

Oracle Partitioning

卓越したデータ管理とパフォーマンス

ホワイト・ペーパー/2019年2月19日

本書の目的

本書では、Oracle Partitioning の機能の概要と強化された点が説明されています。本書は、Oracle Database のビジネス上の利点の評価と IT プロジェクトの計画立案を支援することのみを目的としています。

免責事項

本文書には、ソフトウェアや印刷物など、いかなる形式のものも含め、オラクルの独占的な所有物である占有情報が含まれます。この機密文書へのアクセスと使用は、締結および遵守に同意した Oracle Software License and Service Agreement の諸条件に従うものとしてします。本文書と本文書に含まれる情報は、オラクルの事前の書面による同意なしに、公開、複製、再作成、またはオラクルの外部に配布することはできません。本文書は、ライセンス契約の一部ではありません。また、オラクル、オラクルの子会社または関連会社との契約に組み込むことはできません。

本書は情報提供のみを目的としており、記載した製品機能の実装およびアップグレードの計画を支援することのみを意図しています。マテリアルやコード、機能の提供をコミットメント（確約）するものではなく、購買を決定する際の判断材料になさらないで下さい。本書に記載されている機能の開発、リリース、および時期については、弊社の裁量により決定されます。

製品アーキテクチャの性質により、コードが大幅に不安定化するリスクなしに、本書に記載されているすべての機能を安全に含めることができない場合があります。

目次

本書の目的.....	2
はじめに.....	4
パーティション化の基本.....	5
パーティション化の概念.....	5
パフォーマンス向上のためのパーティション化.....	7
管理性を高めるためのパーティション化.....	8
可用性を得るためのパーティション化.....	10
パーティション化による情報ライフサイクル管理.....	10
パーティション化戦略.....	11
パーティション・オブジェクトのデータ分散方法.....	11
パーティション化の拡張機能.....	13
Partition Advisor.....	15
パーティション化機能の概要.....	16
結論.....	17

はじめに

20 年以上にわたって開発されてきた Oracle Partitioning は、Oracle データベースでもっとも成功し、もっともよく使用されている機能としての地位を確立しています。Oracle Partitioning を使用すると、データベースの単一の論理オブジェクトは、パーティションと呼ばれる複数の小さい物理オブジェクトに分割されます。この物理的なパーティション化の知識があれば、どのようなアプリケーションであっても、データベースによってパフォーマンス、管理性、可用性を向上させることができます。OLTP、データウェアハウス、または混合ワークロード・アプリケーションを使用している場合でも、システムが数百 GB やペタバイト範囲の規模であっても、Oracle Partitioning が役立ちます。問合せとメンテナンス操作が桁違いに高速化している一方で、処理に必要なリソースが最小化されています。ゾーン・マップ機能と併用すれば、プルーニング機能の制約が取り払われます。表とパーティションをさらに小さい物理ゾーンに分割すれば、表のパーティションに関する情報に加えて、データ・プルーニングの粒度を上げることができます。

"階層型アーカイブ"を使用することで、古い関連情報をオンラインのまま、最適な圧縮形式で低コストのストレージ・デバイスに保存しながら、最新データをオラクルのインメモリ列ストアに保存できるため、データの総所有コストを大幅に削減できます。自動データ最適化とヒートマップを Oracle Partitioning と併用すれば、情報ライフサイクル管理 (ILM) 戦略をシンプルかつ自動的に実装できます。Oracle Database 19c で導入されたハイブリッド・パーティション表では、パーティション化された表は同一の論理表の中で内部ストレージおよび外部ストレージに格納されるようになり、ILM がいっそう高いレベルに引き上げられました。

Oracle Partitioning は、何万ものお客様、何十万ものアプリケーションのパフォーマンス、管理性、可用性を改善してきました。読者の皆様も含め、誰もが Oracle Partitioning からメリットを得ることができます。

パーティション化の基本

パーティション化の概念

パーティション化によって、表と索引をさらに細かく分割できます。分割されたそれぞれのデータベース・オブジェクトをパーティションと呼びます。パーティションには固有の名前があり、独自のストレージ特性を持つ場合もあります。データベース管理者の視点からすると、パーティション・オブジェクトには、まとめて管理することも個別に管理することも可能な複数の単位があります。これによって、管理者はパーティション化されたオブジェクトをかなり柔軟に管理できるようになります。一方、アプリケーションにとっては、パーティション表は非パーティション表と同じであるため、SQL DML コマンドを使用してパーティション表にアクセスする際に変更は必要ありません。論理的には、1 つの表のままであり、アプリケーションは非パーティション表の場合と同様にこの 1 つの表にアクセスできます。

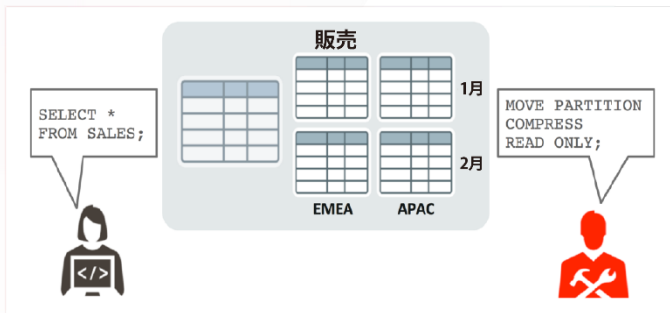


図1：アプリケーションとDBAから見た、パーティション表

表や索引などのデータベース・オブジェクトは、**パーティション・キー**を使用してパーティション化されます。パーティション・キーは、指定された行がどのパーティションに存在するかを決定する一連の列です。表のパーティションは物理的にデータを保存しますが、表自体はメタデータのみで構成されています。たとえば、図 1 の販売表は、月ごとのパーティション化戦略によって販売（注文）日でレンジ・パーティション化されます。表は、1 つの通常の表としてアプリケーションに表示されます。ただし、データベース管理者は月ごとのパーティションを個別に管理、保存でき、データの重要度と使用頻度に応じてデータ保管を最適化することができます。

