



EDB

Postgres® for the AI Generation

Migration Handbook

リリース 2025-05-12

EnterpriseDB Corporation

2025 年 05 月 12 日

目次

第 1 章 Migration Handbook	1
1.1 Factors to consider when migrating	1
1.1.1 スキーマの移行の考慮事項	2
1.1.2 データ移行の考慮事項	2
1.1.3 インフラストラクチャの考慮事項	3
1.1.4 アプリケーションの移行の考慮事項	4
1.1.5 運用上の考慮事項	4
1.1.6 サイズ設定と構成の考慮事項	5
1.1.7 ベストプラクティスの考慮事項	5
1.2 Migration techniques	6
1.2.1 データベースを変換するための手動アプローチ	6
1.2.2 データベース移行への自動アプローチ	6
1.3 The database migration “journey”	7
1.3.1 1. 移行することを決定する	7
1.3.2 2. 実現可能性と代替案を分析する	8
1.3.3 3. 移行を計画する	9
1.3.4 4. データベーススキーマ、コード、およびデータを移行する	10
1.3.5 5. インターフェイスとアプリケーションを移行する	11
1.3.6 6. レポートと管理ツールを移行する	12
1.3.7 7. 移行のテスト	13
1.3.8 8. 移行後のシステムの最適化と構成	14
1.3.9 9. 完全なカットオーバー/ゴーライブ	14
1.4 EDB capabilities for the migration journey	15
1.4.1 移行作業で使用する機能	15
1.4.2 EDB 移行ソフトウェアとサービス	16
1.5 Comparison of EDB Postgres Advanced Server with Oracle	23
1.5.1 Comparison of general characteristics	23
1.5.2 Comparison of database features	25
1.5.3 Comparison of database operations-related capabilities	30
1.5.4 Comparison of application development capabilities	37
1.5.5 Comparison of nonrelational data support	40
1.5.6 Notable differences	41

第1章 Migration Handbook

組織は、次のようなさまざまな理由で、Oracle アプリケーションを他のデータベーステクノロジーを使用するように移行します。

- コストの削減
- ライセンスの柔軟性
- オープンソースの取り組み
- アプリケーションとテクノロジーを最新化および統合するための取り組み
- 新しいプラットフォームと展開環境に移行したいという欲求

Postgres は、その豊富な機能、品質、および強力な開発コミュニティにより、Oracle からの移行のターゲットデータベースとしてよく選択されます。EDB は、Oracle とのビルトイン互換性を備えた PostgreSQL のディストリビューション、EDB Postgres Advanced Server、および組織がより簡単にビジネスへの影響を減らして Oracle から移行するのを支援することを目的とした一連の移行ツールとサービスを提供します。

このドキュメントの目的は、Oracle から EDB Postgres Advanced Server への移行を計画および実行するのに役立つことです。移行を実行する際に考慮すべき要素、移行の手順、および移行を支援するために EDB が提供する機能、ツール、サービスを特定および説明します。また、Oracle と EDB Postgres Advanced Server の主要な類似点と相違点、および既知の相違点または非互換性の回避策についても説明します。

1.1 Factors to consider when migrating

移行の分析と計画を行うときは、次の要素を検討する必要があります。

- スキーマ
- データ
- インフラストラクチャ
- アプリケーション
- オペレーション
- サーバーのサイジング

1.1.1 スキーマの移行の考慮事項

スキーマはデータベースの構造を定義します。スキーマには、テーブルと、シーケンス、インデックス、制約、ビューなどの他の構造の定義が含まれます。これらのオブジェクトを作成するために Postgres が提供する構文は、多くの場合 Oracle の構文と似ているまたは同一ですが、移行時に対処する必要がある違いがあります。Oracle と Postgres がスキーマオブジェクトを編成および保存する方法にも違いがあるため、理解することが重要です。

データ型は、テーブルおよびデータベースで使用するその他のオブジェクトで使用されます。移行の場合、Oracle のデータ型と Postgres のデータ型をどのようにマッピングするかについて考える必要があります。

データベーススキーマには、パッケージ、プロシージャ、ファンクション、トリガーを含むコードオブジェクトも含まれており、データベースに直接書き込まれます。Oracle では、コードオブジェクトは Oracle の PL/SQL 言語で作成されます。PostgreSQL には、Oracle のものと互換性のない独自のビルトインプログラミング言語があります。ただし、EDB Postgres Advanced Server は、SPL として知られる PL/SQL のビルトイン実装を提供します。EDB の SPL は、最も一般的に使用される Oracle PL/SQL コンストラクトの大部分をカバーしていますが、Oracle コンストラクトの完全なセットを実装しているわけではありません。その結果、EDB Postgres Advanced Server で動作するように一部の Oracle コードオブジェクトを変換する必要があります。

多くの場合、ソースデータベースとターゲットデータベースがデータと対話する方法には構文の違いがあります。Oracle には、Postgres でサポートされていない情報を選択、挿入、更新、削除するために構造化照会言語 SQL のバージョンで使用する特定の規則とキーワードがあることに注意する必要があります。

1.1.2 データ移行の考慮事項

スキーマは構造を提供しますが、データはデータベースのコンテンツです。検討する必要があるのは、データをターゲットデータベースに移動する方法です。この方法は、移動する必要があるデータの量と、どれだけのダウンタイムが許容できるかによって異なります。大量のロードが自分にとって適切であるかどうか、つまりソースシステムからデータのスナップショットを取得し、ターゲットシステムに転送し、そこから続行するかどうかを判断する必要があります。または、データをターゲットデータベースに移行している間、ソースアプリケーションとデータベースを実行したままにする必要がある場合があります。ソースデータベースはまだ更新を受信しているため、そのデータをターゲットデータベースに継続して移動できるようにする必要があります。

フォールバックは別の考慮事項です。移行プロセスを進めると、最終的に移行が完了します。移行したシステムのテストを最終決定するときは、移行が完全に成功しなかった場合に備えて、フォールバックポジションが必要になることがよくあります。古いデータベースが、移行されたデータベースに適用されるデータの変更との同期が維持されている場合、移行したシステムで操作を開始した後、元のソースデータベースとソースアプリケーションに簡単にフォールバックできます。フォールバック計画では、移行したデータベースに入力された新しいデータがタイムリーに元のソースデータベースに戻せることを保証する必要があります。

データを移行する場合、ソースデータベースからターゲットデータベースにデータをプッシュするために提供するコアツールに加えて、より複雑な抽出-変換-ロード ETL ニーズなど、より高度なデータ移行シナリオをサポートするツールを使用することもできます。これらのツールは、ソースデータベースからデータを取得して変換し、ターゲットデータベースにロードします。

データがターゲットシステムに配置されたら、ターゲットシステムに配置したデータがソースシステムのデータと一致することを確認する必要があります。

1.1.3 インフラストラクチャの考慮事項

データベースはインフラストラクチャ上にあるため、ソースデータベースのホスティング環境とターゲット環境がどのようなものかを検討する必要があります。たとえば、現在のデータベースがオンプレミスで展開されている場合、同様にオンプレミスで実行されている別のデータベースに移行することを決定する場合があります。しかし、ますます多くの組織が、データベースのオンプレミス展開からクラウドベースの展開への移行を検討しています。クラウドベースの展開が最終的な目標である場合、オンプレミスで実行されているデータベースからクラウドで実行されているデータベースに直接移行することにする場合があります。または、最初にデータベースをオンプレミスで実行されているものに移行し、次に、移行したデータベースをクラウドに移動することもできます。

最終状態またはクラウドへの足がかりとしてオンプレミスデータベースに移行する場合、移行を実行およびテストし、運用データベースと関連ツールをホストするためのサーバーリソースを手元にあることを確認する必要があります。クラウドに移行する場合、ニーズは同様ですが、移行プロセスをサポートするために、必要に応じてサーバーを追加および削除する柔軟性が高くなります。

考慮すべきもう1つの要素は、多くの古いアプリケーションデータベースが専用のハードウェアとオペレーティングシステムに展開されていることです。サーバーオプションをターゲットとする可能性が高いことを考えると、この要素は、移行に使用されるツールの種類とアプローチに影響を与える可能性があります。使用する予定のツールがこれらのシステムに展開できるか、それらにアクセスして移行する情報を取得できることを確認する必要があります。

オンプレミスリソースへの移行を検討する場合、EDB Postgres Advanced Server で使用可能な展開オプションは次のとおりです。

- 物理サーバー
 - 仮想マシン
 - Kubernetes のクラウドネイティブの Postgres コンテナ
 - プライベートクラウドインスタンス

通常、データベースは物理サーバーにインストールして、パフォーマンスを最適化します。時間の経過とともに、仮想化テクノロジーが向上するにつれて、データベースが仮想マシンにインストールされることがより一般的になりました。ここ数年、コンテナはデータベース展開の実行可能なオプションとして浮上し始めており、特に業界がオーケストレーションフレームワークとして Kubernetes で多かれ少なかれ標準化が進み、EDB のようなデータベースベンダーがコンテナにデータベースを展開および維持するための Kubernetes オペレーターを開発しているため、。

クラウドへの移行の場合、以下は EDB Postgres Advanced Server の展開オプションです。

- BigAnimal クラスター
- クラウドベースのサーバーインスタンスにインストールされた自己管理 EDB Postgres Advanced Server
- クラウドネイティブ PostgreSQLCNP

ますます多くの組織が、管理対象データベースサービス DBaaS としても知られ、このサービスが企業に提供するメリットを利用することに関心を持っています。BigAnimal は、主要な商用クラウドで利用可能な EDB Postgres DBaaS オファリングであり、PostgreSQL または EDB Postgres Advanced Server データベースクラスターを簡単に展開するオプションを提供します。クラウドに移行したいが、データベースインスタンスと基になるサーバーの管理をより詳細に制御したい場合は、クラウドベースのサーバーインスタンスに EDB Postgres Advanced Server を自分でインストールおよび管理できます。EDB のクラウドネイティブ PostgreSQLCNP を使用すると、Kubernetes オーケストレーションコンテナに PostgreSQL または EDB Postgres Advanced Server クラスターを展開および管理できます。データベース用のコンテナの広範な採用は、まだ初期段階です。しかし、アプリケーションがマイクロサービスに移行するにつれて、データベース用のコンテナの使用が増加しています。Kubernetes クラスターは、クラウドで立ち上げることができます。実際、大手クラウドプロバイダーもマネージド Kubernetes サービスを提供しています。

異なるソースインフラストラクチャとターゲットインフラストラクチャがある場合、インフラストラクチャに基づいてソースシステムを最適化している可能性を検討してください。ターゲットシステムに移動するときは、別の手法を使用して同様の最適化を実行する必要がある場合があります。

1.1.4 アプリケーションの移行の考慮事項

アプリケーションは、データベースに情報を送信し、データベースから情報を取得します。Oracle や Postgres などのリレーショナルデータベース管理システム RDBMS バックエンドを使用するほとんどのアプリケーションは、構造化照会言語 SQL でデータベース照会を発行します。SQL 標準の重要なコンポーネントは、すべての主要なリレーショナルデータベースに採用されました。ただし、それぞれは標準から逸脱し、互いに異なる機能を持っています。そのため、移行作業の一部として、静的または動的に生成された、アプリケーションコードに埋め込まれた SQL コードの何らかの変換が必要になる可能性があります。

さらに、アプリケーションは通常、古いデータベースシステムとオペレーティング環境で動作するように適切にパフォーマンスするように調整および構成されます。古いシステムから新しいシステムに移動するときは、新しいデータベースとオペレーティング環境に対する正常な実行をサポートするために、アプリケーションにいくつかの変更を加える必要がある場合があります。

