



EDB Postgres Advanced Server

リリース **13**

Database Compatibility Table Partitioning Guide

2020 年 12 月 31 日

目次

第 1 章	はじめに	1
第 2 章	Selecting a Partition Type	3
2.1	Interval Range Partitioning	4
2.2	Automatic List Partitioning	5
第 3 章	Using Partition Pruning	7
3.1	Example - Partition Pruning	10
第 4 章	Partitioning Commands Compatible with Oracle Databases	13
4.1	CREATE TABLE...PARTITION BY	13
4.1.1	Example - PARTITION BY LIST	19
4.1.2	Example - AUTOMATIC LIST PARTITION	20
4.1.3	Example - PARTITION BY RANGE	21
4.1.4	Example - INTERVAL RANGE PARTITION	22
4.1.5	Example - PARTITION BY HASH	23
4.1.6	Example - PARTITION BY HASH...PARTITIONS num...	24
4.1.7	Example - PARTITION BY HASH...PARTITIONS num...STORE IN	25
4.1.8	Example - HASH/RANGE PARTITIONS num...	26
4.1.9	Example - LIST/HASH SUBPARTITIONS num...	26
4.1.10	Example - HASH/HASH PARTITIONS num... SUBPARTITIONS num...	27
4.1.11	Example - HASH/HASH SUBPARTITIONS num... STORE IN	28
4.1.12	Example - HASH/HASH PARTITIONS num ...STORE IN SUBPARTITIONS num... STORE IN	29
4.1.13	Example - RANGE/HASH SUBPARTITIONS num...	30
4.1.14	Example - RANGE/HASH SUBPARTITIONS num... IN PARTITION DESCRIPTION . . .	31
4.1.15	Example - LIST/HASH SUBPARTITIONS num STORE IN... IN PARTITION DESCRIPTION	32
4.1.16	Example - LIST/HASH STORE IN...TABLESPACES	33
4.1.17	Example - PARTITION BY RANGE, SUBPARTITION BY LIST	34
4.2	ALTER TABLE...ADD PARTITION	36
4.2.1	Example - Adding a Partition to a LIST Partitioned Table	38
4.2.2	Example - Adding a Partition to a RANGE Partitioned Table	39
4.2.3	Example - Adding a Partition with SUBPARTITIONS num...IN PARTITION DESCRIPTION	40

4.2.4	Example - Adding a Partition with SUBPARTITIONS num...	41
4.2.5	Example - Adding a Partition with SUBPARTITIONS num... STORE IN	43
4.3	ALTER TABLE...ADD SUBPARTITION	45
4.3.1	Example - Adding a Subpartition to a LIST/RANGE Partitioned Table	46
4.3.2	Example - Adding a Subpartition to a RANGE/LIST Partitioned Table	48
4.4	ALTER TABLE...SPLIT PARTITION	50
4.4.1	Example - Splitting a LIST Partition	52
4.4.2	Example - Splitting a RANGE Partition	54
4.4.3	Example - Splitting a RANGE/LIST Partition	56
4.4.4	Example - Splitting a Partition with SUBPARTITIONS num...	58
4.4.5	Example - Splitting a Partition with SUBPARTITIONS num...STORE IN	59
4.5	ALTER TABLE...SPLIT SUBPARTITION	62
4.5.1	Example - Splitting a LIST Subpartition	64
4.5.2	Example - Splitting a RANGE Subpartition	66
4.6	ALTER TABLE...EXCHANGE PARTITION	70
4.6.1	Example - Exchanging a Table for a Partition	71
4.7	ALTER TABLE...MOVE PARTITION	74
4.7.1	Example - Moving a Partition to a Different Tablespace	75
4.8	ALTER TABLE...RENAME PARTITION	77
4.8.1	Example - Renaming a Partition	78
4.9	ALTER TABLE...SET INTERVAL	79
4.9.1	Example - Setting an Interval Range Partition	79
4.10	ALTER TABLE...SET [PARTITIONING] AUTOMATIC	81
4.10.1	Example - Setting an AUTOMATIC List Partition	81
4.11	ALTER TABLE...SET SUBPARTITION TEMPLATE	83
4.11.1	Example - Setting a SUBPARTITION TEMPLATE	84
4.11.2	Example - Resetting a SUBPARTITION TEMPLATE	85
4.12	DROP TABLE	87
4.13	ALTER TABLE...DROP PARTITION	88
4.13.1	Example - Deleting a Partition	88
4.14	ALTER TABLE...DROP SUBPARTITION	90
4.14.1	Example - Deleting a Subpartition	90
4.15	TRUNCATE TABLE	92
4.15.1	Example - Emptying a Table	92
4.16	ALTER TABLE...TRUNCATE PARTITION	95
4.16.1	Example - Emptying a Partition	95
4.17	ALTER TABLE...TRUNCATE SUBPARTITION	98
4.17.1	Example - Emptying a Subpartition	98

第 6 章	Specifying Multiple Partitioning Keys in a RANGE Partitioned Table	111
第 7 章	Retrieving Information about a Partitioned Table	113
7.1	Table Partitioning Views - Reference	114
7.1.1	ALL_PART_TABLES	114
7.1.2	ALL_TAB_PARTITIONS	116
7.1.3	ALL_TAB_SUBPARTITIONS	117
7.1.4	ALL_PART_KEY_COLUMNS	118
7.1.5	ALL_SUBPART_KEY_COLUMNS	119
第 8 章	Conclusion	121
索引		123

第 1 章

はじめに

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、テーブル全体に散在するインデックスおよび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、バルクオペレーションよりもはるかに高速です。また、大量の DELETE によって引き起こされる VACUUM のオーバーヘッドも完全に回避します。
- めったに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。親指の良いルールは、テーブルのサイズは、データベースサーバの物理的メモリを超えなければならないことです。