古いデータ・レンジを保管するパーティションは、表圧縮を使って異なるストレージ階層に保管し（または読取り専用の表領域に保管することも、読取り専用パーティションとマーク付けることも可能）、最新のパーティションは、オラクルのインメモリ列ストアに保管するようにマーク付けることができます。ハイブリッド・パーティション表の場合、一部の表は内部ストレージに、一部の表は外部ストレージに格納されますが、すべて同一の論理表内にあります。

コンポジット・パーティション表の場合は、2 つ目の一連の列を使用して、各パーティションをパーティション内でさらに小さいサブパーティションに再分割します。特定の行のデータ配置は、両方のパーティション・キーの基準によって決められ、適切なサブパーティションに配置されます。コンポジット・パーティション表では、パーティション・レベルはメタデータ・レイヤーになります。サブパーティションのみが物理的にディスクに格納されます¹。

論理オブジェクトは 1 つ、物理パーティションは多数

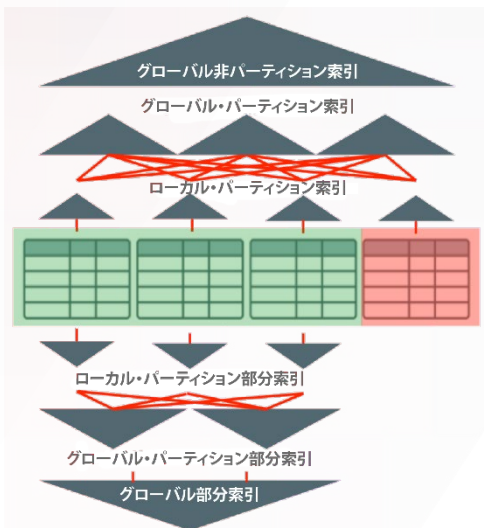
アプリケーションの側から見ると、アクセスしている表は 1 つです。管理者の視点からは、個別に管理している複数のパーティションがあります。

¹ 説明をわかりやすくするため、このドキュメントでは以降、パーティションのみについて言及します。

パーティション外部表の場合、表のさまざまな部分にさまざまな物理セグメントを配するという概念は、データベース外部の物理ストレージに拡大されます。外部表の各パーティションには、パーティションのデータのサブセットを表す 1 つまたは複数の個別ファイルがあります。ただし、通常のパーティション表と違って、データ配置はデータベースによって行われません。外部表はパーティション化されているかどうかにかかわらず、読取り専用です。

ハイブリッド・パーティション表は、内部パーティション表と外部パーティション表の両方のコンセプトを結合したものです。名前が示すとおり、そのようなパーティション表では、内部（データベース内）と外部（データベースの外の物理ストレージ）の両方にパーティションが格納されます。パーティション化された外部表のパーティションに関する法則は、ハイブリッド・パーティション表の外部パーティションにも適用されます。つまり、データベースはデータを配置せず、そのようなパーティションのコンテンツは読取り専用になります。

アプリケーション開発者は一般的に、表がパーティション化されているかどうかで悩む必要がないだけでなく、パーティション化を利用して効果をあげることもできます。たとえば、表からデータを消去する、リソース消費量の多い DML 操作は、パーティション・メンテナンス操作を使用して実行できます。これによって実行時間が劇的に向上すると同時に、リソース消費量が大幅に軽減します。



選択した表のパーティション化戦略に関係なく、パーティション表の索引は、その表の基本となるパーティション化戦略に結合されるか、結合解除されます。Oracle Database 18c は、索引を 3 種類に区別します²。

ローカル索引は、基本となるパーティション表に結合されるパーティション表の索引です。表からパーティション化戦略を'継承'します。そのため、ローカル索引の各パーティションは、基礎となる表の唯一のパーティションに対応します。結合によって、パーティション・メンテナンスを最適化できます。たとえば、表のパーティションが削除された場合、Oracle Database ではそれに対応する索引のパーティションを削除するだけで対応できま

す。索引パーティションは定義上、その表パーティションのみに関連付けられるため、コストのかかる索引メンテナンスは必要ありません。ローカル索引セグメントには、他のパーティションのデータが含まれることはありません。データウェアハウス環境では、ローカル索引がもっとも一般的です。

グローバル・パーティション索引は、表とは異なるパーティション・キーまたはパーティション化戦略でパーティション化される、パーティション表または非パーティション表の索引です。グローバル・パーティション索引では、レンジ・パーティション化またはハッシュ・パーティション化を使用してパーティション化することで、基礎となる表と切り離すことが可能です。たとえば、表を月ごとにレンジ・パーティション化して 12 個のパーティションに分け、この表の索引は、異なるパーティション・キーを使用してハッシュ・パーティション化し、異なる数のパーティションに分

²パーティション外部表に索引付けすることはできません。

けることができます。表から索引を切り離すと、表のパーティション・メンテナンス操作によって、索引メンテナンス操作が行われる可能性が必然的に生じます。グローバル・パーティション索引は、データウェアハウス環境よりは OLTP 環境で一般的です。

