<i>numeric</i>	この列は常に 0 になります。
<i>pct_increase</i>	<i>numeric</i>	この列は常に 0 になります。
<i>freelists</i>	<i>numeric</i>	この列は常に NULL になります。
<i>freelist_groups</i>	<i>numeric</i>	この列は常に NULL になります。
<i>logging</i>	<i>character varying(7)</i>	この列は常に YES になります。
<i>compression</i>	<i>character varying(8)</i>	この列は常に NONE になります。
<i>num_rows</i>	<i>numeric</i>	約。このサブパーティションの行数。
<i>blocks</i>	<i>integer</i>	約。このサブパーティション内のブロックの数。
<i>empty_blocks</i>	<i>numeric</i>	この列は常に NULL になります。
<i>avg_space</i>	<i>numeric</i>	この列は常に NULL になります。
<i>chain_cnt</i>	<i>numeric</i>	この列は常に NULL になります。
<i>avg_row_len</i>	<i>numeric</i>	この列は常に NULL になります。
<i>sample_size</i>	<i>numeric</i>	この列は常に NULL になります。

次のページに続く

表 3 – 前のページからの続き

Column	Type	Description
<i>last_analyzed</i>	<i>timestamp without time zone</i>	この列は常に NULL になります。
<i>buffer_pool</i>	<i>character varying(7)</i>	この列は常に NULL になります。
<i>global_stats</i>	<i>character varying(3)</i>	この列は常に YES になります。
<i>user_stats</i>	<i>character varying(3)</i>	この列は常に NO になります。
<i>backing_table</i>	<i>regclass</i>	このサブパーティションのバックアップテーブルの OID。

6.1.4 ALL_PART_KEY_COLUMNS

</ div>

次の表に、ALL_PART_KEY_COLUMNS ビューで使用可能な情報を示します。

Column	Type	Description
<i>owner</i>	<i>name</i>	テーブル所有者の名前。
<i>schema_name</i>	<i>name</i>	テーブルが存在するスキーマの名前。
<i>name</i>	<i>name</i>	テーブルの名前。
<i>object_type</i>	<i>character(5)</i>	この列は常に `TABLE` になります。
<i>column_name</i>	<i>name</i>	パーティショニングキー列の名前。
<i>column_position</i>	<i>integer</i>	パーティショニングキー内のこの列の位置（最初の列の列位置は 1、2 番目の列の列位置は 2 ...）

6.1.5 ALL_SUBPART_KEY_COLUMNS

</ div>

次の表に、ALL_SUBPART_KEY_COLUMNS ビューで使用可能な情報を示します。

Column	Type	Description
<i>owner</i>	<i>name</i>	テーブル所有者の名前。
<i>schema_name</i>	<i>name</i>	テーブルが存在するスキーマの名前。
<i>name</i>	<i>name</i>	テーブルの名前。
<i>object_type</i>	<i>character(5)</i>	この列は常に `TABLE` になります。
<i>column_name</i>	<i>name</i>	パーティショニングキー列の名前。
<i>column_position</i>	<i>integer</i>	サブパーティションキー内のこの列の位置（最初の列の列位置は 1、2 番目の列の列位置は 2 ...）