このドキュメントでは、AdvancedServer 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。

注釈: このドキュメントで、この章で提示し、特にパーティショニング>宣言パーティショニングは、オラクルと互換性のテーブルパーティショニングに加えて、AdvancedServer10 にサポートされていることを PostgreSQL>10 注で導入されました*宣言型のパーティショニング* 機能は、説明しませんこの章で説明されているとおり。宣言的なパーティショニングについては、<https://www.postgresql.org/docs/current/静的/ddl-partitioning.html> で入手可能な PostgreSQL コア文書を参照してください。

PostgreSQL9.6 の 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ます。

第 2 章

Selecting a Partition Type

パーティション表を作成するときは、LIST、RANGE または HASH パーティショニング規則を指定します。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。新しい行がパーティションテーブルに追加されると、サーバは、各行が含まれているべきパーティションを決定するためにパーティショニング・ルールを使用します。

AdvancedServer 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ブレーニングを実施し、ユーザ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。

List Partitioning

リストパーティションテーブルを作成するとき、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、サーバはパーティショニングルールで指定されたキー値を行内の対応する列と比較しキー。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パーティションに格納されます。

注釈: 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。

Range Partitioning

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。列の値は、パーティショニングルールで指定された条件を満たした場合、行はルールで名前付けパーティションに格納されます。

Hash Partitioning

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。もしハッシュパーティションテーブルに行を追加すると、サーバは、ハッシュに応じてパーティション内の行を指定された列のデータ

(または列) のハッシュ値を計算し、記憶します。

注釈: AdvancedServer 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。

Subpartitioning

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プ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。

2.1 Interval Range Partitioning

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。

インターバルを1か月に設定し、現在の移行ポイントの2か月後にデータを挿入すると、その月のパーティションのみが作成され、介在するパーティションは作成されません。あなたは2019年5月10日のために、4月15日のために、必要なパーティション5月15日にデータを挿入しようとした場合例、あなたは、今、毎月のインターバルと2月15日、2019年の現在の遷移点とのインターバルレンジパーティション表を作成することができます、2019が作成され、データがそのパーティションに挿入されます。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ます。

間隔範囲パーティション構文の詳細については、[*CREATETABLE...PARTITIONBY*](#) を参照してください。

Restrictions on Interval Range Partitioning

INTERVAL 句には次の制限が適用されます。

- 間隔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することはできません。
- 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対して定義できません。
- 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。
- 区間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表のパーティションは唯一のオーダーで作成されます。

2.2 Automatic List Partitioning

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値の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 CALIFORNIA と FLORIDA が含まれている sales_state 列で sales 名前付けのテーブルを検討してください。sales_state の各値は、州ごとの売上の増加とともに増加します。新しい状態（例、INDIANA および OHIO）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、[CREATETABLE...PARTITIONBY](#) を参照してください。

Restrictions on Automatic List Partitioning

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は、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。
- コンポジット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 3 章

Using Partition Pruning

AdvancedServer 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:

- リストパーティションテーブル、WHERE 句が等号 (=) 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
- 範囲パーティションテーブル、パーティションプルーニングは、WHERE 句が等号 (=)、小より小さい (<)、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ときに有効です。
- 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。

- 高速剪定
- 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。

高速の剪定と制約による除外の違いは、制約による除外にはないながら、高速剪定は、オラクル・パーティション表内のパーティション間のリレーションを理解していることです。例、高速な制約による除外は、各*パーティション* 用に定義されたコンストレインを検討しなければならないが、特定のパーティションは、その値を保持することができることを理由を剪定リスト・パーティション表内の特定の値のクエリーを検索、。計画プロセスの早い段階で高速プルーニングが行われ、プランナが考慮しなければならない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後半で行われます。

Using Constraint Exclusion

`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そのパーティションにど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。

Fast Pruning

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）句を含むクエリーのみを最適化し、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、AdvancedServer は特定のクエリーを高速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。

注釈: 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ーを最適化することも、複数の列でパーティション最適化範囲テーブルパーティションに対するクエリー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、AdvancedServer は、パーティショニング列をリテラル値に制約する WHERE 句を含むクエリーを高速にブルーニングできます。例、リスト・パーティションテーブルのような与えられました:

```
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、高速プルーニングはパーティション `europa`, `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、高速プルーニングはパーティション `americas`, `europa` および `asia` 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 句に指定する演算子は、等号 (=) またはパーティショニング列のデータタイプに適した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、AdvancedServer は、パーティショニング列をリテラル値に制限する WHERE 句を含むクエリを高速にプ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。

```
>
```

```
>=
```

```
=
```

```
<=
```

```
<
```

また、高速プルーニングは、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)
)
```

高速プルーニングは、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、高速プルーニングでは、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）。

注釈: 高速プルーニングは、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。

3.1 Example - Partition Pruning

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、AdvancedServer 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')
  ),
)
```

(次のページに続く)

(前のページからの続き)

```

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 4 章

Partitioning Commands Compatible with Oracle Databases

次のセクションでは、AdvancedServer 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。

4.1 CREATE TABLE...PARTITION BY

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。

List Partitioning Syntax

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> ]...)]
  (<list_partition_definition>[, <list_partition_definition>]...);
```

Where list_partition_definition is:

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

Range Partitioning Syntax

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> ]...)]
  (<range_partition_definition>[, <range_partition_definition>]...);
```

Where range_partition_definition is:

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

Hash Partitioning Syntax

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> ]...)]
  [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>, ...)]
```

Subpartitioning Syntax

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>]...)
```

(次のページに続く)

(前のページからの続き)

```

[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>]... ) ]

```

Description

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します。DEFAULT キーワードの使用方法の詳細については、[LIST または RANGE パーティションテーブルでの浮遊値の処理](#) 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、[自動リストパーティション分割](#) 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は、二つの演算子を有するデータタイプの少なくとも一つの列を含まなければならない（すなわち、より大きな、より演算子に等しく、および小なり演算子 OR）。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。