グローバル非パーティション索引は、非パーティション表の索引と基本的に同じです。索引構造はパーティション化されず、基本となる表から分離されます。データウェアハウス環境では、グローバル非パーティション索引のもっとも一般的な使用目的は、主キー制約の強制です。一方、OLTP 環境では、グローバル非パーティション索引にほとんど依存しています。

前述の索引の種類はすべて、パーティション表の全パーティションで作成することも（デフォルトの**完全索引**）、パーティション表のパーティションのサブセットのみで作成することもできます（**部分索引**）³。

部分索引はパーティション表の索引のみに適用できます。特定のパーティションが索引付けされるかどうかは、パーティションのプロパティで決まり、すべての部分索引に適用されます。たとえば部分索引の場合、データ挿入時の索引のメンテナンス作業を回避するために、最新のパーティションには索引付けを行わないようにすることで、データ・ロード速度を最大化することができます。ゾーン・マップのデータ・ブルーニングと併用すると、最新のパーティションの個別データ・アクセスに索引を使用しないことによる影響が最小限に抑えられます。

適切な索引付け戦略は、ビジネス要件とアクセス・パターンに基づいて選択されます。このため、あらゆるアプリケーションに対応する適切なパーティション化を実行できます。

パフォーマンス向上のためのパーティション化

特定の行の配置は、パーティション・キーのその値によって決まります。表のデータが複数のパーティション全体でどのように分割されるかが、表または索引のパーティション・メタデータとして保存されます。このメタデータは、問合せ、DML、およびパーティション・メンテナンス操作といったすべての SQL 操作で、指定の操作に関連のある表のパーティションはどれかを決定するために使用され、データベースは自動的に関連パーティションのみを扱います。ゾーン・マップ併用時には、パーティションまたは表の一部のみ扱います。パーティション化は、調査または操作対象のデータ量を制限することによって、パフォーマンス面でのメリットを多数もたらします。

パーティション・ブルーニング（パーティション削除とも呼ばれる）は、パフォーマンスを改善するもっとも簡単で効果的な手段です。パーティション・メタデータを利用して、SQL 操作に関連するデータのみを処理することで、多くの場合に問合せのパフォーマンスを格段に改善できます。たとえば、アプリケーションに注文の履歴データを含む注文表があり、注文日に基づいて日次でパーティション化されているとします。1 週間の注文をリクエストする問合せは、注文表の 7 つのパーティションにのみアクセスします。注文表に 2 年分の履歴データがある場合、この問合せでは 730 のパーティションではなく 7 つのパーティションにアクセスします。この問合せの実行速度は、単純にパーティション・ブルーニングの効果で 100 倍速くなる可能性があります。パーティション・ブルーニングは、オラクルが提供する製品のその他すべてのパフォーマンス機能と連携します。パーティション・ブルーニングは、索引付けや結合の技術、またはパラレル・アクセスの手法と組み合わせて使用されます。

ゾーン・マップ⁴は、表のパーティション・メタデータ以上にオラクルのブルーニング機能を拡張するものです。データ・ブルーニングをパーティション・レベルで実行できるほか、さらにきめ細

パーティション表の柔軟な索引付け

- 索引は、パーティション化してもしなくてもかまいません。
- 索引は表パーティションに関連付けることも、表パーティションから独立させることもできます。
- 表全体に索引を付けることも、パーティションのサブセットに部分的に付けることもできます。

³一意索引を部分索引にすることはできません。

⁴ゾーン・マップの概要と詳細については、『Oracle Database データ・ウェアハウス・ガイド』を参照してください。

かく'ゾーン'でも実行できます。ゾーンは連続したブロック領域であり、ゾーン・マップはこの領域で指定の列の最大値と最小値を追跡します。これらの列がパーティション・キー列ではない点に留意してください。パーティション・キー列を含めることはできますが、ゾーン・マップのもっとも一般的な用途は、他の非パーティション・キー列を使用することです。パーティション表の場合、パーティションごとに集計された最小値と最大値がゾーン・マップに含まれています。ゾーン・マップに指定されている列を使用して SQL 操作で対象のデータを制限（フィルタ）する場合は常に、フィルタとゾーン・マップの情報が比較され、一致するデータが含まれないゾーンやパーティションにはアクセスは行われません。ゾーン・マップはその点で Exadata ストレージ索引に似ていますが、ストレージ索引を補完する他のメリットを提供します。ゾーン・マップはデータベースで処理される永続的なデータ構造であり、ローカルの列（ゾーン・マップを使用する表の列）と結合列を指定できます。

データベースにゾーン・マップを使用すると、あらゆる SQL 文でメリットが得られます。例として、前述の販売表を使用して説明すると、特定の期間に出荷された販売注文の情報をリクエストする問合せでは、注文表のすべてのパーティションにアクセスする必要があります（パーティション・キーが出荷日でなく注文日であるため）。注文日と出荷日には相関関係がありますが、アクセスするパーティションを出荷日だけで制限することは不可能です。ゾーン・マップを使用した場合、データベースには出荷日の最小値と最大値が含まれているため、ゾーン・マップにはパーティションごとにこの情報が格納されています。注文日から 1 営業週以内の出荷日である場合、過去 3 週間に出荷された製品の問合せでは、過去 4 週間の注文のパーティションにアクセスするだけでよく、これらのパーティション内の、この期間に出荷されたゾーンにアクセスすれば済みます。パーティション・キー列にフィルタ条件を指定しなくても、パーティション・プルーニングとゾーン・マップ・プルーニングを利用できます。