1.1.5 運用上の考慮事項

古いデータベースを新しいデータベーステクノロジーに移行すると、新しいデータベースと連携するように運用プロセスとツールを更新する必要があります。Oracle から Postgres への移行の場合、使用されている Oracle 固有のツールは Postgres 固有のツールに置き換える必要があります。考慮すべき主な運用面には、高可用性、バックアップとリカバリー、監視と管理、セキュリティ、暗号化が含まれます。

また、古いデータベースが古い専用のハードウェアとオペレーティングシステムで実行されている場合、データベースの移行には、基になるホストシステムを最新のプラットフォームとオペレーティングシステムに移行することも必要になる場合があります。オンプレミスの移行の場合、これにより、新しいサーバープラットフォームとオペレーティングシステムで動作するように運用とメンテナンスの手順とツールを変更する必要がある場合があります。クラウドへの移行の場合、運用とメンテナンスアクティビティの一部はクラウドサービスプロバイダーが処理しますが、その他は引き続きお客様の責任です。おそらく、これらのアクティビティには、現在使用しているものとは異なるツールと手順が必要です。

1.1.6 サイズ設定と構成の考慮事項

「どれくらいの大きさのサーバーが必要ですか?」

この質問は、Oracle から Postgres への移行を検討するときによく起こります。最短の答えは「場合によって」ですが、これは良い答えではありません。より良い方法は、「1:1 対応つまり同じ数のコア、RAM など」で開始し、そこから行くことです。」

それはなぜでしょうか？

Postgres と Oracle は、内部的には大きく異なり、SQL プランニングが異なり、ストレージとアクセスパターンが異なります。バージョン管理とトランザクションの分離でさえ、別の方法で実装されます。そのため、Oracle の構成とアーキテクチャの考えられるすべての順列をカバーする単純な式を設定することが困難になっています。すべてのデータベースは、実行中のワークロードに対してそれぞれ異なる応答をします。

EDB Postgres Advanced Server に移行した EDB の顧客は、同様のハードウェアを使用する場合、Oracle と同等のパフォーマンスを見つけました。場合によっては、ワークロードによっては、Oracle を上回るパフォーマンスを発揮する場合があります。移行したすべてのお客様は、少なくとも Oracle のパフォーマンスに匹敵し、可能であれば、同等のハードウェアで、または必要に応じてリソースを追加することにより、それを超えることを目指していました。また、特定のワークロードを使用してベンチマークを実行して、システムがパフォーマンスの期待を満たしていることを確認し、それに従って生産を計画することをお勧めします。

1.1.7 ベストプラクティスの考慮事項

次に、Oracle から Postgres データベースアプリケーションの移行を計画する際に考慮するベストプラクティスをいくつか示します。

- 移行手順の手順をスキップまたは流し読みしないでください。
- 計画を立てます。
- 機能的要件と非機能要件を特定して理解します。
- 早期にソリューションデザインを完了して、IT リソースのニーズを理解します。
- 開発環境とテスト環境を使用して、運用環境に移行する前に、移行の問題を特定および解決します。
- 移行の早い段階でアプリケーション開発者とシステム管理者を参加させて、データベースチームと協力します。
- 運用環境と使用法に近い方法で、データベースを使用してアプリケーションをテストできる環境をセットアップします。
- データベーススキーマを移行する場合、データの移行後までインデックスと制約のロードを遅延します。
- より複雑な移行については、移行の専門家と PostgreSQL の専門家の助けを借ります。
- 移行後の操作の前に、運用担当者が新しいシステムに関するトレーニングを受けていることを確認します。

1.2 Migration techniques

高レベルでは、Oracle から Postgres へのデータベースの移行には以下が含まれます。

- Oracle の SQL 標準の拡張バージョンから、一般的に標準への準拠のより高い Postgres 準拠バージョンへのスキーマの変換
- Oracle 固有のデータ型定義を Postgres 固有のデータ型定義に変換する
- クエリーとストアドプロシージャの書き換え
- データのコピー
- Oracle が独自の拡張機能で標準プロトコルを拡張したため、Postgres JDBC、.NET、ODBC ドライバーを使用するようにアプリケーション API を更新
- 移行したデータベースが、パフォーマンス、管理性、高可用性、およびエンタープライズセキュリティ要件との統合に関連するすべての非機能要件を満たしていることを確認します。

移行は手動または自動で実行できます。

1.2.1 データベースを変換するための手動アプローチ

これらの変換を手動で実行すると、費用がかかり、エラーが発生しやすくなります。ビジネスロジックと非常に単純なアプリケーションロジックを持たない非常に小規模なデータベースを除き、このアプローチは現実的ではありません。

1.2.2 データベース移行への自動アプローチ

通常、純粋な手動アプローチの代わりに2つの自動アプローチが使用されます。

`_native stable_`アプローチは、あるデータベースの機能を拡張し、API やプロトコルを含む SQL 標準の特定の拡張機能のネイティブ実装を作成します。EDB Postgres Advanced Server は、次の領域で Oracle との互換性を提供する PostgreSQL の強化バージョンです。

- Oracle 固有および構文互換のデータベースオブジェクトタイプ
- Oracle 固有のデータ型
- Oracle 固有の SQL 拡張機能
- ビルトインネイティブプロシージャル言語としての Oracle PL/SQL のサポート
- Oracle データディクショナリビューつまり ALL_、DBA_、USER_ビュー
- Oracle ビルトイン PL/SQL パッケージ
- Oracle のようなデータベースドライバー
- DBA 用の Oracle 同等ツール EDB Postgres Advanced Server は、これらの領域の Oracle 固有の機能をすべて完全に実装しているわけではありません。それにもかかわらず、互換性は広範であり、最も一般的に使用される Oracle コンストラクトの多くをカバーしています。その結果、実装された互換

性機能により、Oracle からのほとんどの移行が簡単になります。_translation_アプローチでは、自動ツールを使用して、Oracle 定義、クエリー、およびストアドプロシージャを Postgres 互換バージョンに書き換えまたは変換します。EDB 移行ポータルは修復ハンドラーは、このアプローチを使用して、EDB Postgres Advanced Server に互換性のある実装を持たないコンストラクトを含む多数のオブジェクトを自動的に変換します。変換アプローチは、オープンソース移行ツール、および純粋なオープンソースの PostgreSQL をターゲットとする他のベンダーが提供するツールで使用可能な唯一のアプローチです。ただし、翻訳する必要があるオブジェクトの数によっては、純粋なオープンソースソリューションに到達するまでに必要な時間と労力が、スケジュールまたはリソースでサポートされていない場合があります。

ほとんどの移行は、これら 2 つの自動化アプローチを組み合わせることにより大幅に簡素化されますが、通常は手動変換が必要な一部のオブジェクトがまだ残っています。Migration Portal ナレッジベースの回避策を参照するか、EDB のプロフェッショナルサービスに連絡して、残っている手動変換の解決策を特定します。

1.3 The database migration “journey”

データベースの移行は、9 つのステップの「ジャーニー」で構成されています。

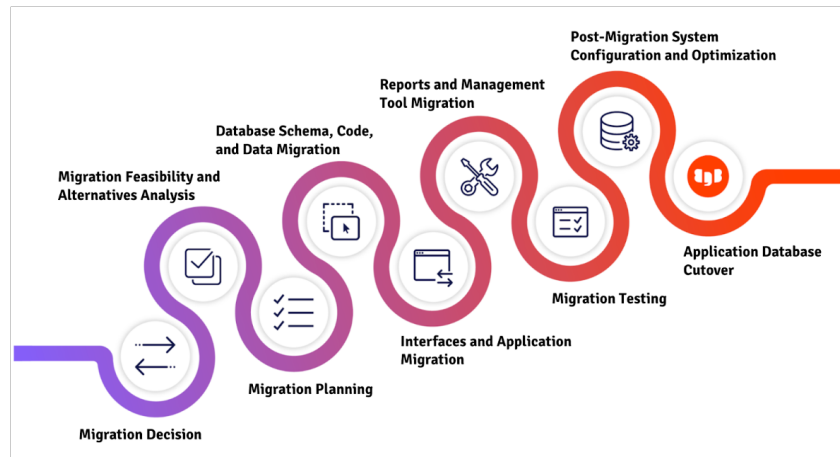


図 1: Migration journey

1.3.1 1. 移行することを決定する

最初に、組織は移行することを決定する必要があります。古いデータベースシステムにお金を費やしすぎていると判断する場合があります。現在のデータベーステクノロジーで十分にサポートされていないインシニアチブがある可能性があります。または、クラウドに移行するか、オープンソースソフトウェアを利用したいと考えている場合もあります。これらは、移行のビジネスケースにつながる可能性のある一般的な要因です。

ビジネスケースがあり、組織がそれに同意したら、組織は移行の実行にコミットする必要があります。コミットメントには、移行を実行するための資金とリソースが含まれます。

1.3.2 2. 実現可能性と代替案を分析する

代替案を分析し、特定のデータベースへの移行の実現可能性を検討します。

アプリケーションポートフォリオを確認して、すべてのアプリケーションを移行するかどうかを判断します

このレビューの一部として、移行可能な古いデータベースを持つアプリケーションを確認します。移行の優先順位を IT 戦略と調整して、これらが移行を検討する十分に重要なビジネスアプリケーションであることを確認します。

アプリケーションポートフォリオを確認する一環として、一部のアプリケーションをリタイアしてもよいと決定する場合があります。ポートフォリオには、移行したいが、既存のデータベースでのみサポートされているためできないアプリケーションが存在する可能性があります。

移行するアプリケーションの候補セットを特定するときは、それらを移行したいということは、移行を完了するという組織のコミットメントに対応している必要があることに注意してください。時間、リソース、予算が必要です。

各アプリケーションの移行ターゲットとしてどのデータベーステクノロジーが適切な選択であるかを評価します

多くの場合、EDB Postgres Advanced Server は、Oracle に代わる理想的な選択です。ただし、一部のアプリケーションには、他のソリューションで対処したほうが良い可能性のある要件があります。

ターゲットデータベーステクノロジーの適切なセットを選択した後、後続の手順は、移行がどの程度複雑になるかについてのより詳細な評価を実行することです。EDB Postgres Advanced Server が潜在的なターゲットである場合、EDB 移行ポータルを使用して、データベース内の Oracle スキーマおよびプロシージャオブジェクトと EDB Postgres Advanced Server との互換性を評価できます。この情報は、旅の次のステップで重要です。

アプリケーションの優先順位付け

大規模なアプリケーションの移行を検討している場合は、第 1 レベルパスを実行して、最初に移動するアプリケーションを特定します。移動する方がリスクが高いほど重要なものもありますか？また、移行の難易度を確認し、それに基づいて優先順位を付けることもできます。

移行のターゲットとするホスティング環境を決定する

EDB Postgres Advanced Server データベースをホストするためのオプションの詳細については、[インフラストラクチャの考慮事項](#)を参照してください。現在オンプレミスで実行されていますか？オンプレミスに残りますか、クラウドに移行しますか？それとも、クラウド上にありますが、おそらくそれがうまくいかず、オンプレミスまたは別のクラウドプロバイダーに移動する必要がありますか。

潜在的な移行の問題を特定する

ソリューションのより徹底的なレビューを実行して、旅の後半で発生する特定の互換性問題に対処しますが、この初期段階で潜在的な移行問題を特定することも重要です。Oracle から EDB Postgres Advanced Server の移行の場合、詳細については [Comparison of EDB Postgres Advanced Server with Oracle](#) を参照してください。