レンジ・パーティション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、範囲パーティションテーブルの上部境界を超える値を持つ行を格納する 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 キーワードの使用方法の詳細については、[LIST または RANGE パーティションテーブルでの浮遊値の処理](#) 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 句を指定することにより、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、[間隔範囲分割](#) 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。HASH パーティションを指定する場合、列（または列の組み合わせ）を選択するヘルプ、できるだけ一意の近くにある、データが均等にパーティション間で分散されていることを保証。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の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。

注釈: AdvancedServer 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 句が LIST を指定している場合、DEFAULT サブパーティション。
- SUBPARTITION BY 句が RANGE を指定している場合、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。

Parameters

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、制約情報。

<https://www.postgresql.org/docs/current/静的/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および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。

注釈: 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。何 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。

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格納されます。

constant | expression

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

(value[, value]...)

使用テーブルエントリがパーティションにグループ化されることにより引用リテラル値（またはリテラル値のカンマ区切りリスト）を指定する value。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。

4.1.1 Example - PARTITION BY LIST

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');
```

4.1.2 Example - AUTOMATIC LIST PARTITION

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、データをテーブルに追加します。

```
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。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
-----+-----+-----
SALES      | P_CAL         | 'CALIFORNIA'
```

(次のページに続く)

(前のページからの続き)

SALES	P_FLO	'FLORIDA'
SALES	SYS106900103	'INDIANA'
SALES	SYS106900104	'OHIO'

(4 rows)

4.1.3 Example - PARTITION BY RANGE

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 列の値を持つ行は、q1_2012 名前付けのパーティションに格納されます。
- 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格納されます。
- 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格納され

ます。

- 2013 年 1 月 1 日より前の date 列の値を持つ行は、q4_2012 名前付けの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文を評価し、q3_2012 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');
```

4.1.4 Example - INTERVAL RANGE PARTITION

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。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)
```

4.1.5 Example - PARTITION BY HASH

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
);
```

テーブルには、part_no 列で指定されたハッシュ・パーティション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              |
 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   |
```

(次のページに続く)

(前のページからの続き)

```

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。

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

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
)
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

```

(次のページに続く)

(前のページからの続き)

```

SALES      | SYS0103
SALES      | SYS0104
SALES      | SYS0105
SALES      | SYS0106
SALES      | SYS0107
SALES      | SYS0108
(8 rows)

```

4.1.7 Example - PARTITION BY HASH...PARTITIONS num...STORE IN

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。作成され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が指定されます。

```

CREATE TABLE sales
(
  dept_no      number,
  part_no      varchar2,
  country      varchar2(20),
  date         date,
  amount       number
)
PARTITION BY HASH (part_no) 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
SALES      | SYS0104       | TS1
SALES      | SYS0105       | TS2
(5 rows)

```

4.1.8 Example - 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 によってハッシュ分割され、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)
```

4.1.9 Example - LIST/HASH SUBPARTITIONS num...

次の例は、country によってリストパーティション分割され、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分割されたテーブル (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
)
```

(次のページに続く)

(前のページからの続き)

```

)
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)

```

4.1.10 Example - 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)
```

4.1.11 Example - 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
)
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に格納され、サブパーティションは表領域 (ts1) および (ts2) に格納されます。

```
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;
```

(次のページに続く)

(前のページからの続き)

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)

4.1.12 Example - 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
)
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
-----+-----+-----+-----
```

(次のページに続く)

(前のページからの続き)

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)						

注釈: パーティションおよびサブパーティションに `STORE IN` 句が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。

4.1.13 Example - 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,
    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` 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
-----+-----+-----
```

(次のページに続く)

(前のページからの続き)

SALES	P1	Q1_EUROPE
SALES	P2	SYS0101
SALES	P2	SYS0102

(3 rows)

4.1.14 Example - 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、パーティション記述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 に対して 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)
```

4.1.15 Example - 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、パーティション記述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
SALES      | P1             | SYS0103           | TS1
SALES      | P2             | SYS0104           | TS2
SALES      | P2             | SYS0105           | TS2
(5 rows)
```

4.1.16 Example - 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 でリストパーティション表 (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、5 つのサブパーティションを持つ p2 名前付けの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
-----+-----+-----+-----
```

(次のページに続く)

(前のページからの続き)

SALES	P2	SYS0105	TS1
SALES	P2	SYS0106	TS1
SALES	P2	SYS0107	TS1
SALES	P2	SYS0108	TS1
SALES	P2	SYS0109	TS1

(5 rows)

4.1.17 Example - PARTITION BY RANGE, SUBPARTITION BY LIST

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')
    ),
  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('2012-Dec-01')
    (
      SUBPARTITION q4_europe VALUES ('FRANCE', 'ITALY'),
      SUBPARTITION q4_asia VALUES ('INDIA', 'PAKISTAN'),
      SUBPARTITION q4_americas VALUES ('US', 'CANADA')
    )
);
```

(次のページに続く)

(前のページからの続き)

```

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。サーバーが値のマッチを見つけると、行は対応するサブパーティションに保存されます。

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');

```

4.2 ALTER TABLE...ADD PARTITION

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。構文は次の構文です。

```
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>]...)
[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>]... ) ]
```

Description

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参照する必要があるリファレンス。

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、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 ルールを使用してテーブルにパーティションを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 パーティションが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。

Parameters

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および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]...)

