



ORACLE

Oracle Partitioning

卓越したデータ管理とパフォーマンス

2023年12月、バージョン1.0

Copyright © 2023, Oracle and/or its affiliates

公開

本書の目的

本書は、Oracle Partitioningの機能と機能強化の概要を示すものです。本書は、Oracle Databaseに関するビジネス上の利点の評価と、説明した製品機能の実装の計画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。本書は、ユーザーとのライセンス同意書の一部をなすものではなく、またオラクルやその子会社および関連会社とのいかなる契約上の合意事項にも含まれるものではありません。

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ください。本文書に記載されている機能の開発、リリース、時期および価格については、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上、本書に記述されているすべての機能を安全に組み込むことができず、コードの不安定化という深刻なリスクを伴う場合があります。

目次

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	9
パーティション化による情報ライフサイクル管理	9
パーティション化戦略	10
パーティション・オブジェクトのデータ分散方法	10
シングル（1つの）レベルのパーティション化	11
コンポジット（2つの）レベルのパーティション化	11
パーティション化の拡張機能	12
インターバル・パーティション化	12
自動リスト・パーティション化	12
参照パーティション化	12
バーチャル・カラム・パーティション化	13
自動パーティション化	13
Partition Advisor	13
パーティション化機能の概要	14
まとめ	15

図表一覧

図1：アプリケーションとDBAから見た、パーティション表	5
図2：パーティション表の索引付け	6
図3：パーティション・ワイズ結合の例	8
表1：使用できる基本的なパーティション化手法	14
表2：パーティション化の拡張機能	15

はじめに

30年近く開発されてきた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の1つとして確立されています。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。この物理パーティション化の知識により、データベースはどのようなアプリケーションでもパフォーマンス、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、制約に囚われないプルーニング機能が実現します。表内のパーティションに関する知識に加えて、表とパーティションをさらに小さい物理ゾーンに分割して、きめ細かいデータ・プルーニングに使用できます。

パーティション化で"階層型アーカイブ"を使用することにより、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。また、ハイブリッド・パーティション表では、同じ論理表内に内部ストレージと外部ストレージを生成することもできるため、ILMを次のレベルに引き上げることができます。特にOracle Autonomous Databaseなどのクラウド環境では、クラウド内で生成されてObject Storageに格納されるデータをデータベース内から簡単に統合して処理できます。この際、処理する前にすべてのデータをデータベースにロードする必要はありません。

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、可用性を改善しています。誰もがOracle Partitioningからメリットを得ることができます。皆様もそうです。

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、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。

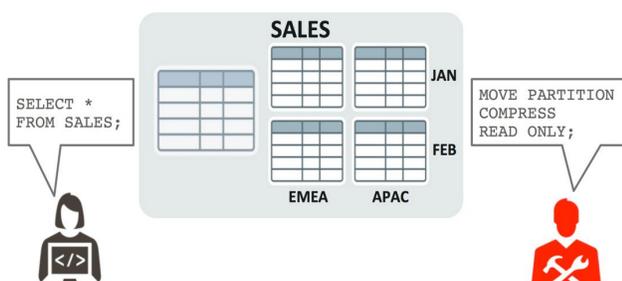


図1: アプリケーションとDBAから見た、パーティション表

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、指定された行ごと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。この表は、アプリケーションからは1つの‘通常’の表として見えます。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、格納でき、データの重要度と使用頻度に応じてデータ格納を最適化することができます。古いデータ・レンジを

格納するパーティションは、表圧縮を使用して異なるストレージ階層に格納できます（または読取り専用の表領域に格納することも、読取り専用パーティションとマーク付けすることも可能です）。対照的に、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格納するようにマーク付けされます。ハイブリッド・パーティション表の場合は、一部のパーティションが内部ストレージ内に存在すると同時に、その他のパーティションが外部ストレージ内に存在しますが、これらはすべて同じ論理表にあります。

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。指定された行のデータ配置は、両方のパーティション・キーの基準（パーティションとサブパーティションのキー列）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。

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である、ファイル・システムやオブジェクト・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い、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を問わず読取り専用です。

ハイブリッド・パーティション表では、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概念が組み合わされています。この名前が示唆するように、このようなパーティション表の場合、（データベース内の）内部パーティションと（データベース外部の物理ストレージ上の）外部パーティションの両方を使用できます。これと同じルールが、パーティション化された外部表のパーティションに対してではなく、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適用されます。すなわち、データ配置はデータベースによって行われず、このようなパーティションのコンテンツは読取り専用です。