パーティション化を通じて複数表の結合のパフォーマンスも向上できます。これには**パーティション・ワイズ結合**という方法を使用します。パーティション・ワイズ結合は、2 つの表が結合される場合に適用でき、2 つの表のうち少なくとも 1 つは結合キーでパーティション化されます。パーティション・ワイズ結合では、結合表の大きい結合を、'同一'のデータセットの小さい結合に分割します。ここでの'同一'は、結合の両側で完全に同じセットのパーティション・キー値も含めて定義されています。これら'同一'のデータセットの結合だけが結果を生むので、他のデータセットを考慮する必要はありません。すでに物理的に等価パーティション化された結合する表（フル・パーティション・ワイズ結合）を使用するか、実行時に透過的に 1 つの表（小さい方）を再分散（"再パーティション化"）して他の表のパーティション化と一致する等価パーティション化されたデータセットを作成（**パーシャル・パーティション・ワイズ結合**）すると、全体の結合時間が短縮され、使用するリソースも少なくて済みます。これによって、シリアル実行とパラレル実行両方のパフォーマンスが大幅に向上します。

管理性を高めるためのパーティション化

表と索引をより小さい、管理しやすい単位にパーティション化することで、データベース管理者はデータ管理に対して"分割統治"アプローチを使用できます。オラクルでは、パーティション表を管理する包括的な一連の SQL コマンドを提供しています。これには、パーティションの追加、削除、分割、移動、マージ、切捨て、および交換を実行するコマンドが含まれます。

パーティション化によって、表の特定の部分だけにメンテナンス操作を行うことができます。たとえば、データベース管理者は、2017 年の表のデータを含む 1 つのパーティションを圧縮できます。表全体を圧縮する必要はありません。圧縮操作の一部として、このパーティションはより低コストのストレージ層へと移動することもでき、さらに保存されたデータの総所有コストを削減できます。

このパーティション・メンテナンス操作は完全にオンラインで実行できるため、データ・メンテナンス操作の処理中に、問合せと DML 操作の両方を実行できます。

Oracle Database 18c 以降では、複数のパーティションでのパーティション・メンテナンス操作を単一のアトミック操作として実行できます。たとえば、3 つのパーティション 'January 2018'、'February 2018'、'March 2018' を、1 回のパーティション・マージ操作で 1 つのパーティション 'Q1 2018' にマージできます。

管理のためにパーティション化を行うもう 1 つの典型的な例として、データウェアハウスでローリング・ウィンドウ方式のロード・プロセスをサポートする場合があります。DBA が毎日新しいデータを表にロードするとします。各パーティションに 1 日分のデータが入るように、この表をレンジ・パーティション化できます。ロード・プロセスは単に新しいパーティションの追加になります。DBA は他のパーティションを変更する必要がないため、単一パーティションの追加は表全体の変更よりはるかに効率的です。

極めて効率的かつスマートな方法でデータを削除することも、パーティション化のもう 1 つの重要な利点です。たとえば、パーティション表からデータを消去するには、1 つまたは複数のパーティションを削除または切り捨てるだけで済みます。同等の削除コマンドを実行して、多数のリソースを使用し、削除されるすべての行を処理するよりも、非常に低コストで迅速なデータ・ディクショナリ操作です。Oracle Database 18c からは、削除や切捨てといったパーティション・メンテナンス操作でデータを削除する一般的な操作は最適化されています。これらの操作では、索引のメンテナンスを即座に実行しなくてもすべての索引が有効に保たれるため、高速なメタデータのみ処理となります⁵。

パーティション・メンテナンス操作でデータを高速に削除できる一方で、そのような操作の粒度は、削除または切り捨てられるパーティションの境界によって左右されます。ただし、よくあることですがルールには例外があります。たとえば、ローリング・ウィンドウ操作の一部として、3 年以上経ったデータをすべて削除する必要がありますが、正式にクローズしていない注文は削除できずとします。このようなことは業務ではめったに発生しませんが、この業務要件のために、パーティションの切捨てまたは削除をそのまま行うことができなくなります。この例外を維持するには、プログラムによってこの状況に対処する必要があります。Oracle Database 18c 以降、パーティション・メンテナンス操作が強化されて、データのフィルタリングを任意のパーティション・メンテナンス操作の一部として実行できるようになりました。この例の場合、パーティションを移動し、正式にクローズしていない古いすべてのレコードを保持することで、データの削除を達成します。フィルタ付きパーティション・メンテナンス操作により、データ・メンテナンスとパーティション・メンテナンスを同時に実行できます⁶。

さらに、既存の非パーティション表とパーティション表を変更して、パーティション表としたり、完全オンラインの方法でパーティション戦略を変更したりできます。既存の全索引を変更することもできます。拡大するシステムにパーティション化を導入する必要が生じた場合でも、または変化するビジネス要件への適応やさらなる成長への対応でパーティショニング戦略の調整が必要な場合のどちらであっても、Oracle Database は対応します。

⁵ 非同期グローバル索引のメンテナンスについては、『VLDB およびパーティショニング・ガイド』で説明されています。

⁶ フィルタ付きパーティション・メンテナンス操作では、フィルタ条件はパーティション表のみで使用され、結合やその他の複雑な SQL 構文はサポートされません。

可用性を得るためのパーティション化

パーティション化されたデータベース・オブジェクトによって、パーティションの独立性が確保されます。このパーティションの独立性という特長は、高可用性戦略の重要な部分です。たとえば、パーティション化された表で 1 つのパーティションが使用できない場合でも、この表の他のパーティションはすべてオンラインのまま使用できます。アプリケーションは、問合せとトランザクションの実行をこのパーティション化された表に引き続き実行できるため、使用不可能なパーティションにアクセスする必要がないのであれば、これらのデータベース操作は成功します（ある操作によって、使用不可能なデータにアクセスしようとするれば、その操作は当然ながら失敗します。オラクルはどんな場合でも、真実かつ有効な結果のみを返します）。