使用は、行がパーティションに分配される引用リテラル値（またはリテラル値のカンマ区切りリスト）を指定する value。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、[LIST または RANGE パーティションテーブルでの浮遊値の処理](#) 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。何 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。

4.2.1 Example - Adding a Partition to a LIST Partitioned Table

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),
  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'
```

(次のページに続く)

(前のページからの続き)

```
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)
```

4.2.2 Example - Adding a Partition to a RANGE Partitioned Table

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,
  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)
```

4.2.3 Example - Adding a Partition with SUBPARTITIONS num...IN PARTITION DESCRIPTION

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て、あなたは、サブパーティションの指定された番号を追加する SUBPARTITIONS 句を指定することができます。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')
```

(次のページに続く)

(前のページからの続き)

);

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名前付けていないので、システム生成の名前が割り当てられています。

```
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      | 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 と新しく追加された2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      | AMERICAS       | SYS0107
SALES      | AMERICAS       | SYS0108
SALES      | ASIA           | SYS0103
SALES      | ASIA           | SYS0104
SALES      | EUROPE         | SYS0101
SALES      | EUROPE         | SYS0102
(6 rows)
```

4.2.4 Example - Adding a Partition with SUBPARTITIONS num...

次の例では、パーティションのサブパーティションを3からなるリスト・パーティションテーブル sales を加算します。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,
```

(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 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 および asia 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 を呼び出した後、パーティション 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)

```

4.2.5 Example - Adding a Partition with SUBPARTITIONS num... 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3つのパーティション (`americas`, `asia` および `europe`) 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;
 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、それらを (`ts1`) 名前付けのテーブルスペースに格納します。

```
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した後、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)

4.3 ALTER TABLE...ADD SUBPARTITION

ALTER TABLE... ADD SUBPARTITION コマンドは、既存のサブパーティションパーティションにサブパーティションを追加します。構文は次の構文です。

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

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]
```

Description

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参照する必要があるリファレンス。

テーブル内の既存の行と追加され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。

範囲サブパーティション化されたテーブルレンジの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・サブパーティションが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。

Parameters

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]...)

使用テーブルエントリがパーティションにグループ化されることにより引用リテラル値（またはリテラル値のカンマ区切りリスト）を指定する value。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、[LIST または RANGE パーティションテーブルでの浮遊値の処理](#) 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。

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

次の例では、リストパーティション化された sales テーブルに RANGE サブ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,
```

(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 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、europe, asia および 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'
AMERICAS        | AMERICAS_2012     | '01-JAN-13 00:00:00'
(7 rows)
```

注釈: 新しいレンジの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化ルールは、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。

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

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')
  ),
```

(次のページに続く)

(前のページからの続き)

```

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 名前付けの 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'
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)

```

4.4 ALTER TABLE...SPLIT PARTITION

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、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。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>]... ) ]
  );
```

Description

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。テーブルが持つことができる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、AdvancedServer 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表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。

Parameters

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および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および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インによって new_part1 に送信されない行を受け取ります。

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・ルールは、二つの演算子（すなわち、より大きい、等しいより-又は演算子に、および小なり演算子）を有するデータタイプの少なくとも一つの列を含んでいなければなりません。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]...)

使用は、行がパーティションに分配される引用リテラル値（またはリテラル値のカンマ区切りリスト）を指定する `value`。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、[LIST または RANGE パーティションテーブルでの浮遊値の処理](#) 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ています。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。何 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。

4.4.1 Example - Splitting a LIST Partition

この例つの新しいパーティションにリスト・パーティション `sales` テーブルのパーティションの 1 つを分割し、それらの間のパーティションの内容を再配布します。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
)
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'),
```

(次のページに続く)

(前のページからの続き)

```
(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
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 名前付けの 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)					

4.4.2 Example - Splitting a RANGE Partition

この例つのパーティションに（レンジ・パーティション sales テーブルの）q4_2012 パーティションを分割し、パーティションの内容を再配布します。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。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');
```

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 名前付けの 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、行が新しいパーティションに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_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_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)

4.4.3 Example - Splitting a RANGE/LIST Partition

次の例では、新しく作成されたパーティション内のオリジナルの表のパーティショニングを維持し、それらの間のパーティションの内容を再配布、新しいパーティションにレンジ・パーティション sales テーブルのパーティションの 1 つを分割します。

```
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)
  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 が含まれています。

```

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 名前付けの 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'

(次のページに続く)

(前のページからの続き)

SALES	Q1_2012_P2	SYS0110	'INDIA', 'PAKISTAN'
-------	------------	---------	---------------------

(6 rows)

4.4.4 Example - Splitting a Partition with SUBPARTITIONS num...

次の例では、新しく作成されたパーティション内のオリジナルの表のパーティショニングを維持し、それらの間のパーティションの内容を再配布、新しいパーティションにリスト・パーティション sales テーブルのパーティションの 1 つを分割します。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 名前付けの 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)

4.4.5 Example - Splitting a Partition with SUBPARTITIONS num...STORE IN

次の例では、新しく作成されたパーティション内のオリジナルの表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再配布する、2 つのパーティションにリスト・パーティション sales テーブルの europe パーティションを分割します。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名が割り当てられた 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;
 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 名前付けの 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
```

(次のページに続く)

(前のページからの続き)

SALES		FRANCE		SYS0120		TS1
SALES		FRANCE		SYS0121		TS1
SALES		FRANCE		SYS0122		TS1
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)

4.5 ALTER TABLE...SPLIT SUBPARTITION

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、範囲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レンジ。

```
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>]
  );
```

Description

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、AdvancedServer 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参照する必要がありリファレンス。

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。

Parameters

`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]...)`

使用テーブルエントリがパーティションにグループ化されることにより引用リテラル値（またはリテラル値のカンマ区切りリスト）を指定する `value`。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）にすることもできます。

`DEFAULT` または `MAXVALUE` パーティションの作成については、[LIST または RANGE パーティションテーブルでの浮遊値の処理](#) 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。

4.5.1 Example - Splitting a LIST Subpartition

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 および second_half_2012 名前付けの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、2つの範囲定義された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'),
(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'),
(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;
```

(次のページに続く)

(前のページからの続き)

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)