このステップの主な目標は次のとおりです。

- どのアプリケーションが移行の適切な候補であるかを理解します。
- 移行のためのさまざまなテクノロジーとベンダーオプションを分析します。
- すべての移行を完了するまでの全体的なレベルの労力、時間、およびコストの全体像を理解します。

また、現在のテクノロジーとプラットフォームを使用し続ける場合と、新しいテクノロジーとプラットフォームに移行する場合のコストも検討します。

1.3.3 3. 移行を計画する

移行の計画には、次のプロセスが含まれます。

- 移行するアプリケーションの優先順位付け。移行するアプリケーションを決定したら、移行する順序に優先順位を付けます。組織は、最も困難な移行を実行できれば、他のすべての移行が簡単になると考えているため、最も困難な移行を選択することを検討することがよくあります。ただし、通常は、自分にとって中程度に重要であり、必ずしも顧客にとって最も価値のあるものではなく、最も難しいものではないものを選択することをお勧めします。このようにして、プロセスから学ぶことができます。次に、最初の移行などから学んだことに基づいて次のものを選択し、最終的に最も価値があり最も難しいものに到達します。
- パフォーマンス、可用性、管理、統合などの非機能要件の特定。
- 移行アーキテクチャと移行後アーキテクチャの両方を含む高レベルでのソリューションの設計。たとえば、データベース、プラットフォーム、移行の種類、高可用性要件などを選択します。
- 移行作業の見積もり。分析から、移行の実行にどのくらいの時間とリソースが必要かがわかります。
- 移行環境の準備。必要なソフトウェアを入手してインストールし、サーバー間の接続を確立します。

この手順に十分な注意を払うと、成功の可能性が高まり、移行プロセスの後半での不要な予期せぬ事態の数を減らすことができます。ソースシステムのアーキテクチャと実装に関する十分な知識を持つ人を含む移行チームを編成します。

1.3.4 4. データベーススキーマ、コード、およびデータを移行する

アプリケーションデータベースの移行プロジェクトに関して、これは多くの場合最初に思い浮かぶ手順です。この手順では、実際のデータベースがソースシステムからターゲットシステムに移行します。データベースを移行するには、次の手順を実行する必要があります。

スキーマを移動する

スキーマは、データがデータベースに保存および編成される構造を提供します。そのため、最初にスキーマを移動する必要があります。

スキーマを移行する場合、通常、Oracle データベースに存在するスキーマオブジェクトとターゲットデータベースで作成できるオブジェクトに違いがあります。Oracle 機能に対する EDB Postgres Advanced Server の互換性のため、他のターゲットデータベースオプションでは対処できない多くの互換性の問題が自動的に対処されます。

この手順の主なタスクは、互換性がないまたは自動的に変換できないスキーマオブジェクトの回避策を実装することです。スキーマオブジェクトは、SQL ベースのデータ定義言語 DDL を使用して作成されます。EDB の移行ポータルを使用して、Oracle DDL と EDB Postgres Advanced Server の互換性を評価できます。また、評価プロセスの一部として適用される repair handlers を介していくつかの自動変換も実行します。

移行ポータル評価を実行した後の次のステップは、DDL 移行の結果を確認し、評価レポートで特定された問題を解決することです。

データベース機能つまりデータベースプロシージャルオブジェクトを移行する

これには、データベースが使用するアプリケーションまたはトリガーによって呼び出されるプロシージャやファンクションなど、データベースに存在するコードが含まれます。スキーマオブジェクトと同様に、データベースコードオブジェクトも DDL を使用して作成されます。さらに、コードオブジェクトはスキーマオブジェクトと密接に関連付けられ、それらを参照するか、それらによって参照されます。したがって、データベースコードオブジェクトは、通常、他のスキーマオブジェクトと同時に移行されます。

他のスキーマオブジェクトと同様に、このステップの主なアクティビティは、互換性がないまたは自動的に変換できないデータベースコードオブジェクトの回避策を特定して実装することです。Oracle のプロシージャ言語 PL/SQL は他のデータベースネイティブビルトイン言語とは異なるため、これは多くの場合、データベースの移行で最も困難なタスクになる場合があります、特にソースデータベースに大量のコードオブジェクトが含まれている場合、またはコードオブジェクトが構成されている場合多くの行の複雑なコード。ここで、最も一般的に使用される PL/SQL 機能の多くをネイティブに提供する EDB Postgres Advanced Server の PL/SQL 実装が重要になります。これらの機能は、この手順を完了するために必要な労力を大幅に削減できます。移行ポータルは、他のスキーマ DDL に加えて Oracle データベースコードオブジェクトを評価し、互換性のないオブジェクトを更新および再評価する機能を備えています。

データベース内のデータを移行する

データを移行する場合、2つの基本的なオプションがあります。1つ目は、スナップショットプロセス特定の時点でのすべてのデータのコピーを使用して、目的のすべてのデータを移動することです。このオプションは、データベースが比較的小さい場合、または運用移行の場合、移行を完了するために必要なアプリケーションのダウンタイムが許容できる場合に選択されることがよくあります。

スナップショットレプリケーションを定期的に行って、ターゲットデータベースの内容をソースデータベースの現在の内容でリフレッシュできます。ただし、このような定期的なリフレッシュでは、通常、以前にコピーしたすべてのデータをターゲットシステムから削除し、すべてのデータを再度移行する必要があります。2番目のオプションは、変更データキャプチャ CDC 技術を使用して、継続的かつ継続的にデータを移行することです。すべての変更データキャプチャプロセスの一部として、初期ロードはある種のスナップショットによって実行されます。ただし、初期スナップショットの後、変更データキャプチャメカニズムを使用して、ソースデータベースで行われたその後の変更を識別し、変更をターゲットデータベースに転送します。CDC ベースのデータ移行オプションは、通常、最小限のダウンタイムが必要で、ソースシステムで行われるデータの変更とターゲットシステムの同期を保つことが重要な場合に選択されます。

EDB Migration Toolkit を使用してスナップショットタイプのデータ移行を実行し、EDB Replication Server を使用して Oracle から Postgres への CDC ベースのデータ移行を実行できます。スナップショットベースのデータ移行と CDC ベースのデータ移行は2つの主なアプローチですが、他のハイブリッドまたはカスタムアプローチを使用して、2つの主要な方法では対応できない特別なデータ移行のニーズへのソリューションを提供できます。たとえば、データベースリンクを介したクエリまたはターゲットデータベースからソースデータベースへの直接接続を介してデータを取得することが役立つ場合があります。

1.3.5 5. インターフェイスとアプリケーションを移行する

新しく移行した Postgres データベースで動作するようにアプリケーションを変換します。アプリケーションは、データベースと対話するデータベースの外部のコードです。

アプリケーションの移行には以下が含まれます。

- アプリケーション接続文字列とドライバー JDBC、ODBC、OCI、.NET などの更新。JDBC や ODBC などのオープン標準の接続を使用するアプリケーションは、通常、データベース接続文字列と EDB ドライバーの選択の変更のみが必要です。ドライバーのインストールと、EDB Postgres Advanced Server データベースに接続するようにアプリケーションの構成の詳細については、次のドライバー固有のドキュメントを確認してください。
- [EDB JDBC Connector](#)
- [EDB ODBC Connector](#)
- [EDB .NET Connector](#)
- [EDB OCI Connector](#) Oracle OCI ドライバーの EDB バージョン

.NET アプリケーションの場合、Oracle .NET ドライバーを使用するアプリケーションを更新して、Oracle ドライバー固有のクラス名の呼び出しを EDB ドライバーの同等の名前に置き換える必要があります。

- アプリケーション内の埋め込み SQL の変換。SQL は、ソースデータベースで動作するように構築されている可能性があります。Oracle がサポートしている特定の構文が Postgres でサポートされてい

ない場合がある場合があります。この場合、Postgres で動作するように SQL を変換する必要があります。EDB Postgres Advanced Server および EDB ドライバーに組み込まれている Oracle 機能と EDB の互換性により、必要な変換の量が削減されます。

- 必要に応じてアプリケーションコードを更新します。場合によっては、アプリケーションには、Postgres ドライバーでは使用できない Oracle ドライバー固有の呼び出しを含むコードが含まれる場合があります。このようなコードを更新して、Postgres ベースの回避策を実装する必要があります。
- アプリケーションの移行。移行プロジェクトによっては、実際のアプリケーションのホストされる場所とその構成の移動が発生する場合があります。ソースコードの移行とともに発生する必要があります。通常、これにより互換性の問題は発生しません。ただし、新しい環境でアプリケーションのデータベース接続、機能、およびパフォーマンスを確認することが重要です。さらに、Oracle アプリケーションを Postgres に移行する場合、接続プーリングとロードバランシングのニーズに対処するためにさまざまなテクニック、ツール、および構成が必要になる場合があります。

1.3.6 6. レポートと管理ツールを移行する

このフェーズでは、レポート、DBA ユーティリティ、およびスクリプトを移行します。組織は多くの場合、コアアプリケーションをサポートするためにデータベースを照会および対話するレポートメカニズムやその他の管理ツールを持っています。これらのアーティファクトの一部は、Oracle ベースのプロセスをサポートするように特別に構築されているため、それらのすべてが移行後のシステムに適用されない場合、または必要とされるわけではありません。したがって、最初にレポート、ユーティリティ、スクリプト、および同様のアーティファクトのインベントリを確認して、残し、置き換え、および更新するものを決定することをお勧めします。

スクリプトまたはユーティリティの更新には、多くの場合、Oracle 用に構築された互換性のない SQL をターゲットデータベースと互換性のあるものに更新することが含まれます。また、Oracle コマンドラインインターフェイスではなく、ターゲットデータベースが提供するコマンドラインインターフェイスで動作するようにそれらを更新することも含まれる場合があります。たとえば、SQL *Plus は Oracle のプライマリコマンドラインインターフェイスであり、psql は Postgres で提供されているものです。SQL ステートメントは両方のインターフェイスで発行できますが、それらが提供する他のサポートコマンドとキーワードは大きく異なります。多くの Oracle スクリプトとユーティリティは、SQL*Plus コマンドを使用してビルドされています。これらのスクリプトを psql で実行するには、代わりに psql コマンドを使用するように変換する必要があります。

幸いなことに、EDB は EDB *Plus を提供しています。これは、最も一般的に使用される SQL *Plus コマンドの一部を理解する SQL *Plus 作品同様のユーティリティです。この機能は、EDB Postgres Advanced Server の Oracle 機能の他のビルトイン互換性と組み合わせて、Oracle 用にビルドされた多くの既存のスクリプトを EDB Postgres Advanced Server で実行できます。EDB Postgres Advanced Server ユーザーは、EDB Postgres Advanced Server に組み込まれている Oracle の互換性機能と動作するように拡張された psql のバージョンを使用することもできます。psql に慣れた後、ユーザーは多くの場合、EDB*Plus の代わりにそれを使用して、その機能を利用することにします。したがって、SQL*Plus ベースのスクリプトとユーティリティを psql ベースのものに変換することを選択する場合があります。

Oracle が提供するもう 1 つの人気のあるインタフェースは SQL *Loader で、これは Oracle にデータを大量にロードするために使用されるツールです。EDB は、EDB Postgres Advanced Server にデータをロードするために使用できる EDB*Loader と呼ばれる同等のユーティリティを提供します。EDB *Plus と同様に、EDB*Loader は、EDB Postgres Advanced Server で使用できる使い慣れたインタフェースを Oracle

ユーザーに提供します。このツールを使用すると、既存のデータのロード手順に必要な変更が削減されます。EDB Postgres Advanced Server ユーザーは、データをロードするための Postgres ネイティブ方法も利用できます。これらの方法を選択した場合、Oracle 固有の機能に基づいてデータのロードプロセスを更新する必要があります。