データベース管理者は、別の表領域に格納する各パーティションを指定できます。これによって、その表の他のパーティションとは独立して、(partition-to-tablespace マッピングにより) パーティションごとにバックアップとリカバリ操作を実行できます。障害が発生した場合、使用中のデータを含むパーティションでデータベースをリカバリできます。他のパーティションの使用していないデータは必要に応じてリカバリできます。これにより、システムの停止時間が低減します。データベース全体のサイズに関係なく、もっとも関連性の高いデータが極めて短時間で再び使用可能になります。

さらに、パーティション化によって予定の停止時間を短縮できます。パーティション化によるパフォーマンス面でのメリットによって、データベース管理者は、大型データベース・オブジェクトのメンテナンス処理を比較的短いバッチ時間で実行できます。

パーティション化による情報ライフサイクル管理

できるだけ低コストで大量のデータを保存するという今日の課題には、自動データ最適化およびヒートマップと Oracle Partitioning を使用することで適切に対処できます。各パーティションの独立性は、パーティションでの効率的で透過的なデータ・メンテナンス操作と併せて、“階層型アーカイブ”戦略のオンライン部分に対応する上で非常に重要な要素です。特に履歴データを含む表の場合、データの重要性和アクセス方式は、データの古さに大きく依存します。Oracle Partitioning を使用すると、異なる物理属性（圧縮やデータが読取り専用かどうかなど）および価格を条件として、各種ストレージ層に各パーティション（またはパーティションのグループ）を格納できます。ハイブリッド・パーティション表では、古いパーティションの一部が、オラクル・データベースの外の外部ストレージにある場合があります。そのようなデータは当然ながら、読取り専用です。

内部パーティションでは、表領域の基盤であるストレージ・コンテナへの物理的な変更を防ぐ読取り専用表領域に加えて、個々のパーティションを読取り専用を設定できます。パーティションを読取り専用を設定すると、パーティション内のデータの DML の実行を防止して、読取り専用パーティション内のデータが不適切に変更されるのを防ぐことができます。厳密に言うと、パーティションが読取り専用を設定された時点で表内に存在していた全列のデータを変更できなくなります。たとえば、5 年分のデータを含む販売表の場合、直近の四半期分のみを高価な高性能ストレージ層に格納して、表の残り（データの約 90 %）を安価なストレージ層に格納できます。さらに、もっとも古い 2 年分のデータを外部パーティションとしてデータベース外に、および次の 2 年分を読取り専用パーティションとして格納できます。変更できるのは直近のデータのみです。古いものは全デー

タが変更不可ですが、規制遵守のためにシステム内で利用することはできます。これは、全データがデータベース内に保存されているわけではない場合も適用されます⁷。

自動データ最適化 (ADO) が加わったことで、ヒートマップによって自動収集される使用状況の統計に基づいて、ストレージ階層化と圧縮の階層化を特定のパーティションに実行するタイミングを指定するポリシーを定義できます。ADO ポリシーは手動による操作を必要とせず、Oracle Database によって自動的に評価されて実行されるため、複雑なスクリプトやジョブを作成することなく、ストレージ階層化と圧縮によるコスト節減とパフォーマンスの利点を達成できます⁸。

パーティション化戦略

オラクルは、もっとも包括的な一連のパーティション化戦略を提供します。これによってお客様は、データを実際のビジネス要件に最適な方法で分割できます。使用できるすべてのパーティション化戦略は、単一 (1 つのレベル) のパーティション表またはコンポジット (2 つのレベル) パーティション表のいずれかに使用できる**基本データ分散方法**に依存します。この機能によって、パーティション・キー選択の柔軟性が向上し、必要に応じて自動的にパーティションを作成でき、親子関係を通じて論理的に接続された表グループ全体でパーティション化戦略を共有し、パーティション化されていないオブジェクトのパーティション化戦略をアドバイスできます。また、オラクルは、さまざまな**パーティション化の拡張機能**を提供しています。

パーティション・オブジェクトのデータ分散方法

Oracle Partitioning は、パーティションに配置されるデータの移動を制御する以下の 3 つの基本的なデータ分散方法を提供します。

- **レンジ**：パーティション・キーの値の範囲に基づいて、データが分散されます (パーティション・キーが日付列の表の場合、'January-2018'のパーティションには、'01-JAN-2018'から'31-JAN-2018'のパーティション・キーの値を持つ列が含まれます)。レンジの分散は、途切れることなく連続して行われます。レンジは常に、パーティションの排他的上限として定義され、パーティションの下限は、先行するパーティションの排他的上限によって自動的に定義されます。パーティションの境界は常に広がり続けています。そのため、表の最初のパーティション (レンジが最下限のパーティション) は、より低い値が無制限に続きます。最後のパーティション (境界が最上限のパーティション) も、無制限に値が続くよう (MAXVALUE) に任意で設定できます。レンジ・パーティション化には、1 つまたは最大 16 列までの複数のパーティション・キー列を含めることができます。
- **リスト**：データの分散が、パーティション・キーの個々の値リストによって定義されます (パーティション・キーが地域列の表の場合、'North America'のパーティションには、'Canada'、'USA'、'Mexico'の値が含まれます)。特殊な'DEFAULT'パーティションを定義して、リストで明示的に定義されていないパーティション・キーのすべての値を取得できます。ヒープ表の場合、リスト・パーティション化には、1 つまたは最大 16 列までの複数のパーティション・キー列を含めることができます。索引構成表では、1 つのパーティション・キー列のみがサポートされます。