4.5.2 Example - Splitting a RANGE Subpartition

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
)
PARTITION BY LIST(country)
SUBPARTITION BY RANGE(date)
(
```

(次のページに続く)

(前のページからの続き)

```

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'),
(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'),

```

(次のページに続く)

(前のページからの続き)

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

(次のページに続く)

(前のページからの続き)

);

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 および 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

(17 rows)

4.6 ALTER TABLE...EXCHANGE PARTITION

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。パーティション表にラージのデータを追加する場合は、バルク・ロードを実装するために ALTER TABLE... EXCHANGE PARTITION コマンドを使用することができます。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];
```

Description

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 target_table の構造とマッチしている必要があります (両方のテーブルは、列およびデータ・タイプとマッチング必要があります)、およびデータがテーブル内に含まれ、パーティショニングのコンストレインに準拠しなければなりません。

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 のインデックスは再構築され、逆もまた同様です (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 句を EXCHANGE PARTITION で指定されている場合、(target_partition と source_table にマッチングインデックスが交換されるが、source_table でマッチと target_partition インデックスが無効とその逆としてマークされている、でマッチと source_table のインデックスであります target_partition も無効としてマークされます)。

以前に使用*マッチング>は*用語は、照合順序オーダー、昇順または降順方向、第一ヌルまたは最後のヌルの順序

付け、等 CREATE INDEX コマンドによって決定されるように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 句で使用する 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。

Parameters:

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。

4.6.1 Example - Exchanging a Table for a Partition

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。次のコマンドを使用して、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 パーティションにある行は 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 テーブルの定義と一致するテーブル (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'),
(40, '4577b', 'US', '11-Nov-2012', '25000'),
```

(次のページに続く)

(前のページからの続き)

```
(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 パーティションに保存されていた行が 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
-----+-----+-----+-----+-----+-----
n_america | 40 | 9519b | US | 12-APR-12 00:00:00 | 145000
(1 row)
```

4.7 ALTER TABLE...MOVE PARTITION

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>;
```

Description

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表領域に移動しテーブルスペース。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 句で使用する 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。

Parameters

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。

4.7.1 Example - Moving a Partition to a Different Tablespace

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 ビューを照会すると、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、テーブルスペースの名前付け ts_3 にテーブルスペースの名前付け ts_2 から q1_2013 パーティションを移動するために ALTER TABLE... MOVE PARTITION コマンドを呼び出します。

```
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
-----+-----
 Q1_2012        |
```

(次のページに続く)

(前のページからの続き)

Q2_2012	
Q3_2012	
Q4_2012	TS_1
Q1_2013	TS_3
(5 rows)	

4.8 ALTER TABLE...RENAME PARTITION

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>;
```

Description

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 句で使われる 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。

Parameters

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。

4.8.1 Example - Renaming a Partition

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)
```

4.9 ALTER TABLE...SET INTERVAL

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。データベースは、INTERVAL が設定されている場合、パーティション化されたテーブルに対して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。構文は次の構文です。

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

時インターバル範囲パーティション表を無効にし、レンジパーティション表に変換するための構文は次の構文です。

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

Parameters

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 値を指定します。

Description

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、間隔レンジパーティショニングを使用できインターバル。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はレンジ・パーティションにテーブルを分割してレンジ・パーティションの境界にインターバルレンジ・パーティションの境界を設定し変換します。

4.9.1 Example - Setting an Interval Range Partition

以下の例では、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するためにレンジ・パーティショニングから sales テーブルのインターバルレンジのパーティションを設定します。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。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)

```

4.10 ALTER TABLE...SET [PARTITIONING] AUTOMATIC

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。構文は次の構文です。

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

AUTOMATIC LIST PARTITIONING を無効にして通常の LIST パーティションに変換するための構文は次の構文です。

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

Parameters

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）。

Description

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。

4.10.1 Example - Setting an AUTOMATIC List Partition

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化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。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)
```

4.11 ALTER TABLE...SET SUBPARTITION TEMPLATE

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。構文は次の構文です。

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

サブパーティションテンプレート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、次の構文を使用します。

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

Parameters

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

サブパーティションの数は、パーティションに追加します。

Description

ALTER TABLE... SET SUBPARTITION TEMPLATE コマンドを使用して、テーブル内のサブパーティションテンプレートを更新できます。あなたはパーティションのサブパーティションディスクリプタを指定している場合は、サブパーティションディスクリプタではなく、サブパーティション・テンプレートを使用しています。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。

注釈: ALTER TABLE... SET SUBPARTITION TEMPLATE コマンドを呼び出した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。

4.11.1 Example - Setting a SUBPARTITION TEMPLATE

次の例では、date によってレンジパーティション化され、country によって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,
  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 テーブルは、サブパーティションテンプレートを 8 に設定して変更されます。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)
```

4.11.2 Example - Resetting a SUBPARTITION TEMPLATE

次の例は、country によってリストパーティション化され、part_no によって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
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 テーブルは、サブパーティションテンプレートがデフォルトの 1 にリセットされて変更されます。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)
```

4.12 DROP TABLE

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。構文は次の構文です。

```
DROP TABLE <table_name>
```

Parameters

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）。

Description

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すべて削除されます。

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。

Example

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>

4.13 ALTER TABLE...DROP PARTITION

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。構文は次の構文です。

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

Parameters

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。

Description

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、または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。

4.13.1 Example - Deleting a Partition

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'),
```

(次のページに続く)

(前のページからの続き)

```
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'
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)
```

4.14 ALTER TABLE...DROP SUBPARTITION

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格納されているデータを削除します。構文は次の構文です。

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

Parameters

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。

Description

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の所有者、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。

4.14.1 Example - Deleting a Subpartition

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
-----+-----
EUROPE             | 'ITALY', 'FRANCE'
ASIA               | 'PAKISTAN', 'INDIA'
SYS0101            | DEFAULT
(3 rows)
```