他のレポートまたは管理ツールを使用して、Oracle データベースから情報を取得する場合があります。これらのツールによって発行された SQL クエリーを評価して、ターゲットデータベースと互換性があるかどうかを判断します。クエリが新しいシステムに適用されると仮定すると、SQL の非互換性またはその他の操作の違いを特定した場合は、クエリを更新する必要があります。

1.3.7 7. 移行のテスト

移行のサイズに関係なく、移行されたアプリケーションが予想通りに動作することを確認する必要があります。ずっとテストを続けますが、ある時点で、次のことを行うための正式なテストを実行する必要があります。

- データがターゲットデータベースに完全に移行され、ソースデータベースと一貫していることを検証します。 - データベースとアプリケーションの機能を確認します。 - データベースとアプリケーションのパフォーマンスが許容可能なものであることを検証します。

この手順では、データベースとアプリケーションの機能とパフォーマンスを適切にテストできるテスト環境をセットアップすることが重要です。予想される運用ワークロードをシミュレートしたワークロード、および運用環境と同様の動作環境で、実際のアプリケーションまたはアプリケーションのコピーを使用して、これらのアプリケーションとデータベースのテストを実行します。機能テストは、アプリケーションコードを実行し、通常のワークロードで使用される頻度が低いデータベースオブジェクトと対話するための十分なカバレッジを提供する必要があります。

データ検証に関しては、ソースデータベースとターゲットデータベースのデータの比較に使用されるツールとプロセスは、違いを簡単に識別できるようにする必要があります。違いが存在する場合は、分析を実行して、違いが発生した理由と、移行プロセスの変更が必要かどうかを理解する必要があります。ターゲットシステムのデータがソースシステムのデータで更新される頻度によっては、データ検証チェックを複数回実行する必要がある場合があります。進行中の CDC ベースのデータ移行の場合、データを定期的に検証するための計画を確立します。

この手順では、可用性要件を満たし、バックアップとリカバリ操作を実行し、データベースを監視および管理するために使用するツールとプロセスをセットアップおよび実行することをお勧めします。これは、運用環境での使用の準備として、ツールと関連システムの操作手順または構成に必要な変更を特定するのに役立ちます。

このステップの主な目的の 1 つは、実稼働システムへの最終的な移行を続行する前に、対処すべき問題を特定することです。問題を見つけて修正する場合は、さらにテストを実行する必要があります。実稼働前移行テストの結果に満足したら、実稼働環境の最終準備を実行できます。

1.3.8 8. 移行後のシステムの最適化と構成

このフェーズでは、必要なソフトウェアがすべてインストールされていることを確認し、テストフェーズ中に特定および検証された変更で実稼働環境を更新します。また、以前のパフォーマンステスト結果から導き出された推奨事項に基づいて、システムの構成と調整も行います。データベースパラメーターとアプリケーション設定を調整して、システムが最適に実行されるように準備する必要がある場合があります。また、テストフェーズで特定および検証したクエリチューニング関連の更新も適用します。

このフェーズは、高可用性、ディザスタリカバリー、およびセキュリティ要件への対応を終了する時期でもあります。ユーザーとアプリケーションがデータベースに対して認証する方法、各アクションを実行する権限を誰に与えるかなど。必要に応じて、これらの要件が満たされていることを確認し、データベースを監視および維持するための標準操作手順 SOP を更新します。

1.3.9 9. 完全なカットオーバー/ゴースト

このフェーズは次で構成されます。

- 運用データベースへのデータの移行の完了。CDC ベースのアプローチを使用してソースデータベースからデータを移行する場合、ソースデータベースへのさらなる変更を無効にして、新しいデータベースが古いデータベースと同期するようにします。通常、必要に応じて、ソースデータベースの読み取り専用クエリを引き続き許可できます。
- ロールバックオプションの設定。古いデータベースが新しいデータベースから更新を受信し、新しいデータベースでアプリケーションの実行に問題が発生した場合にフォールバック位置として存在するように、ロールバック構成を設定します。この方法により、元のソースデータベースに対するアプリケーションの実行に戻ることができます。データをターゲットデータベースに戻すことは、多くの場合、変更データキャプチャメカニズムを介して実行されます。
- 新しいデータベースで移行したアプリケーションの使用を開始するようにシステムの構成。これは通常、新しいデータベースが古いデータベースからすべての最終更新を受信したことを確認した後にのみ実行されます。
- 続行/停止の決定が下されるまで、新しいシステムをテストします。新しいアプリケーションとデータベースを一定期間テストします。アプリケーションが新しい環境で適切にパフォーマンスしていることを確認したら、アプリケーションを完全な運用環境での使用に「go」と宣言できます。
- 運用へのカットオーバーを終了します。カットオーバーが完了し、それに満足したら、通常、データの変更をターゲットデータソースデータベースに送信するフォールバック位置を削除します。その時点で、新しい実稼働システムでのみ実行されます。

1.4 EDB capabilities for the migration journey

Oracle から EDB Postgres Advanced Server への移行を実行する利点の 1 つは、EDB がこれらの移行の実行において持っている経験です。フェーズごとに、移行を簡単にするツールとサービスを提供します。また、お客様のビジネス要件について話し合い、移行の決定とお客様に最適なアプローチをお手伝いします。

1.4.1 移行作業で使用する機能

移行を支援するために EDB が提供するツールとサービスを特定する前に、移行作業の各ステップに必要な機能を確認する価値があります。この議論の文脈では、機能という用語はソフトウェアツールまたは従業員のスキルと経験を指す場合があります。移行行程のすべての手順は、経験豊富な個人が実行することによりメリットがありますが、それらの多くは、適切なツールセットを使用するとより簡単に実行できます。

これらは、移行作業の各ステップを正常に完了するために必要な重要な種類のツールとスキルの一部です。

実現可能性と代替案の分析

- ソースデータベースをターゲットデータベースに移動する可能性を評価するツールと機能
- 移行の複雑さを分類するツールと機能

移行を計画する

- 移行の労力を見積もる機能
- 移行の順序を優先する機能
- 非機能要件を定義し、高レベルのソリューションを設計する能力

データベーススキーマ、コード、データを移行する

- データベースの互換性
- 互換性のないオブジェクトを変換するツールと機能
- スナップショットまたは CDC ベースのデータの移行を実行するツールと機能

インターフェイスとアプリケーションを移行する

- データベースドライバ/コネクタの互換性
- アプリケーション SQL を解析および評価するツール
- アプリケーションコードをリファクタリングし変換するツールまたは機能

レポートと管理ツールを移行する

- レポートとユーティリティで SQL を解析および評価するツール
- 互換性のない SQL を変換する機能

移行をテストする

- データの一貫性の検証に役立つツール
- 機能およびパフォーマンスの検証とこれらのアクティビティを実行する機能を支援するツール

移行後のシステムの最適化と構成

- データベース、クエリ、およびアプリケーションのチューニングを支援するツールと機能
- 高可用性、データ復旧、セキュリティ、認証/承認要件に対応して構成するツールと機能

完全なカットオーバー/ゴーライブ

- ロールバックをサポートするためのリバースデータ移行を実行するツールと機能
- データの一貫性の検証に役立つツール
- 移行したシステムを操作および保守するツールと機能

1.4.2 EDB 移行ソフトウェアとサービス

表は、EDB が提供する移行関連機能と、サポートする移行過程のステップとのマッピングを示しています。各機能の概要は、表の後に提供されます。詳細は、EDB ウェブサイトの製品ドキュメントとパッケージサービスの説明を参照してください。

表を確認するときは、以下が適用されます。 **太字**=EDB ツール **斜体**=EDB プロフェッショナルサービスパッケージ [括弧付き=Oracle との互換性]

Journey Step	EDB Capability
Analyze feasibility and alternatives	• Migration Portal
 - <i>Migration Assessment Service</i>
Plan the migration	• <i>Migration Assessment Service</i>
 - <i>Enterprise Architecture Service</i>
 - <i>Solution Design Service</i>
Migrate database schema, code, and data	• [EDB Postgres Advanced Server]
 - Migration Portal
 - Migration Toolkit
 - Replication Server
 - Database Links and dblink_oracle Functions
 - <i>Quick Deploy Service</i>
 - <i>Embedded Postgres SME Service</i>
Migrate interfaces and application	• [EDB Postgres Advanced Server]
- [EDB Database Connectors]
 - <i>Embedded Postgres SME Service</i>
Migrate reports and management tools	• [EDB*Plus]
 - [EDB*Loader]
 - <i>Embedded Postgres SME Service</i>
Test the migration	• LiveCompare
 - Postgres Enterprise Manager (PEM) [¹]
 - <i>Embedded Postgres SME Service</i>
Optimize and configure the post migration system	• Postgres Enterprise Manager (PEM) [¹]
 - Barman
 - Failover Manager [²]
 - EDB Postgres Distributed [³]
 - <i>Performance Tuning Service</i>
 - <i>Monitoring Best Practices Service</i>
 - <i>Backup Best Practices Service</i>
 - <i>Embedded Postgres SME Service</i>
Complete cutover/go live	• Replication Server
 - LiveCompare
 - <i>Embedded Postgres SME Service</i>

[¹]: PEM は BigAnimal クラスターをモニタできますが、いくつかの制限があります。クラウドサービスプロバイダーが提供する監視サービスは、PEM と組み合わせて、またはその代わりに使用されます。PEM は、クラウドネイティブ Postgres インスタンスのモニタリングには適用されません。[²]: フェールオーバーマネージャーは、提供する機能が BigAnimal サービスまたは CNP オペレーターにネイティブに組み込まれているため、BigAnimal またはクラウドネイティブ PostgreSQLCNP 展開には適用されません。[³]: EDB Postgres Distributed は、現在 BigAnimal および Cloud Native PostgreSQL オフファリングでは利用できません。利用可能になると、その機能はオプションとして BigAnimal サービスとクラウドネイティブ Postgres オペレーターに直接統合されます。

EDB Postgres Advanced Server

EDB Postgres Advanced Server は、PostgreSQL の強化バージョンです。これには、コア PostgreSQL のすべての機能が含まれているだけでなく、Oracle とのビルトイン互換性および強化されたセキュリティ機能を含むその他のエンタープライズグレードの機能も含まれています。EDB Postgres Advanced Server は、次の領域で Oracle との互換性を提供します。

- Oracle 固有および構文互換のデータベースオブジェクトタイプ
- Oracle 固有のデータ型
- Oracle 固有の SQL 拡張機能
- ビルトインネイティブプロシージャル言語としての Oracle PL/SQL のサポート
- Oracle データディクショナリビューつまり ALL、DBA、USER_ビュー
- Oracle ビルトイン PL/SQL パッケージ

これらの機能により、Oracle からの移行が簡素化されます。EDB Postgres Advanced Server の互換性機能は、Oracle で動作したものは EDB Postgres Advanced Server で動作するため、変換するものが少なくなことを意味します。EDB Postgres Advanced Server は、すべての Oracle 機能を完全に実装しているわけではありませんが、実装される Oracle 機能との互換性は、最も一般的に使用される構成体の一部です。さらに、EDB は、各メジャーリリースと互換性機能を追加し続けています。Oracle と EDB Postgres Advanced Server 間の互換性については、[Comparison of EDB Postgres Advanced Server with Oracle](#) を参照してください。

EDB Postgres Advanced Server は、さまざまな Linux プラットフォームと Windows バージョンにインストールできます。サポートされているプラットフォームの完全なリストについては、EDB ウェブサイトの [Platform Compatibility](#) を参照してください。さらに、EDB Postgres Advanced Server を BigAnimal サービスとして、および the EDB Postgres for Kubernetes operator を使用して Kubernetes 環境に展開できます。