¹説明をわかりやすくするため、本書では以降、パーティションのみについて言及します。

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はありません。それでもなお、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。これによってランタイム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。

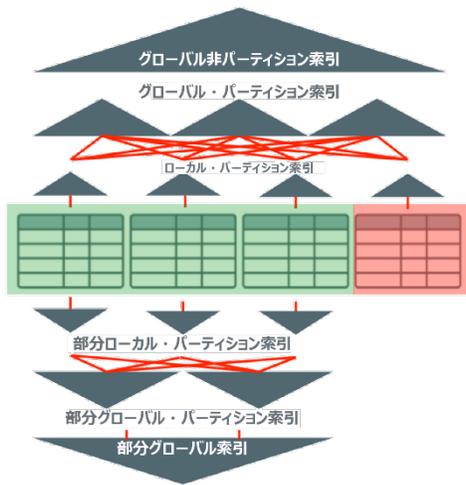


図2: パーティション表の索引付け

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を使用す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、基本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を索引付けから除外することで、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、あらゆる種類の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。

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、基本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、その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。分析用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。

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。

³ 一意索引を部分索引にすることはできません。

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、特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さえ扱います。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・プルーニング（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・プ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・プルーニングは、オラクルが提供する製品の他の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・プルーニングは、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・メタデータを超えてオラクルのプルーニング機能を拡張します。データ・プ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・キー列に関連付けられてい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さらには完全パーティションに対する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内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・マップを所有する表自体の列）と結合列を指定できます。

データベース内でゾーン・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・キーが出荷日では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。しかし、出荷日（パーティション・キー列ではありません）が含まれるゾーン・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も含まれているため、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・キー列の注文日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。

⁴ ゾーン・マップの詳細で総合的な説明については、『Oracle Databaseデータウェアハウス・ガイド』を参照してください。

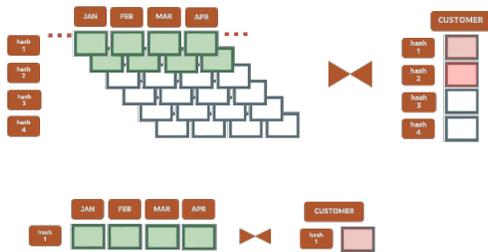


図3：パーティション・ワイズ結合の例

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。すでに物理的に同一レベル・パーティション化された結合する表のファクトを使用するか、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同一レベル・パーティション化されたデータセットを作成すると、全体の結合時間が短縮され、少なく済みます。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。

自動パーティション化は、Oracle Autonomous Databaseの一部で、データベースを利用して、ワークロードのパフォーマンスを高めるために最適なパーティション化戦略を選択できます。ユーザーの介入なしで、表と索引のパーティション作成が分析および自動化されます。ユーザーによる操作やメンテナンスは必要ありません。

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。Oracle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、2024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、保存されたデータの総所有コストをさらに削減できます。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。

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 2024'、'February 2024'、'March 2024'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 2024'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。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。その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。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、グローバル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。

⁵ 非同期グローバル索引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』を参照してください。

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年超が経過したデータをすべて削除する必要があるが、正式にクローズしていない注文は削除できないとします。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。この状況にプログラムによって対処するには、外れ値（まだクローズしていない注文）を保持する必要があります。パーティション・メンテナンス操作を使用すると、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ます。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・メンテナンス操作とパーティション・メンテナンス操作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1つのパーティション表にしたり、完全なオンライン方法でパーティション化戦略を変更したりできます（既存のすべての索引の変更を含む）。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合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。

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表に対して引き続き実行できるため、使用できないパーティションへアクセスする必要がなければ、これらのデータベース操作の実行は成功します（操作で使用できないデータにアクセスしようとする場合、このような操作は明らかに失敗します。Oracleでは、どのような操作であれ、trueと有効な結果のみが返されます）。

データベース管理者は、各パーティションを別の表領域に格納するよう指定したり、古い静的データが含まれる表領域を読み取り専用を設定したりできます。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）パーティションまたは一連の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。また、表領域の全体バックアップを実行する必要がない場合もありますが、今後のバックアップから読み取り専用表領域をすべて除外することもできます。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、データの重要性とアクセス方式は、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（圧縮やデータが読み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。