4.15 TRUNCATE TABLE

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定義。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。構文は次の構文です。

```
TRUNCATE TABLE <table_name>
```

Description

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すべて削除されます。

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。

Parameters

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）。

4.15.1 Example - Emptying a Table

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)
(
    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'),
```

(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');
```

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)
```

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;
```

現在、sales テーブルを照会すると、データは削除されていますが、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>

4.16 ALTER TABLE...TRUNCATE PARTITION

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。構文は次の構文です。

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

Description

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。

Parameters

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。

DROP STORAGE および 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。

4.16.1 Example - Emptying a Partition

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
)
```

(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')
);
```

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)

```

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)

```

4.17 ALTER TABLE...TRUNCATE SUBPARTITION

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にします。構文は次の構文です。

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

Description

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。

Parameters

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。

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。

4.17.1 Example - Emptying a Subpartition

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,
```

(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 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'),
(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

(次のページに続く)

(前のページからの続き)

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)

第 5 章

Handling Stray Values in a LIST or RANGE Partitioned Table

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。

Defining a DEFAULT Partition

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 ルールです。

DEFAULT ルールの構文は次の構文です。

```
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格納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、AdvancedServer 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 others パーティションに europe, asia のパーティショニング、または americas パーティションに指定された値とマッチしないすべての行を格納します。

```
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'),
    PARTITION others VALUES (DEFAULT)
);
```

DEFAULT パーティションをテストするために、パーティショニングコンストレインで指定された国の一つとマッチしない country 列の値を持つ行を追加します。

```
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)

AdvancedServer 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。

- あなたは長い間、追加する既存のテーブルの行とパーティションの値の間に矛盾する値がないとしてとして DEFAULT ルールを表にパーティションを追加する ALTER TABLE... ADD PARTITION コマンドを使用することができます。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します。この箇条書きリストの後に例を示します。
- 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 コマンドを使用して、DEFAULT ルールでサブ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 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。

Adding a Partition to a Table with a DEFAULT Partition

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す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
```

Splitting a DEFAULT Partition

次の例では、DEFAULT パーティション、パーティションのコンテンツを 2 つの新しいパーティションに再分配します。テーブルは、このセクションの最初に示した 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'),
```

(次のページに続く)

(前のページからの続き)

```
(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 名前付けの 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);
```

パーティションは今 DEFAULT others パーティションと一緒に africa 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'
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分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)
```

Defining a MAXVALUE Partition

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 パーティションです。

MAXVALUE ルールの構文は次の構文です。

```
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、AdvancedServer 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
```

(次のページに続く)

(前のページからの続き)

```
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 パーティションします。サーバーは、エラーをスローする代わりに、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 others パーティションに保存します。

```
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'),
  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た最後の日付値を超え date 列の値を持つ行を追加します。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

(次のページに続く)

(前のページからの続き)

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)										

AdvancedServer 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。
- あなたは MAXVALUE ルールをテーブルにサブパーティションを追加する ALTER TABLE... ADD SUBPARTITION 文を使用することはできませんが、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することができます。

第 6 章

Specifying Multiple Partitioning Keys in a RANGE Partitioned Table

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でパフォーマンスを改善できます。列の小さなセットで比較演算子を使用して（大なりまたは小なりの値に基づいて）行を選択することが多い場合、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。

Specifying Multiple Keys in a Range-Partitioned Table

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 であるすべての行は q3_2012 パーティションに格納されるため、AdvancedServer はそのパーティションのみを検索します。

第 7 章

Retrieving Information about a Partitioned Table

AdvancedServer 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。

Querying the Partitioning Views

次のビューをクエリーて、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、[テーブル分割 Views-リファレンス](#) 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は、次を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、以下のコードは、サブパーティションテーブルの名前が表示されます。

```
edb=# SELECT subpartition_name, partition_name FROM ALL_TAB_SUBPARTITIONS;
 subpartition_name | partition_name
-----+-----
EUROPE_2011      | EUROPE
```

(次のページに続く)

(前のページからの続き)

EUROPE_2012		EUROPE
ASIA_2011		ASIA
ASIA_2012		ASIA
AMERICAS_2011		AMERICAS
AMERICAS_2012		AMERICAS
(6 rows)		

7.1 Table Partitioning Views - Reference

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。

7.1.1 ALL_PART_TABLES

次の表に、ALL_PART_TABLES ビューで使用可能な情報を示します。

コラム	種類	説明
owner	name	テーブルの所有者。
schema_name	name	テーブルが存在するスキーマ。
table_name	name	テーブルの名前。
partitioning_type	text	RANGE、LIST、または HASH
subpartitioning_type	text	RANGE、LIST、HASH、または NONE
partition_count	bigint	パーティションの数。
def_subpartition_count	integer	デフォルトのサブパーティションカウント-これは常に 0 です。
partitioning_key_count	integer	句ごとのパーティションにリストされる列の数。
subpartitioning_key_count	integer	サブパーティション by 句の列の数。
status	character varying(8)	この列は常に有効です。
def_tablespace_name	character varying(30)	この列は常に NULL です。
def_pct_free	numeric	この列は常に NULL です。
def_pct_used	numeric	この列は常に NULL です。
def_ini_trans	numeric	この列は常に NULL です。
def_max_trans	numeric	この列は常に NULL です。
def_initial_extent	character varying(40)	この列は常に NULL です。
def_next_extent	character varying(40)	この列は常に NULL です。
def_min_extents	character varying(40)	この列は常に NULL です。
def_max_extents	character varying(40)	この列は常に NULL です。
def_pct_increase	character varying(40)	この列は常に NULL です。
def_freelists	numeric	この列は常に NULL です。
def_freelist_groups	numeric	この列は常に NULL です。
def_logging	character varying(7)	この列は常に YES になります
def_compression	character varying(8)	この列は常に NONE になります
def_buffer_pool	character varying(7)	この列は常にデフォルトです
ref_ptn_constraint_name	character varying(30)	この列は常に NULL です
interval	character varying(1000)	この列は常に NULL です