Migration Portal

[Migration Portal](#) は、EDB が提供する無料のオンラインサービスで、Oracle データベーススキーマと EDB Postgres Advanced Server の互換性を評価します。移行ポータルプロジェクトを作成し、Oracle データベースの SQL 形式のデータ定義言語 DDL ファイルをアップロードすると、移行ポータルは DDL オブジェクトの互換性を分析し、修復ハンドラーを適用して既知の非互換性を修正し、レポートします結果。次に、移行ポータルで結果の確認、すべての DDL オブジェクトの検査、エラーでオブジェクトを更新して問題を修正、および移行ポータルでプロジェクトを再評価できます。移行プロジェクトの DDL が完全に変換されると、100 の互換性が作成されると、移行ポータルは、そのスキーマを EDB Postgres Advanced Server インスタンスにロードするのを支援できます。

Migration Toolkit

Migration Toolkit は、いくつかの異なるデータベース移行のユースケースをサポートする Java ベースのコマンドラインツール。ただし、Oracle から EDB Postgres Advanced Server の移行のコンテキストでは、その主な使用はスナップショットタイプのデータ移行を実行することです。ソース情報のどのセットをターゲットデータベースにロードする方法と制御するための豊富なコマンドオプションを提供します。1 回限りの試用期間は利用できますが、長期使用には EDB データベースのサブスクリプションが必要です。

Replication Server

Replication Server は、グラフィカルユーザーインターフェイスとコマンドラインインターフェイスの両方を提供する Java ベースのアプリケーションです。Oracle を含む非 PostgreSQL データベースからシングルマスターモードで Postgres にデータをレプリケートするための堅牢なデータプラットフォームを提供します。Migration Toolkit の代替として、Oracle からデータを移行するための変更データキャプチャ CDC ベースのオプションを提供します。これにより、アプリケーションのダウンタイムが懸念される場合に特に役立ちます。Replication Server は、Postgres から Oracle へのレプリケーションもサポートしています。その結果、それを使用して、移行の最終段階で問題が発生した場合のフォールバックオプションを確立できます。

1 回限りの試用期間は利用できますが、長期使用には EDB データベースのサブスクリプションが必要です。

LiveCompare

LiveCompare は、Oracle および Postgres データベースを含む任意の数のデータベースを比較し、同一であることを確認するために使用できるコマンドラインユーティリティ。データベースを比較した後、ツールは比較レポート、相違点のリスト、およびデータ操作言語 DML SQL スクリプトを生成します。DML を適用し、データベースの不一致を修正できます。

LiveCompare は、いくつかの異なるユースケースをサポートしていますが、移行のコンテキストでは、それを使用して、あるデータベースから別のデータベースに移行されたデータを検証できます。Oracle から新しい Postgres データベースにデータを移行した後に実行すると、その比較レポートは、データが完全に移行されたかどうか、または対処すべきデータの不一貫性があるかどうかを知るのに役立ちます。フォールバック位置にデータを Oracle に移行する場合、LiveCompare は 2 つのデータベースが同じデータを持つことを確認できます。

LiveCompare は、1 回限りの試用期間の一部として使用できますが、長期使用するには EDB データベースサブスクリプションが必要です。

Oracle 互換データベースコネクタ

EDB は、Oracle 機能との互換性を含むように強化されたバージョンの PostgreSQL データベースドライバコネクタのバージョンを提供し、もともと EDB Postgres Advanced Server で実行するように Oracle 用に構築された実行アプリケーションのサポートを強化します。次の Oracle 互換の Postgres ベースのコネクタが使用できます。

- EDB JDBC Connector

- EDB ODBC Connector
- EDB .NET Connector

EDB は、[EDB OCL Connector](#) を提供して、Oracle Call InterfaceOCI ベースのアプリケーションを EDB Postgres Advanced Server に対して実行できるようにします。

この表では、EDB 拡張コネクタで提供される重要な互換性機能のいくつかを強調しています。

Oracle compatibility feature	JDBC	ODBC	.NET	OCL
PL/SQL Support	OK	OK	OK	OK
REF_CURSOR - OracleTypes.CURSOR	OK	OK	OK	OK
User-defined Exceptions - vendor code	OK	OK		OK
Named Parameters - parameter names	OK	OK	OK	OK
Data Types - VARCHAR2 , STRUCT, ARRAYS	OK	OK	OK	OK
STRUCT - Enhanced Manipulation	OK		OK	OK
Upper Column Names - (OPTIONAL)	OK			
Multiple INOUT/OUT parameters	OK	OK	OK	OK

EDB 拡張データベースコネクタは、EDB Postgres Advanced Server で使用できますが、EDB データベースサブスクリプションが必要です。

その他の EDB ソフトウェア

EDB*Plus

[EDBPlus](#) は、EDB Postgres Advanced Server への Java ベースのコマンドラインインターフェイスです。EDB *Plus は、SQL コマンド、SQL 匿名ブロック、および EDB* Plus コマンドを受け入れます。EDB * Plus コマンドは Oracle SQL* Plus コマンドと互換性があるため、Oracle 用に既に構築したスクリプトを EDB Postgres Advanced Server データベースに対して使用できます。

EDB*Loader

[EDBLoader](#) は、EDB Postgres Advanced Server の Oracle データベースと互換性のあるインタフェースを提供する高性能大量データのローダーです。EDB *Loader ユーティリティは、Oracle SQL* Loader が提供するパラメータのサブセットを使用して、入力ソース通常はファイルから 1 つ以上のテーブルにデータをロードします。

データベースリンクと `dblink_ora` ファンクション

`Database links` および `dblink_ora` ファンクションは、SQL クエリーとデータベースファンクションコールを介して Oracle から Postgres にデータをプルできる Postgres 拡張対応機能です。これらのテクノロジーを使用して、Oracle データベースからのデータの移行を支援することもできます。これらの方法は両方とも、Oracle Call Interface OCI を使用して Oracle に接続します。接続した後、SQL ステートメントを使用して「リンクされた」Oracle データベースからデータを選択し、EDB Postgres Advanced Server データベースにデータを挿入します。

これら 2 つのオプションのいずれかを使用することを決定する場合、通常はデータベースリンクオプションが使いやすいため、優先されます。ただし、`dblink_ora` ファンクションは、BLOB および CLOB データなど、データベースリンクオプションで移行されない一部のデータ型を移行するのに役立ちます。また、Oracle データベースで実行される問合せでの Oracle 固有のファンクションの使用をサポートします。これにより、CLOB または BLOB 列に埋め込まれた JSON データなど、特定の種類の情報を抽出する際にある程度の柔軟性を提供できます。これらの機能は両方とも EDB Postgres Advanced Server で提供されていますが、使用するには設定が必要です。

EDB Postgres Enterprise Manager PEM

EDB Postgres Enterprise Manager は、オープンソースの pgAdmin 4 プロジェクトに基づいた包括的なデータベースモニタリングおよび管理アプリケーションです。これも EDB でサポートされています。PostgreSQL および EDB Postgres Advanced Server データベースと、それらが実行されているシステムを検査、監視、管理、および照会するためのグラフィカルユーザーインターフェイスを提供します。データベースの状態とパフォーマンスを監視するために使用できるレポート、ダッシュボード、プローブ、およびアラートを提供します。PEM には、多くのデータベース管理およびデータベース開発機能も含まれています。PEM の機能は、データベースの移行の一部として使用され、パフォーマンスを監視および調整し、必要に応じてデータベースオブジェクト定義を更新します。PEM を使用して、EDB データベースサブスクリプションの下にある Postgres データベースを監視および管理できます。

フェールオーバーマネージャー

Failover Manager は、Postgres データベースクラスターを管理するための高可用性ソリューションであり、ストリーミングレプリケーションを使用したプライマリスタンバイ展開アーキテクチャの高可用性を有効にします。フェールオーバーマネージャーは、ソフトウェアまたはハードウェアに障害が発生した場合に、Postgres プライマリデータベースノードをスタンバイデータベースノードに自動フェールオーバーを提供します。データベースの移行のコンテキストでは、移行の実行またはテストには使用されません。これは、移行後のシステムで使用して、特定の高可用性要件に対応できる運用ツールです。フェールオーバーマネージャーを使用して、EDB データベースサブスクリプションの下にあるプライマリデータベースとスタンバイデータベースを管理できます。

EDB Postgres Distributed

EDB Postgres Distributed は、高度な競合管理、データ損失保護、およびネイティブ論理レプリケーションより最大 5 倍高速のスループットを備えたマルチマスターレプリケーションとデータ Distributed を提供する PostgreSQL 拡張機能です。最大ファイブナインの高可用性を備えた DistributedPostgreSQL クラスタが有効になります。より極端な高可用性シナリオの場合、EDB Postgres Distributed 対 EDB Failover Manager をお勧めします。これは、Oracle から EDB Postgres Advanced Server の移行を実行するために使用するツールではありませんが、移行後のシステムで使用することを目的としている場合、ターゲットシステムに展開し、テストして識別する必要がありますその使用をサポートするために必要なアプリケーションまたはデータベースの変更。EDB Postgres Distributed を使用するには、EDB サブスクリプションが必要です。

Barman バックアップおよびリカバリーマネージャー

Barman バックアップおよびリカバリーマネージャー は、Postgres サーバーのリモートバックアップとディザスターリカバリーのために EDB によってサポートおよび保守されているオープンソース管理ツールです。移行後のオペレーティング環境には、適切なバックアップおよびリカバリーの手順とツールを使用することが重要です。さらに、これらのツールは、多くの場合、移行テストおよび関連する開発作業をサポートするプロセスに統合されています。Barman に加えて、EDB は、サブスクリプションサービスの一部として、人気の Postgres バックアップおよびリカバリーツールである pgBackRest もサポートしています。

EDB プロフェッショナルサービス

EDB は、データベースの移行のサポートでよく使用される次のプロフェッショナルサービスパッケージを提供しています。

- **Migration Assessment Service** データベースの詳細な分析を実行します。その分析から、移行がどれほど難しいか、およびデータベースの移行の実行にどれくらいの時間と労力がかかるかがわかります。
- **Enterprise Architecture Service** は、移行したシステムの操作、インフラストラクチャ、および環境要件を定義し、可用性とスケーラビリティのニーズをサポートするアーキテクチャを推奨します。
- **Solution Design Service** は、データベース戦略、サービスレベル契約、インフラストラクチャのニーズを満たす Postgres データベースソリューションを推奨します。
- **Quick Deploy Service** は、標準のリファレンスアーキテクチャとベストプラクティスに基づいて、可用性、スケーラブル、そして安全な方法で Postgres を展開する方法を理解するのに役立ちます。
- **パフォーマンスチューニングサービス** アプリケーションのニーズを満たすようにデータベースを調整するのに役立ちます。
- **Monitoring Best Practices Service** は、移行後のデータベースの操作と保守のための適切なモニタリングプラクティスを確立するのに役立ちます。
- **Backup Best Practices Service** は、データを決して失わないよう、適切なバックアップとリカバリーの実践を確立するのに役立ちます。
- Postgres の専門知識でスタッフを強化し、移行と移行後の操作のさまざまな段階をガイドします。
- スタッフが移行後の Postgres データベースの操作とメンテナンスに習熟するのを助ける **トレーニング**