⁶ フィルタ付きパーティション・メンテナンス操作では、フィルタ条件をパーティション表だけに適用でき、結合またはその他の複雑なSQL構文はサポートされません。

ハイブリッド・パーティション表の場合、古いパーティションの一部が外部ストレージ上のOracleデータベース外に存在することもあります。ファイル・システム上に存在することや、最近ではオブジェクト・ストア内に存在することもより一般的です。このようなデータは本質的に読み取り専用です。

内部パーティションの場合、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み取り専用表領域に加えて、個々のパーティションを読み取り専用を設定できます。パーティションを読み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、読み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み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注文表の場合、最新の四半期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年分のデータをデータベース外の外部パーティションとして格納し、その後2年分のデータを読み取り専用パーティションとして格納することができます。ここ数年間のデータのみを変更でき、その他の古いデータはすべて不変でありながら、一部のデータがデータベース内に保存されていない場合でも、依然として規制遵守のためにシステム内から使用できます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削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-2024'のパーティションには、'01-JAN-2024'から'31-JAN-2024'のパーティション・キーの値を持つ行が含まれます）。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、より小さ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。

⁷ 外部パーティションは、データベース内からは読み取り専用ですが、データベースにはそのコンテンツに対する制御権はなく、データがデータベース外から変更されないことも保証できないことに注意してください。オラクルでは、読み取り専用の内部パーティションについてのみ、パーティションが読み取り専用設定されている場合に、このようなパーティションのデータを変更できないことを保証できます。

⁸ ADOは、現在のところ、内部（データベース管理の）ストレージ層のみをサポートしています。

- **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。
- **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・レイヤーで管理する必要があります（アプリケーションでパーティション・ブルー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のみ制御されます（興味深い事実：参照パーティション化はシステム・パーティション化の上層に構築されます）。

シングル（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を使用します⁹。

コンポジット（2つの）レベルのパーティション化

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。

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管理されるヒープ表でのみサポートされています。

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、主にOLTP環境で使用されます。

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管理される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同一レベル・パーティション化の範囲を定義して、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、ユーザーが操作しなくても、最初にパーティション表のパーティションが1つのみ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・パーティション化された表を定義できます。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。この際、親表のパーティション・キー列を子表内に物理的に格納する必要はありません。親表のパーティションごと、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 2024'の注文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 2024'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と、バーチャル・カラム・パーティション化およびインターバル・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、メタデータとしての式のみを保存して、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、その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。

自動パーティション化

Oracle Autonomous Databaseの自動パーティション化は、既存のインターバル手法と自動リスト手法の拡張機能であり、内部実装を使用して、ユーザーによる操作やメンテナンスが必要ないよう徹底します。データベースにより、大規模な表や索引が完全に自律的に管理されます。自動パーティション化では現在、単一系列のパーティション・キーを単一レベルのパーティション化と組み合わせて使用しています。このパーティション化手法は外部に公開されていないため、パーティション・メンテナンス操作は現在、自動パーティション化では許可されていません。

Partition Advisor

SQLアクセス・アドバイザは、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	注文日でレンジ・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	顧客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	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	order_date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 国別にリスト・パーティション化されて注文日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顧客IDのハッシュでサブパーティション化された注文表
<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] 		

表1: 使用できる基本的なパーティション化手法

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注文日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。	注文日でレンジ・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。注文日列は、親の注文表にのみ存在
バーチャル・カラム・パーティション化	パーティション・キーがバーチャル・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。バーチャル・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バーチャル・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。
自動パーティション化	Autonomous Databaseは、ワークロードを分析し、実際のワークロードに基づいて、インターバルまたは自動リストなどの内部パーティション化アルゴリズムを実装する。	データベースのサイズとワークロードにより、適用可能な自動パーティション化戦略が定義される（存在する場合）。

表2.パーティション化の拡張機能

まとめ

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。これは、Oracle Databaseの最新リリースも同様です。

Oracle Databaseは、Autonomous Databaseおよび自動パーティション化により、お客様のためにワークロード分析とパーティション化に関する意思決定を引き継ぎ、作業環境のパフォーマンスを改善します。ユーザーによる操作やメンテナンスは一切必要ありません。

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。

Connect with us

+1.800.ORACLE1までご連絡いただくか、oracle.comをご覧ください。北米以外の地域では、oracle.com/contactで最寄りの営業所をご確認いただけます。

 blogs.oracle.com  facebook.com/oracle  twitter.com/oracle

Copyright © 2023, Oracle and/or its affiliates. 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、Java、MySQLおよびNetSuite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。