⁷ 外部パーティションはデータベース内からの読取り専用ですが、データベースはそのコンテンツを制御することはできませんし、データがデータベース外から変更されていないことを保証することもできません。読取り専用の内部パーティションのみを対象として、オラクルはそのようなパーティションが読取り専用設定されている限り、データが変更されていないことを保証します。

⁸ ADO は現時点で、内部 (データベースが管理する) ストレージ層のみをサポートしています。

- **ハッシュ**：あるパーティション・キーのパーティションを決定するために、内部ハッシュ・アルゴリズムがパーティション・キーに適用されます。他の 2 つのデータ分散方法とは異なり、データとパーティションの論理的なマッピングは行われませんが、パーティションのサイズはほぼ同じバランスになります。パーティション・キーの個々の値が十分にあり、2 の累乗（4、16、64 など）であるパーティション数を選択すれば、もっともバランスの取れたパーティション・サイズが得られます。ハッシュ・パーティション化には、1 つまたは最大 16 列までの複数のパーティション・キー列を含めることができます。

レンジ、リスト、およびハッシュという、これらの 3 つの基本的なデータ分散方法を使用して、単一のパーティション表またはコンポジット・パーティション表として表をパーティション化できます。

これらの基本的な方法に加えて、オラクルは**システム・パーティション化**も提供しています。データベースには表をパーティション化するためのフレームワークのみが用意されており、データの配置を判断するためのメタデータは格納されません。データ挿入とデータ・アクセスの両方について、データの配置はアプリケーション・レイヤーで管理されます（アプリケーションでパーティション・プルニングを利用する必要がある場合）。システム・パーティション化は、ドメイン索引など、データの配置やアクセスでの特殊なニーズに対応する開発フレームワークとして設計されており、単一（1 つのレベル）のパーティション化が行われたヒープ表のみがサポートされています。パーティション・キーに相当するものの定義と管理は、アプリケーションによって制御されます。

単一（1つのレベル）のパーティション化

上述のデータ分散方法のいずれかを指定して表を定義します。パーティション・キーとして 1 つ以上の列を使用します。たとえば、パーティション・キーに数値列を使用して 'less_than_five_hundred' と 'less_than_thousand' の 2 つのパーティションを持つ表の場合、'less_than_thousand'パーティションには、'500 <= パーティション・キー < 1000'の条件を満たす行が含まれます。単一のパーティション表または索引のパーティションは、オブジェクトの実際のデータを格納する、データベース内の個々の物理セグメントです。

レンジ、リスト、ハッシュ、およびシステム・パーティション・ヒープ表と索引構成表を指定できます。ハッシュ・クラスタをパーティション化するには、レンジ・パーティション化のみ使用します⁹。

コンポジット（2つのレベル）パーティション化

2 つのデータ分散方法を組み合わせて、コンポジット・パーティション表を定義します。最初に、1 番目のデータ分散方法を使用して、表をパーティション化します。次に、2 番目のデータ分散方法を使用して、各パーティションをサブパーティションに分割します。たとえば、レンジ・リスト・コンポジット・パーティション表は、最初にレンジ・パーティション化されます。次に、リスト・パーティション化を実施して、各レンジ・パーティションをサブパーティション化します。コンポジット・パーティション表のパーティションはメタデータであり、実際のデータ保存を表すものではありません。コンポジット・パーティション表または索引のパーティションのサブパーティションは、特定のパーティションのデータを保存する、データベース内の物理セグメントです。

⁹ クラスタは、データ構造に格納されている複数の表で構成される、スキーマ・オブジェクトです。ハッシュ・クラスタでは、同じハッシュ値を持つ行がデータベースと一緒に格納されます。クラスタは、IO を最小限に抑えるために、おもに OLTP 環境で使用されます。

使用できるコンポジット・パーティション化技術は、レンジ-ハッシュ、レンジ-リスト、レンジ-レンジ、リスト-レンジ、リスト-リスト、リスト-ハッシュ、ならびにハッシュ-ハッシュ、ハッシュ-レンジ、ハッシュ-リストです。コンポジット・パーティション化はデータベースによってヒープ表でのみサポートされています。

グローバル・パーティション索引は、レンジまたはハッシュ・パーティション化を使用してパーティション化できます。コンポジット・パーティション化は、グローバル・パーティション索引ではサポートされていません。

パーティション化の拡張機能

Oracleでは、基本的なパーティション化戦略の利用を強化するパーティション化の拡張機能を提供しています。パーティション化の拡張機能によって、パーティション化されたオブジェクトの管理性を強化し、表のパーティション・キー、あるいは親子関係で論理的に接続された表グループのパーティション・キーもより柔軟に定義できます。パーティション化の拡張機能はデータベースによってヒープ表でのみサポートされています。