これらおよび他の EDB パッケージサービスの詳細については、[Consulting Services](#) ページを参照してください。

1.5 Comparison of EDB Postgres Advanced Server with Oracle

Oracle から EDB Postgres Advanced Server への移行の実現可能性を評価するときは、2つのデータベーステクノロジーの類似点と相違点を理解する必要があります。EDB Postgres Advanced Server は、多数のビルトイン Oracle 互換性機能を含む PostgreSQL の強化バージョンです。これらの互換性機能により、コア PostgreSQL よりも Oracle に似ていますが、違いはまだ残っています。以下の比較の目的は、Oracle が提供する機能と機能について、自動的にまたは最小限の労力で移行するものと、より多くの労力を必要とする機能の雰囲気を感じていただくことです。

この比較では、次のトピックについて説明します。

- *Comparison of general characteristics*
- *Comparison of database features*
- *Comparison of database operations-related capabilities*
- *Comparison of application development capabilities*
- *Comparison of nonrelational data support*
- *Notable differences*

このことを理解することは、EDB Postgres Advanced Server が Oracle データベースを置き換える良いオプションかどうかを判断するのに役立ちます。この情報は、データベースを EDB Postgres Advanced Server に移行するために必要な作業を計画するのに役立ちます。EDB Postgres Advanced Server に移行することを決定した後、一般的な移行問題の詳細な回避策については、[Migration Portal](#) **ポータル Wiki** タブで利用可能なナレッジベースを参照してください。

1.5.1 Comparison of general characteristics

検討中のユーザーは、Oracle のデータベースを EDB Postgres Advanced Server データベースと比較するとき、いくつかの基本的な詳細を理解する必要があります。これらの基本的な特性を理解することは、Oracle の専門家と Postgres の専門家間のディスカッション中に特に価値があります。

用語

互換性の問題を詳細に説明する前に、多くの SQL ベースの製品で使用される命名法に違いがあることに注意してください。表は、これらの違いの一部を示しています。

Oracle	EDB Postgres Advanced Server
Table or index	Table, index, or relation

[次のページに続く](#)

表 3 – 前のページからの続き

Oracle	EDB Postgres Advanced Server
Row	Row or tuple
Column	Column or attribute
Data block	Page—When block is on disk
Page	Buffer—When block is in memory

さらに、EDB Postgres Advanced Server の各インスタンスはクラスターと呼ばれます。クラスターは、単一のプログラムインスタンスによって管理されるデータベースのコレクションです。これは、すべてのデータと構成ファイルを含むデータディレクトリで構成されています。これは2つの方法で参照できます。データディレクトリの場所によるか、ポート番号による。単一のサーバーに多くのプログラムをインストールでき、複数のクラスターを作成できます。

一般的な機能

Oracle と EDB Postgres Advanced Server は、両方とも成熟したエンタープライズクラスのオブジェクトリレーショナルデータベースであり、アトミック、一貫性、分離、および耐久性 ACID コンプライアンスの業界標準を満たしています。どちらも System R に関する同じ IBM の研究から開発され、多くの同じ問題を解決するように設計されました。これらのデータベースプログラムには多くの類似点があります。

General capabilities	Oracle	EDB Postgres Advanced Server (EPAS)
Design origin	Commercial implementation based on IBM's original research for System R	Academic implementation (UC Berkeley) based on IBM's original research for System R
Continuous development	Since 1979	PostgreSQL development started in 1986. EPAS development started in 2004. EPAS is based on PostgreSQL and continuously merged.
Object relational database	Yes	Yes
Processing architecture	Process based and thread based	Process based
Full ACID compliance	Yes	Yes
Multiversion concurrency control	Yes	Yes
Multitenant architecture	Yes	Yes
Automatic workload management	Yes	No
Enterprise database management	Oracle Enterprise Manager	EDB Postgres Enterprise Manager
Multicore support	Yes	Yes
Write-ahead durability	Redo logs	Write-ahead log
Disk-read buffering	Yes	Yes

キャパシティ

新しいデータベースを検討するときは、新しいソリューションが既存のアプリケーションデータデザイン、ワークロード、および予想される成長をサポートする能力があるかどうかを理解する必要があります。新しいソリューションの容量をワークロードと将来のアプリケーションに適用することは、データベース内の複数の構造にわたってデータをサポートする方法を理解することを意味します。

Capacities	Oracle	EDB Postgres Advanced Server
Max. database size	Unlimited	Unlimited
Max. table size	4 GB x Block Size	32 TB
Max. row size	4 TB	1.6 TB
Max. field size	For BLOB: (4 GB - 1) x DB_BLOCK_SIZE initialization parameter	1 GB
Max. rows per table	Unlimited	Unlimited
Max. columns per table	1000	250 - 1600 depending on column types
Max. indexes per table	Unlimited	Unlimited

1.5.2 Comparison of database features

移行の実現可能性を評価するには、次の表を参照して、移行する要素が Oracle から EDB Postgres Advanced Server への移行でどのように処理されるかを理解します。

テーブルの凡例

表では、次の単語または記号を使用して互換性を示します。

はい/いいえ—機能または特性がデータベースでサポートされているかどうかを示します。

はい—この機能は Oracle と互換性のある方法で動作し、既存の Oracle スキル、プログラムコード、またはデータを継続使用または移行できます。

テーブルとパーティショニング

データベース内の構成の範囲と、これらの構造を編成する際の DBA の柔軟性は、パフォーマンス、メンテナンス、およびその他の操作要件に影響を与える可能性があります。データベースをパーティション化する機能により、たとえば、パフォーマンスが向上します。データを個別の構造に編成し、インフラストラクチャ全体に Distributed することも、管理性、可用性、ロードバランシングを向上させます。マテリアライズドビューを使用すると、DBA は、低速でリソースを大量に使用するランタイムクエリー、複雑な結合、または長時間のデータのスキャンを、事前結合、事前ソート、および保存された結果からのシンプルで高速な読み取りに置き換えることができます。

Oracle Enterprise	EDB Postgres Advanced Server	Ad- Notes
Temporary tables	Partial	グローバル一時テーブルは Postgres ではサポートされていません。
Views	Yes	NA
Materialized views	Yes	Postgres は自動更新機能を提供していません。トリガーは、Postgres ソリューションの一部として使用できます。 - Postgres はインクリメンタル更新オプションを提供せず、完全な更新のみを提供します。
Partitioning	Yes	NA
Partition by range	Yes	NA
Partition by hash	Yes	NA
Partition by list	Yes	NA
Subpartitioning	Yes	NA
Interval partitioning	Yes	NA
Partitioned indexes	No	NA
ANSI constraints	Yes	NA
Tablespaces	Yes	Oracle および Postgres では、テーブルスペースは、テーブル、インデックス、およびその他のデータベースファイルを保存する場所を編成および指定するために使用されます。ただし、Oracle とは異なり、Postgres では、テーブルスペースは、自動的に生成されたテーブル、インデックス、およびその他のデータベースファイルが保存されるディレクトリです。
Index organized tables	No	Postgres では、インデックスでテーブルをクラスター化でき、事前に並べ替えられた構造からデータを読み取る時にも同様のパフォーマンスを向上させます。
Sequences	Yes	Postgres では、一部のキーワードオプションは使用できません。 - 最大シーケンス値の若干の違い。

次のページに続く

表 6 – 前のページからの続き

Oracle Enterprise	EDB Postgres Advanced Server	Ad- Notes
Global indexes	No	パーティションテーブルのグローバルインデックスは、Postgres では使用できません。パーティションキー列を含まないパーティションテーブルで主キーを定義した場合、インデックスは継承されません。ただし、CREATE TABLE で定義されたすべてのパーティションには、列に独立したプライマリインデックスがあります。ALTER TABLE ... ADD CONSTRAINT を使用して、新しく追加されたすべてのパーティションの主キーを再作成できます。このプライマリインデックスは、各パーティションに一意性を適用しますが、パーティション階層全体には適用されません。つまり、2つ以上のパーティションのプライマリインデックス列に同じ値を繰り返すことができます。

データ型

データタイプは、DBMS がシステム内の情報を定義、実装、および使用する方法を提供し、複数のデータ型が使用されている場合にデータベースによってデータが解釈される方法に制約を課します。EDB Postgres Advanced Server は、Oracle データ型との強い互換性があり、拡張性が高いため、人気が高まるにつれて、新しいデータタイプとワークロードを迅速にサポートできます。

Data types	Oracle Enterprise	EDB Postgres Advanced Server
Type system	Static + dynamic (through ANYDATA)	Static
Integer	NUMBER	NUMBER OK , DEC , NUMERIC, SMALLINT (16-bit), INT, BINARY_INTEGER, PLS_INTEGER, INTEGER (32-bit), BIGINT (64 bit)
Floating point	BINARY_FLOAT, BINARY_DOUBLE	BI-FLOAT, REAL (32-bit), DOUBLE PRECISION (64-bit)
Decimal	NUMBER	NUMBER OK , DEC, DECIMAL, NUMERIC

次のページに続く

表 7 – 前のページからの続き

Data types	Oracle Enterprise	EDB Postgres Advanced Server
String	CHAR, VARCHAR2, CLOB, NCLOB, NVARCHAR2, NCHAR, LONG (deprecated)	CHAR OK , VARCHAR OK , CLOB OK , NCLOB OK , NVARCHAR2 OK , NCHAR, CHARACTER, TEXT, CHAR, VARYING, CHARACTER VARYING, VARCHAR
Binary	BLOB, RAW, LONG RAW (deprecated), BFILE	BLOB OK , RAW OK , LONG RAW OK , BYTEA (no compatible type for BFILE)
Date/time	DATE, TIMESTAMP (with/without TIMEZONE), INTERVAL	DATE OK , TIMESTAMP (with/without TIMEZONE), INTERVAL OK , TIME (with/without TIMEZONE)
Boolean	Not Available	BOOLEAN
ROWID	ROWID	ROWID
XMLTYPE	XMLTYPE	XMLTYPE
Key-value	Requires NSWLDB which is a separate database program	Yes, is integrated into the core database
JSON	Use VARCHAR2, CLOB, and BLOB with is_json check constraint.	JSON and fast binary JSONB with 58 JSON operators, functions and relational json converters
Spatial / Geospatial	Yes	Yes
Other	IMAGE, AUDIO, VIDEO, DICOM	ENUM, POINT, LINE, LSEG, BOX, PATH, POLYGON, CIRCLE, CIDR, INET, MACADDR, BIT, UUID, XML, arrays, composites, ranges, custom
Data domains	Yes	Yes

インデックス

サポートされているさまざまなデータ型と、これらのデータ型を使用する新しいワークロードに最適なパフォーマンスを提供するには、データベースがさまざまなインデックスもサポートする必要があります。Postgres は、この点でややユニークで、特にその GiST インデックスにより、新しいデータ型に特化したインデックスを簡単に開発できます。

Indexes	Oracle Enterprise	EDB Postgres Advanced Server	Notes
B-Tree	Yes	Yes	NA
Hash	Yes	Yes	NA
Expressions	Yes	Yes	NA
Partial	Yes	Yes	NA
Reverse	Yes	No	Postgres では、ファンクションインデックスを使用してフィールドの順序を逆にすることができます。
Bitmap	Yes	No	BRIN インデックスの使用は、一部のユースケースに適している場合があります。
Block Range Index	Yes	Yes	NA
GiST Easy creation of specialized indexes	No	Yes	NA
GIN Custom inverted indexes	No	No	NA
K-nearest-neighbor	Yes	No	パッケージ DMBS_DATA_MINING と Spatial オプションを使用して、Oracle で利用できます。
Full-text search	Yes	Yes	NA
Spatial	Yes	Yes	無料の PostGIS 拡張機能を使用して Postgres で利用できます。