7.1.2 ALL_TAB_PARTITIONS

次の表に、ALL_TAB_PARTITIONS ビューで使用可能な情報を示します。

コラム	種類	説明
table_owner	name	テーブルの所有者。
schema_name	name	テーブルが存在するスキーマ。
table_name	name	テーブルの名前。
composite	text	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。
partition_name	name	パーティションの名前。
subpartition_count	bigint	このパーティションのサブパーティションの数。
high_value	text	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。
high_value_length	integer	高いパーティショニング値の長さ。
partition_position	integer	このパーティションの順序位置。
tablespace_name	name	このパーティションが存在するテーブルスペース。
pct_free	numeric	この列は常に 0 です。
pct_used	numeric	この列は常に 0 です。
ini_trans	numeric	この列は常に 0 です。
max_trans	numeric	この列は常に 0 です。
initial_extent	numeric	この列は常に NULL です。
next_extent	numeric	この列は常に NULL です。
min_extent	numeric	この列は常に 0 です。
max_extent	numeric	この列は常に 0 です。
pct_increase	numeric	この列は常に 0 です。
freelists	numeric	この列は常に NULL です
freelist_groups	numeric	この列は常に NULL です
logging	character varying(7)	この列は常に YES になります。
compression	character varying(8)	この列は常に NONE になります。
num_rows	numeric	約このパーティションの行数。
blocks	integer	約このパーティションのブロック数。
empty_blocks	numeric	この列は常に NULL です
avg_space	numeric	この列は常に NULL です
chain_cnt	numeric	この列は常に NULL です
avg_row_len	numeric	この列は常に NULL です

次のページに続く

表 1 – 前のページからの続き

コラム	種類	説明
sample_size	numeric	この列は常に NULL です
last_analyzed	timestamp without time zone	この列は常に NULL です
buffer_pool	character varying(7)	この列は常に NULL です
global_stats	character varying(3)	この列は常に YES になります。
user_stats	character varying(3)	この列は常に NO になります。
backing_table	regclass	このパーティションのバックアップテーブル の OID。

7.1.3 ALL_TAB_SUBPARTITIONS

次の表に、ALL_TAB_SUBPARTITIONS ビューで使用可能な情報を示します。

コラム	種類	説明
table_owner	name	テーブルの所有者の名前。
schema_name	name	テーブルが存在するスキーマの名前。
table_name	name	テーブルの名前。
partition_name	name	パーティションの名前。
subpartition_name	name	サブパーティションの名前。
high_value	text	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。
high_value_length	integer	高いパーティショニング値の長さ。
subpartition_position	integer	このサブパーティションの順序位置。
tablespace_name	name	このサブパーティションが存在するテーブ ルスペース。
pct_free	numeric	この列は常に 0 です。
pct_used	numeric	この列は常に 0 です。
ini_trans	numeric	この列は常に 0 です。
max_trans	numeric	この列は常に 0 です。
initial_extent	numeric	この列は常に NULL です。
next_extent	numeric	この列は常に NULL です。
min_extent	numeric	この列は常に 0 です。
max_extent	numeric	この列は常に 0 です。
pct_increase	numeric	この列は常に 0 です。
freelists	numeric	この列は常に NULL です。
freelist_groups	numeric	この列は常に NULL です。

次のページに続く

表 2 – 前のページからの続き

コラム	種類	説明
logging	character varying(7)	この列は常に YES になります。
compression	character varying(8)	この列は常に NONE になります。
num_rows	numeric	約このサブパーティションの行数。
blocks	integer	約このサブパーティションのブロック数。
empty_blocks	numeric	この列は常に NULL です。
avg_space	numeric	この列は常に NULL です。
chain_cnt	numeric	この列は常に NULL です。
avg_row_len	numeric	この列は常に NULL です。
sample_size	numeric	この列は常に NULL です。
last_analyzed	timestamp without time zone	この列は常に NULL です。
buffer_pool	character varying(7)	この列は常に NULL です。
global_stats	character varying(3)	この列は常に YES になります。
user_stats	character varying(3)	この列は常に NO になります。
backing_table	regclass	このサブパーティションのバックアップテーブルの OID。

7.1.4 ALL_PART_KEY_COLUMNS

次の表に、ALL_PART_KEY_COLUMNS ビューで使用可能な情報を示します。

コラム	種類	説明
owner	name	テーブル所有者の名前。
schema_name	name	テーブルが存在するスキーマの名前。
name	name	テーブルの名前。
object_type	character(5)	この列は常に TABLE です。
column_name	name	パーティショニングキー列の名前。
column_position	integer	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...）

7.1.5 ALL_SUBPART_KEY_COLUMNS

次の表に、ALL_SUBPART_KEY_COLUMNS ビューで使用可能な情報を示します。

コラム	種類	説明
owner	name	テーブル所有者の名前。
schema_name	name	テーブルが存在するスキーマの名前。
name	name	テーブルの名前。
object_type	character(5)	この列は常に TABLE です。
column_name	name	パーティショニングキー列の名前。
column_position	integer	サブパーティション化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...）