インターバル・パーティション化

インターバル・パーティション化は、表のメタデータの一部としてインターバル定義を使用することで、将来のパーティションに対応できるように等間隔のパーティション範囲を定義して、レンジ・パーティション化機能を拡張します。インターバル・パーティション表は、ユーザーが介入しなくても、最初にパーティション表のパーティションが1つで作成された場合でも、最大総数 1048575 まで自動的に拡張できます。将来使用する個別のレンジ・パーティションを明示的に作成するのではなく、このようなパーティションのデータが最初に挿入されるたびに必要に応じて自動的に新しいパーティションを作成します。インターバル・パーティション化によって、パーティション表の管理性が大幅に向上します。たとえば、カレンダー年の各日に新しいパーティションを作成するインターバル・パーティション表を定義できます。この場合、'September, 19th 2031'の最初のレコードがデータベースに挿入されるとすぐにパーティションがこの日用に自動的に作成されます。

インターバル・パーティション化は、レンジ・パーティション化の拡張機能です。将来のパーティション用にインターバル定義を指定することで、レンジ・パーティション表をインターバル・パーティション表にすることができます。その唯一の要件は、変更前にレンジ・パーティション表の最後のパーティションが MAXVALUE ではなく、個別の最上限の境界を持つことです。無制限に値が続く最上限の境界を設ける手法は、インターバル定義に基づいて将来のパーティションを作成する手法とはまったく異なります。

インターバル・パーティション表の利用可能な技術は、インターバル、インターバル-リスト、インターバル-ハッシュ、インターバル-レンジです。Oracleでは、パーティション化の拡張機能であるインターバル・パーティション化と参照パーティション化の組合せもサポートします。インターバルは、現在、上位パーティション化手法に対応するサブパーティション化戦略 (*- Interval) としてはサポートされていません。

自動リスト・パーティション化

インターバル・パーティション化と同様に、新しいパーティション・キー値を自動リスト・パーティション表に挿入すると、自動リスト・パーティション化によってすぐに新しいリスト・パーティションを自動的に作成できます。値が既存のパーティションのパーティション・キー値として含まれていない場合、各個別値はその個々のパーティションに格納されます。

自動リスト・パーティション化は、リスト・パーティション化の拡張機能であり、既存のリスト・パーティション表を自動リスト・パーティション表にすることができます。その唯一の要件は、変更前にリスト・パーティション表でDEFAULTパーティションを定義しないことです。この包括的なパーティションを作成する手法は、新しいパーティション・キー値のために新しいパーティションを自動的に作成する手法とはまったく異なります。

自動リスト・パーティション化はパーティション拡張機能として利用できます。現在、サブパーティション化手法としてはサポートされていません。また、参照パーティション化との組合せもサポートされていません。

参照パーティション化

参照パーティション化では、既存の親子関係を利用して表をパーティション化できます。主キーと外部キーの関係を使用することで、親表のパーティション化戦略は、親のパーティション・キー列を子表に保存しなくても、子表に継承されます。親子表のパーティション化戦略は同一になります。親表のパーティションごとに、子表のパーティションが1つのみあり、子のパーティション化戦略は、主キーと外部キーの関係を通じてのみ定義されます。特定の主キー値のすべての子レコードが、親レコードではなく、子表の"同じ"パーティションに保存されます。参照パーティション化を使用しない場合に、同じパーティション化戦略を利用するときには、親表から子表にすべてのパーティション・キー列を複製する必要があります。参照パーティション化を使用すると、パーティション・キー列を複製することなく、論理データ・モデルの親子関係を利用できるため、非正規化および領域節約のための手動のオーバーヘッドが削減されます。

また、参照パーティション化は、表の論理的な構成を変更するすべてのパーティション・メンテナンス操作を親表から子表に透過的に継承します。パーティション・ワイズ結合は、親表と子表の同一レベル・パーティションを結合すると自動的に有効になり、この操作のパフォーマンスが向上します。たとえば、親表の販売注文が注文日列でレンジ・パーティション化される場合、子表の注文アイテムは、注文日列を含みませんが、販売注文表の参照によってパーティション化できます。注文表が月別にパーティション化される場合、'March 2018'のすべての注文アイテムは、注文アイテム表の単一パーティションに保存され、親表の注文に同一レベル・パーティション化されます。'April 2018'パーティションが明示的に、またはインターバル・パーティション化によって販売注文表に追加されると、同一レベル・パーティションが注文アイテム表に透過的に追加されます。

Oracleでは、参照パーティション化、およびパーティシャル・カラム・パーティション化とインターバル・パーティション化の双方との組合せをサポートしています。自動リスト・パーティション化は、参照パーティション化との組合せではサポートされていません。

バーチャル・カラム・パーティション化

バーチャル・カラムでは、表の 1 つ以上の既存の列を使用し、メタデータの式だけを保存して、式でパーティション・キーを定義できます。バーチャル・カラムを使用したパーティション化により、ビジネス要件に対して包括的に対処できます。表の中で列として明示的に定義されないビジネス属性を使用して、オブジェクトのパーティション化戦略を定義できます。列に情報が多重定義されているのは珍しいことではありません。たとえば、10 桁のアカウント ID には、先頭の 3 桁にアカウントのブランチ情報が含まれる場合があります。拡張機能のバーチャル・カラム・パーティション化によって、アカウント ID 列を含むアカウント表は、この表のパーティション・キーになるアカウント ID 列の最初の 3 桁から派生されるアカウント・ブランチ仮想（派生）列で拡張できます。

Oracle では、他のすべてのパーティション化の拡張機能で、バーチャル・カラム・パーティション化がサポートされています。

Partition Advisor

SQL アクセス・アドバイザーは、索引、マテリアライズド・ビュー、およびマテリアライズド・ビュー・ログの推奨事項に加えて、パーティション化の推奨事項を生成します。SQL アクセス・アドバイザーによって生成される推奨事項は、実装した場合に予想されるパフォーマンス面でのメリットを示しています。生成されたスクリプトと推奨事項は、スクリプト全体または個々の推奨事項として手動で実行するか、Oracle Enterprise Manager 内のキューに送信できます。