SQL 機能

EDB Postgres Advanced Server は、ANSI- SQL:2008 標準に強く準拠しています。また、トランザクション DDL もあり、テーブル作成など、DDL への大きな変更のバックアウトをサポートします。データベースまたはテーブルスペースの追加/ドロップから復旧することはできませんが、他のすべてのカタログ操作は元に戻すことができます。この機能は、スキーマのアップグレードなどの複雑な作業を実行する際の保護によく使用されます。このような変更をすべてトランザクションブロックに配置すると、それらがすべてアトミックに適用されるか、まったく適用されないことを確認できます。これにより、スキーマ変更におけるタイポグラフィまたはその他の同様のエラーによってデータベースが破損する可能性が低くなります。これは、間違いが関係キーを破壊する可能性がある複数の関連テーブルを変更する場合に特に重要です。

SQL capabilities	Oracle Enterprise	EDB Postgres Advanced Server
Union	Yes	Yes
Intersect	Yes	Yes
Except	Yes	Yes
Inner Joins	Yes	Yes

次のページに続く

表 9 – 前のページからの続き

SQL capabilities	Oracle Enterprise	EDB Postgres Advanced Server
Outer Joins	Yes	Yes
Inner Selects	Yes	Yes
Merge Joins	Yes	Yes
Common Table Expressions	Yes	Yes
Windowing Functions	Yes	Yes
Parallel Query	Yes	Yes
Query Hints	Yes	Yes
Transactional DDL	No	Yes
Alter Session	Yes	Yes
Dynamic SQL	Yes	Yes

SQL 拡張機能

Oracle には、Oracle ユーザーに人気の SQL 拡張機能が多数あります。SQL 言語の標準ではありませんが、DBA と開発者にユーティリティと利便性を提供します。EDB Postgres Advanced Server は、EDB 顧客が最も頻繁に要求するものをサポートしています。

Oracle Enterprise extension	EDB Postgres Advanced Server
DUAL	Yes
DECODE	Yes
ROWNUM	Yes
SYSDATE	Yes
SYSTIMESTAMP	Yes
NVL, NVL2	Yes
(+) Syntax for Outer Joins	Yes

1.5.3 Comparison of database operations-related capabilities

移行の実現可能性を分析するときは、これらの操作関連の機能を検討します。

高可用性

ミッションクリティカルなワークロードは、常に動作し続ける必要があり、メンテナンスのための計画的なダウンタイムでさえもほとんど許容されません。この高可用性の需要には、以下を提供するソリューションが必要です。

- システム内の単一障害点を排除するための高速レプリケーションと冗長性

- システムに障害が発生した場合にもシステムの機能を継続するための障害検出と自動フェイルオーバー
- 障害イベントの後に DBA を支援するデータとシステムのリカバリー

Oracle Enterprise	EDB Postgres Advanced Server
Data Guard	Yes Using streaming replication and log shipping
Active Data Guard	Yes Failover Manager
Flashback (querying)	No
Flashback (undo operations)	No
Backup and recovery tools	Yes Barman or pgBackRest
Point-in-time recovery	Yes

パフォーマンスとスケーラビリティ

さまざまな方法でデータベース操作を最適化して、より高いパフォーマンスを提供できます。たとえば、接続プーリングは、毎回新しい接続を作成するのではなく、繰り返し要求を行うアプリケーションのデータベースへのオープン接続を維持する一般的な方法を指します。データレプリケーションでは、マルチプルのエンドユーザーアプリケーションで情報を同時に利用できるようにすることで、パフォーマンスを向上させることができます。これらのパフォーマンスの向上は、データベースの機能強化とさまざまな外部補完ソリューションを介して実現できます。

Performance/scalability	Oracle Enterprise	EDB Postgres Advanced Server
Connection pooling	Yes	Yes
5-9s High availability	Yes Using Real Application Clusters (RAC)	Yes Using EDB Postgres Distributed
Write scalability	Yes RAC	No
Read scalability	Yes Using Active Data Guard or RAC	Yes Achieved with read replicas
In-memory database	Yes	No
Multi-master read/write solution	Yes Using Advanced Replication, Streams, or GoldenGate	Yes EDB Postgres Distributed or Replication Server
Columnar store	Yes Using in-memory column store	No
CPU and I/O resource limits	Yes	Yes

セキュリティ

データベースセキュリティは、安全な接続、パスワード管理、アクセス制御、物理データの暗号化、監査などを含む多くの側面で構成されています。オープンソースおよび商用データベースの中で、EDB Postgres Advanced Server は最も安全なものの 1 つであり、PCI DSS の広範なサポートが含まれています。

Security	Oracle Enterprise	EDB Postgres Advanced Server
Authentication systems support	Yes LDAP, SSL, RADIUS, PAM, Kerberos, GSSAPI, SSPI	Yes LDAP, SSL, RADIUS, PAM, Kerberos, GSSAPI, SSPI
DB connection encryption	Yes	Yes
DB connection white lists	Yes Using before connect triggers	Yes
DB connection black lists	Yes Using before connect triggers	Yes
Profiles for passwords	Yes	Yes
Server code obfuscation	Yes	Yes
ANSI standard SQL GRANT/REVOKE	Yes	Yes
Column-level permissions	Yes	Yes
User/group/role support	Yes	Yes
Virtual Private Database (VPD) / Row Level Security (RLS)	Yes	Yes
View security barriers	No	Yes
Transparent Data Encryption	Yes	Yes
Data masking	Yes	No
Data redaction	Yes	Yes
Real Application Security	Yes	Partial Only DBMS_RLS functionality
Database Vault	Yes	No
Audit Vault and Database Firewall	Yes	Partial Database firewall only (SQL/protect)
Advanced Security	Yes	Yes Multiple options available; see データ暗号化オプション
Auditing	Yes	Yes
Fine-grained auditing	Yes	No
Data encryption toolkit	Yes	Yes DBMS_CRYPT package provide with compatibility for Oracle

データ暗号化オプション

次のデータ暗号化オプションは、アプリケーションのニーズに応じて、さまざまなレベルと粒度の保護を提供します。

pgcrypto を使用する

- Postgres contrib モジュール。
- 選択したテーブルの列に適用されます。
- 暗号化されたフィールドを検索またはインデックス付けすることはできません。
- テーブル作成時に暗号化を適用する必要があるため、事前の計画が必要です。
- アプリケーションは、データベースとの交換が暗号化されたままであるように、暗号化/復号化を処理する必要があります。
- DBA はデータを明確に見ることができません。

DBMS_CRYPTO を使用する

- 同じ機能と制限を持つ pgcrypto の Oracle 互換ラッパー。
- 複数の暗号アルゴリズムをサポートします。
- DES、3DES、AES、および AES128。
- MD4、MD5、および SHA-1 ハッシュファンクション。
- 暗号的に強力なランダム値を生成します。

EDB Transparent Data Encryption を使用する

- ディスクに保存されているすべての Postgres ユーザーデータを暗号化します。
- データベースサーバーおよびバックアップストレージ上のオペレーティングシステムファイルのデータの不正な表示を防止します
- アプリケーションの変更または更新されたクライアントドライバはありません。
- 柔軟なキー管理アーキテクチャ。

ディスクパーティションの暗号化を使用する

- ファイルシステムのディスクパーティションは、OS によって暗号化/復号化されます。
- 一時ファイルを含む、データベースパーティション内のすべてのファイルを保護します。
- データはファイルシステムから読み取られるときに復号化されます。これにより、DBA はデータを参照できるため、ロールと権限をロックダウンできます。
- アプリケーション開発者に対して透過的、たとえば、Red Hat Enterprise Linux は、ディスクフォーマット LUKS での Linux 統一キー設定をサポートしています。

ファイルシステムレベルを使用する

- 個々のファイルまたはディレクトリは、ファイルシステムによって暗号化されます。

- ファイルベースのキー管理が必要です。
- 暗号化されたファイルの個々の管理たとえば、暗号化された形式でもインクリメンタルバックアップ。
- 公開キー暗号化を使用してアクセス制御を適用できます。
- 暗号化キーは、それらによって復号化されるファイルが開いたままである間のみメモリに保持されます。
- アプリケーション開発者に対して透過的 [eCryptfs for Linux](#) など。

Thales CipherTrust Transparent Encryption を使用する

- EPAS データベースとバックアップの保存データを保護します。
- 集中的なキー管理に基づいたファイルシステムレベルの暗号化。
- 特権ユーザーのアクセス制御。
- 詳細なデータアクセス監査ログ。

統合

今日のデータセンターは、一般に 1 つ以上のリレーショナルと多くの非リレーショナルデータベースソリューションで構成され、データタイプとアプリケーションに基づいて特定のワークロードを処理します。リレーショナルデータベースは、インフラストラクチャ全体で他の同様の異なるデータベースソリューションに接続するためのさまざまなメカニズムを使用して、マルチプルのソースからデータを接続し、凝集したデータファブリックを作成します。場合によっては、データベースは、データ統合を強化する特定の機能を使用して設計されています。データベースベンダーは、データベースを他のベンダーソリューションに接続できるようにするアダプターも開発します。

Integration	Oracle Enterprise	EDB Postgres Advanced Server
Database links	Yes	Yes
Native asynchronous log-based replication	Yes	Yes
Native synchronous log-based replication	Yes	Yes
Session-based synchronous replication*	No	Yes
Distributed transactions	Yes	No
Distributed queries	Yes	Yes
Integration with SQL Server	Yes Oracle Database Gateway	Yes Replication Server
Integration with Hadoop	Yes Oracle Data Integrator	Yes HDFS Foreign Data Wrapper
Integration with MongoDB	Yes Oracle GoldenGate	Yes MongoDB Foreign Data Wrapper
Integration with MySQL	Yes Oracle Database Gateway	Yes MySQL Foreign Data Wrapper or Migration Toolkit

次のページに続く

表 14 – 前のページからの続き

Integration	Oracle Enterprise	EDB Postgres Advanced Server
Integration with Sybase	Yes Oracle Database Gateway	Yes Migration Toolkit

*データベースに接続されているセッションに応じて、一部のトランザクションを同期的にコミットし、その他のトランザクションを非同期にコミットすることが可能であり、多くの場合役立ちます。

管理

大企業では、多くの場合、数百以上に大規模なデータベースを展開しています。DBA には、これらのデータファームを簡単かつ迅速に維持し、複数のデータベースにわたって大量の操作を実行するためのツールが必要です。モニタリング、調整、管理、およびアラートのための完全な機能を備えたカスタマイズ可能なグラフィカルコンソールは、DBA が自分の責任の基本を実行するために重要です。管理には、操作タスクで DBA をサポートするデータベース内の機能とデータベース外部のツールの両方が含まれます。

Management	Oracle Enterprise	EDB Postgres Advanced Server
CLI	Yes SQL*Plus	Yes EDB*Plus
Bulk data loader	Yes SQL*Loader	Yes EDB*Loader
Enterprise management	Yes Oracle Enterprise Manager	Yes Postgres Enterprise Manager
System catalog views	Yes	Yes (See Appendix C)
Point-in-time recovery (PITR)	Yes	Yes
Online backup	Yes	Yes
Online reorganization	Yes	No
Automatic memory management	Yes	No
Automatic storage management	Yes	No
Automatic undo management	Yes	No
Diagnostics package	Yes	Yes EDB Wait Sates
Tuning package	Yes	Yes Tuning Wizard, Index Advisor, Postgres Expert in PEM
SQL query profiler	Yes	Yes Available in PEM

EDB Postgres Advanced Server 互換カタログビュー

EDB Postgres Advanced Server は、Oracle データディクショナリビューと互換性のある方法でデータベースオブジェクトに関する情報を提供する 90 を超える Oracle カタログビューを提供します。