第 8 章

Conclusion

EDBPostgresAdvancedServer データベース互換性テーブルパーティション分割ガイド

Copyright©2007-2020EnterpriseDBCorporation。

予約を禁じます。

EnterpriseDB

34CrosbyDrive、Suite201、Bedford、MA01730、アメリカ

T+17813573390F+19784671307E

info@enterprisedb.com

www.enterprisedb.com

- EnterpriseDB および PostgresEnterpriseManager は、EnterpriseDBCorporation の登録商標です。EDB および EDBPostgres は、EnterpriseDBCorporation の商標です。オラクルは、オラクル Inc の登録商標です。その他の商標は、それぞれの所有者の商標です。
- EDB は、存在する場合、EDB 製品のログオン UI の設計、コーディングのベストプラクティスの確立、レビュー、および入力バリデーションを行います。EDB は追加の入力コンポーネントに対して同じアプローチに従いますが、製品の性質上、限定されたバリデーションが可能な信頼できるユーザーによって入力および送信されるフリーフォーム SQL、WMI またはその他の文字列を受け入れる必要がある場合があります。このような場合、ユーザーが誤った入力や危険な入力を禁止することはできません。
- EDB は、フリーフォーム SQL または将来的に認証され、信頼されたユーザーからの他の潜在的に危険な入力を受け入れるの製品に機能を追加する権利を留保しますが、彼らは最低限のリスクの可能性を提供確保する保証に設計され、テストされているようなすべての機能を保証し、どこ可能であれば、スーパーユーザまたは同等の特権が必要です。
- EDB は、すべての潜在的な脅威を予測できることを保証しないため、すべての潜在的な脆弱性が対処または検討されたことを完全に保証することはできません。

索引

ALL_PART_KEY_COLUMNS, 118
 ALL_PART_TABLES, 114
 ALL_SUBPART_KEY_COLUMNS, 119
 ALL_TAB_PARTITIONS, 116
 ALL_TAB_SUBPARTITIONS, 117
 ALTER TABLE...ADD PARTITION, 36
 ALTER TABLE...ADD SUBPARTITION, 45
 ALTER TABLE...DROP PARTITION, 88
 ALTER TABLE...DROP SUBPARTITION, 90
 ALTER TABLE...EXCHANGE PARTITION, 70
 ALTER TABLE...MOVE PARTITION, 74
 ALTER TABLE...RENAME PARTITION, 77
 ALTER TABLE...SET [PARTITIONING] AUTOMATIC, 81
 ALTER TABLE...SET INTERVAL, 79
 ALTER TABLE...SET SUBPARTITION TEMPLATE, 83
 ALTER TABLE...SPLIT PARTITION, 50
 ALTER TABLE...SPLIT SUBPARTITION, 62
 ALTER TABLE...TRUNCATE PARTITION, 95
 ALTER TABLE...TRUNCATE SUBPARTITION, 98
 Automatic List Partitioning, 5

 Conclusion, 121
 CREATE TABLE...PARTITION BY, 13

 DROP TABLE, 87

 Example - Adding a Partition to a LIST Partitioned Table, 38
 Example - Adding a Partition to a RANGE Partitioned Table, 39
 Example - Adding a Partition with SUBPARTITIONS num..., 41
 Example - Adding a Partition with SUBPARTITIONS num... STORE IN, 43
 Example - Adding a Partition with SUBPARTITIONS num...IN PARTITION DESCRIPTION, 40
 Example - Adding a Subpartition to a LIST/RANGE Partitioned Table, 46
 Example - Adding a Subpartition to a RANGE/LIST Partitioned Table, 48
 Example - AUTOMATIC LIST PARTITION, 20
 Example - Deleting a Partition, 88
 Example - Deleting a Subpartition, 90
 Example - Emptying a Partition, 95
 Example - Emptying a Subpartition, 98
 Example - Emptying a Table, 92
 Example - Exchanging a Table for a Partition, 71
 Example - HASH/HASH PARTITIONS num ...STORE IN SUBPARTITIONS num... STORE IN, 29
 Example - HASH/HASH PARTITIONS num... SUBPARTITIONS num..., 27
 Example - HASH/HASH SUBPARTITIONS num... STORE IN, 28
 Example - HASH/RANGE PARTITIONS num..., 26
 Example - INTERVAL RANGE PARTITION, 22
 Example - LIST/HASH STORE IN...TABLESPACES, 33

Example - LIST/HASH SUBPARTITIONS num STORE IN... IN PARTITION DESCRIPTION, 32
 Example - LIST/HASH SUBPARTITIONS num..., 26
 Example - Moving a Partition to a Different Tablespace, 75
 Example - PARTITION BY HASH, 23
 Example - PARTITION BY HASH...PARTITIONS num..., 24
 Example - PARTITION BY HASH...PARTITIONS num...STORE IN, 25
 Example - PARTITION BY LIST, 19
 Example - PARTITION BY RANGE, 21
 Example - PARTITION BY RANGE, SUBPARTITION BY LIST, 34
 Example - Partition Pruning, 10
 Example - RANGE/HASH SUBPARTITIONS num..., 30
 Example - RANGE/HASH SUBPARTITIONS num... IN PARTITION DESCRIPTION, 31
 Example - Renaming a Partition, 78
 Example - Resetting a SUBPARTITION TEMPLATE, 85
 Example - Setting a SUBPARTITION TEMPLATE, 84
 Example - Setting an AUTOMATIC List Partition, 81
 Example - Setting an Interval Range Partition, 79
 Example - Splitting a LIST Partition, 52
 Example - Splitting a LIST Subpartition, 64
 Example - Splitting a Partition with SUBPARTITIONS num..., 58
 Example - Splitting a Partition with SUBPARTITIONS num...STORE IN, 59
 Example - Splitting a RANGE Partition, 54
 Example - Splitting a RANGE Subpartition, 66
 Example - Splitting a RANGE/LIST Partition, 56

 Handling Stray Values in a LIST or RANGE Partitioned Table, 103

 Interval Range Partitioning, 4

 Partitioning Commands Compatible with Oracle Databases, 13

 Retrieving Information about a Partitioned Table, 113

 Selecting a Partition Type, 3
 Specifying Multiple Partitioning Keys in a RANGE Partitioned Table, 111

 Table Partitioning Views - Reference, 114
 TRUNCATE TABLE, 92

 Using Partition Pruning, 7