Partition Advisor は、SQL アクセス・アドバイザーに統合されています。

パーティション化機能の概要

次の表に、使用可能なすべての基本的なパーティション化手法を示します。

基本的なパーティション化手法		
パーティション化戦略	データ分散	ビジネス・ケースのサンプル
レンジ・パーティション化	値の連続レンジ	order_date でレンジ・パーティション化された注文表
リスト・パーティション化	値の不規則なリスト	国別にリスト・パーティション化された注文表
ハッシュ・パーティション化	内部ハッシュ・アルゴリズム	customer_id でハッシュ・パーティション化された注文表
コンポジット・パーティション化 <ul style="list-style-type: none"> • レンジ - [レンジ リスト ハッシュ] • リスト - [レンジ リスト ハッシュ] • ハッシュ - [レンジ リスト ハッシュ] 	上述の基本的な手法であるレンジ、リスト、およびハッシュのうちの2つの組合せ	注文日でレンジ・パーティション化されて customer_id のハッシュでサブパーティション化された注文表 国ごとにリスト・パーティション化されて order_date のレンジでサブパーティション化された注文表 国別にハッシュ・パーティション化されて customer_id のハッシュでサブパーティション化された注文表

基本的なパーティション化手法を、次のパーティション化の拡張機能と組み合わせて使用できます。

パーティション化の拡張機能		
パーティション化の拡張機能	説明	ビジネス・ケースのサンプル
インターバル・パーティション化 <ul style="list-style-type: none"> • インターバル - [レンジ リスト ハッシュ] 	レンジ・パーティション化の拡張機能。等幅レンジを提供するインターバルによって定義される。最初のパーティションを除くすべてのパーティションは、一致するデータが到着するとオンデマンドで自動的に作成される。	'01-Jan-2013'から開始される事前に定義された日ごとのインターバルの order_date でパーティション化された注文表。
自動リスト・パーティション化	リスト・パーティション化の拡張機能。パーティションはキーワード AUTOMATIC で定義され、一致するパーティションがないままパーティション・キー値が挿入されると自動的に作成される。最初に'初期パーティション'を1つだけ作成する必要がある。	国別にリスト・パーティション化された注文表で、'GERMANY'パーティションのみが事前に作成されている。
参照パーティション化	子表のパーティション化は、主キーと外部キーの関係を通じて親表から継承される。パーティション・キーは、子表の実際の列には保存されない。	order_date でレンジ・パーティション化される（親）の注文表は、（子）の注文明細表にパーティション化技術を継承する。order_date 列は、親の注文表にのみ存在
パーティシャル・カラム・パーティション化	パーティション・キーがパーティシャル・カラムに基づく、すべてのパーティション化手法によって定義される。パーティシャル・カラムは、ディスクに保存されず、メタデータとしてのみ存在。	注文表には、顧客のアカウント番号の最初の3桁に基づく売上地域から派生するパーティシャル・カラムがある。注文表は、売上地域でリスト・パーティション化される。

結論

1997年のOracle 8.0における最初の導入以来、オラクルはリリースのたびに、Oracle Partitioningにおける新技術の追加、スケーラビリティの拡張、または管理性やメンテナンスの性能を拡大し、その機能性を強化してきました。最新リリースのOracle Databaseでも、それは変わりません。

Oracle Partitioningはすべてのユーザーを対象とし、ほとんどのデータベース・アプリケーションの管理性、パフォーマンス、および可用性を大幅に向上させることができます。パーティション化はアプリケーションに透過的なので、どんな種類のアプリケーションにも簡単に実装できます。コストと時間のかかるアプリケーションの変更は必要ありません。

Oracle Corporation

World Headquarters

500 Oracle Parkway
Redwood Shores, CA 94065,

海外からのお問い合わせ窓口

電話 : +1.650.506.7000

ファクシミリ :

+1.650.506.7200USA

Connect with us



blogs.oracle.com/oracle



facebook.com/oracle



twitter.com/oracle

Integrated Cloud Applications & Platform Services

Copyright © 2019, Oracle and/or its affiliates. All rights reserved. 本文書は情報提供のみを目的として提供されており、ここに記載されている内容は予告なく変更されることがあります。本文書は、その内容に誤りがないことを保証するものではなく、また、口頭による明示的保証や法律による黙示的保証を含め、商品性ないし特定目的適合性に関する黙示的保証および条件などのいかなる保証および条件も提供するものではありません。オラクルは本文書に関するいかなる法的責任も明確に否認し、本文書によって直接的または間接的に確立される契約義務はないものとします。本文書はオラクルの書面による許可を前もって得ることなく、いかなる目的のためにも、電子または印刷を含むいかなる形式や手段によっても再作成または送信することはできません。

Oracle および Java は Oracle およびその子会社、関連会社の登録商標です。その他の名称はそれぞれの会社の商標です。

Intel および Intel Xeon は Intel Corporation の商標または登録商標です。すべての SPARC 商標はライセンスに基づいて使用される SPARC International, Inc. の商標または登録商標です。AMD、Opteron、AMD ロゴおよび AMD Opteron ロゴは、Advanced Micro Devices の商標または登録商標です。UNIX は、The Open Group の登録商標です。0219

ホワイト・ペーパー Oracle Partitioning

2019年2月 2019年2月

著者：[適宜入力]

共著者：[適宜入力]



Oracle is committed to developing practices and products that help protect the environment