ALL_ Views	DBA_ Views	USER_ Views
ALL_ALL_TABLES	DBA_ALL_TABLES	USER_ALL_TABLES

次のページに続く

表 16 – 前のページからの続き

ALL_ Views	DBA_ Views	USER_ Views
ALL_COL_PRIVS	DBA_COL_PRIVS	USER_COL_PRIVS
ALL_CONSTRAINTS	DBA_CONSTRAINTS	USER_CONSTRAINTS
ALL_CONS_COLUMNS	DBA_CONS_COLUMNS	USER_CONS_COLUMNS
ALL_DB_LINKS	DBA_DB_LINKS	USER_DB_LINKS
ALL_DEPENDENCIES	DBA_DEPENDENCIES	USER_DEPENDENCIES
ALL_DIRECTORIES	DBA_DIRECTORIES	
ALL_INDEXES	DBA_INDEXES	USER_INDEXES
ALL_IND_COLUMNS	DBA_IND_COLUMNS	USER_IND_COLUMNS
ALL_OBJECTS	DBA_OBJECTS	USER_OBJECTS
ALL_PART_KEY_COLUMNS	DBA_PART_KEY_COLUMNS	USER_PART_KEY_COLUMNS
ALL_PART_TABLES	DBA_PART_TABLES	USER_PART_TABLES
ALL_POLICIES	DBA_POLICIES	USER_POLICIES
	DBA_PROFILES	
ALL_QUEUES	DBA_QUEUES	USER_QUEUES
ALL_QUEUE_TABLES	DBA_QUEUE_TABLES	USER_QUEUE_TABLES
	DBA_ROLES	
	DBA_ROLE_PRIVS	USER_ROLE_PRIVS
ALL_SEQUENCES	DBA_SEQUENCES	USER_SEQUENCES
ALL_SOURCE	DBA_SOURCE	USER_SOURCE
ALL_SUBPART_KEY_COLUMNS	DBA_SUBPART_KEY_COLUMNS	USER_SUBPART_KEY_COLUMNS
ALL_SYNONYMS	DBA_SYNONYMS	USER_SYNONYMS
ALL_TABLES	DBA_TABLES	USER_TABLES
ALL_TAB_COLUMNS	DBA_TAB_COLUMNS	USER_TAB_COLUMNS
ALL_TAB_PARTITIONS	DBA_TAB_PARTITIONS	USER_TAB_PARTITIONS
ALL_TAB_PRIVS	DBA_TAB_PRIVS	USER_TAB_PRIVS
ALL_TAB_SUBPARTITIONS	DBA_TAB_SUBPARTITIONS	USER_TAB_SUBPARTITIONS
ALL_TRIGGERS	DBA_TRIGGERS	USER_TRIGGERS
ALL_TYPES	DBA_TYPES	USER_TYPES
ALL_USERS	DBA_USERS	USER_USERS
ALL_VIEWS	DBA_VIEWS	USER_VIEWS
ALL_VIEW_COLUMNS	DBA_VIEW_COLUMNS	USER_VIEW_COLUMNS

表にリストされているビューに加えて、次のビューも提供されます。

- DBMS_AQ_STAT_DATABASES
- DBMS_AQ_STAT_MESSAGES
- DBMS_AQ_STAT_QUEUES
- DBMS_AQ_STAT_WAITERS
- PRODUCT_COMPONENT_VERSION
- V\$VERSION

展開オプション

パブリック、プライベートクラウド、仮想化の進歩に伴い、エンドユーザーのデータベース展開オプションの範囲が増加しました。次は、Oracle Enterprise および EDB Postgres Advanced Server の使用可能な展開オプションのスナップショットです。

Deployment options	Oracle Enterprise	EDB Postgres Advanced Server
On-premises hardware	Yes Intel, AMD, IBM Power, Sun ultraSPARC	Yes Intel, AMD, IBM Power
On-premises virtual	Yes With restrictions	Yes
On-premises cloud, private cloud	Yes Oracle Cloud on an Oracle stack including infrastructure and platform (middleware and database)	Yes Consumption-based solution in HPE GreenLake Database with EDB Postgres
Public cloud - managed service	Yes Oracle Cloud on a proprietary stack including infrastructure and platform (middleware and database)	Yes BigAnimal
Public cloud - self managed	Yes	Yes
Hybrid cloud	Yes	Yes
Kubernetes/containers	Yes	Yes

1.5.4 Comparison of application development capabilities

データベースは今日のデータドリブンエンタープライズの基盤であり、アプリケーションのデータ集約度はますます高まっています。ベンダーは、データベースソリューションを継続的に強化して、選択を行う柔軟性と複雑なタスクを実行する簡単な方法を求めるアプリケーション開発者のニーズをサポートしています。たとえば、トリガー、ファンクション、ストアドプロシージャで複数のサーバーサイド言語のサポートを提供できるデータベースは、開発者にクライアント、中間層、およびデータベースサーバーの両方のプログラミングで言語を選択するオプションを提供します。ユーザー定義オブジェクトタイプのようなオブジェクト指向機能により、データベースにデータの現実世界の表現形式を格納できるため、開発がより簡単、より速く、より理解しやすくなります。

Application development	Oracle Enterprise	EDB Postgres Advanced Server
IDE	Yes SQL Developer	Yes Postgres Enterprise Manager
Database server programming language	Yes PL/SQL (block structured language)	Yes EDB SPL (PL/SQL compatible) (block structured language)

次のページに続く

表 18 – 前のページからの続き

Application development	Oracle Enterprise	EDB Postgres Advanced Server
Additional programming languages for database server stored procedures, triggers, and functions	Yes Java, C, .NET (C#, Visual Basic)	Yes PL/pgSQL (PostgreSQL's procedural language), C, C++, PL/Perl, Python, PL/Tcl
JDBC support	Yes	Yes EDB JDBC Connector
ODBC support	Yes	Yes EDB ODBC Connector
.NET support	Yes	Yes EDB .NET Connector
OCI support	Yes	Yes EDB OCL Connector
PL/SQL debugger	Yes SQL Developer	Yes Postgres Enterprise Manager
Stored procedures	Yes	Yes
Named parameter notation for stored procedures	Yes	Yes
Triggers	Yes	Yes
REF cursors	Yes	Yes
Anonymous blocks	Yes	Yes
Bulk collect/bind	Yes	Yes
Associative arrays	Yes	Yes
Nested tables	Yes	Yes
VARRAYS	Yes	Yes
Hierarchical queries	Yes	Yes
Parallel query	Yes	Yes
PL/SQL supplied packages	Yes	Yes (See <i>EDB Postgres Advanced Server 互換パッケージのサポート</i>)
PRAGMA RE-STRICT_REFERENCE	Yes	Yes
PRAGMA EXCEPTION_INIT	Yes	Yes
PRAGMA AUTONOMOUS_TRANSACTION	Yes	Yes
User-defined functions	Yes	Yes
User-defined objects	Yes	Yes
User-defined exceptions	Yes	Yes

EDB Postgres Advanced Server 互換パッケージのサポート

EDB は、パッケージで最も人気のある機能に焦点を当てています。一部のパッケージでは、すべての Oracle 機能がサポートされているわけではありません。具体的な詳細については、[EDB Postgres Advanced Server ドキュメント](#)を参照してください。

Package name	Package description
DBMS_ALERT	アラートを介してデータベースイベントの非同期通知を許可する機能。このパッケージとトリガーを使用すると、アプリケーションはデータベース内の値が変更されるたびに自分自身に通知できます。
DBMS_ASSERT	ユーザー入力をサニタイズおよび検証して、アプリケーションでの SQL インジェクションから保護することができる機能。
DBMS_AQ	データベース統合された非同期メッセージキューイングは、アクティビティを通信し、さまざまな情報ペイロードを交換することにより、企業全体でアプリケーションを統合するための柔軟なメカニズムを提供します。
DBMS_AQADM	キューとキューテーブルを作成および管理するプロシージャ。
DBMS_CRYPTO	保存されたデータを暗号化および復号化する機能。
DBMS_JOB	DBMS_SCHEDULER に置き換えられましたが、古い Oracle アプリケーションとの互換性のために含まれています。
DBMS_LOB	ラージオブジェクト値へのアクセスと操作を許可するファンクション。
DBMS_LOCK	ロック管理サービスへのファンクションインターフェイス。
DBMS_MVIEW	マテリアライズドビューとその依存関係を管理および更新するプロシージャ。
DBMS_OUTPUT	アプリケーションまたはデバッグに使用するストアドプロシージャ、パッケージ、およびトリガーからメッセージを送信できます。
DBMS_PIPE	同じデータベースインスタンス内の 2 つ以上のセッションが相互に通信できるようにする機能。
DBMS_PROFILER	ストアドプロシージャのワークロードをプロファイリングし、パフォーマンスのボトルネックを特定する機能。
DBMS_RANDOM	ランダムなテキスト、数値、日付値を生成する便利なファンクション。
DBMS_REDACT	リダクションにより、ユーザーは機密データのすべてまたは一部が表示されなくなります。
DBMS_RLS	データベースに行レベルのセキュリティ機能を実装し、ユーザーが同じアプリケーション内でお互いのデータを表示できないようにブロックします。
DBMS_SCHEDULER	データベース内で無人反復タスクを作成および実行するジョブスケジューラー機能。
DBMS_SQL	プロシージャで動的 SQL の使用を許可して、アプリケーションは実行時まで不明なパラメーターテーブル名などを使用して SQL ステートメントを実行できます。
DBMS_SESSION	ロールを有効または無効にする機能を持つファンクション。
DBMS_UTILITY	さまざまなランタイム操作とメタデータに関する情報をデータベースから取得するためのファンクション。
DBMS_XMLDOM	DOM ドキュメントを作成するためのファンクション。
HTF	HTML タグを生成する機能。
HTP	HTML タグを生成するプロシージャ。

次のページに続く

表 19 – 前のページからの続き

Package name	Package description
UTL_ENCODE	ホスト間の転送を目的としたデータの Base64 エンコードおよびデコードを実行する機能。
UTL_FILE	データベースプロシージャが I/O ストリーム方法でオペレーティングシステムテキストファイルを読み書きできるようにします。
UTL_HTTP	HTTP 呼び出しを行って Web サーバー上の情報にアクセスできる機能。
UTL_MAIL	添付ファイル、CC、BCC を含むデータベースからメールを作成、管理、および送信する機能。
UTL_RAW	生のデータ型の操作をサポートするファンクション。
UTL_SMTP	RFC821 仕様に従って、SMTP を介してメールを送信するための機能。
UTL_TCP	PL/SQL アプリケーションが TCP/IP を使用して外部 TCP/IP ベースのサーバーと通信できるようにするプロシージャとファンクション。
UTL_URL	URL 文字列をエスケープおよび「アンエスケープ」するためのファンクション。

1.5.5 Comparison of nonrelational data support

Nonrelational data	Oracle Enterprise	EDB Postgres Advanced Server
Spatial/location/graph	Yes	Yes
JSON support	Yes Text based	Yes Text and high-performance binary based.
Key-value store	NoSQL database	Yes
Support for XML namespaces, DOM, XQuery, SQL/XML, and XSLT	XML DB	Partial XML DOM support is provided via the DBMS_XMLDOM package.
Compression (tables, files, network, and backups)	Yes	No Postgres performs compression of TOASTED rows.
Partitioning	Yes	Yes
Hadoop integration	Yes ETL via Data Integrator Application Adapter for Hadoop	Yes Real-time join with relational data with HDFS Foreign Data Wrapper.
MongoDB integration	Yes Golden Gate adapter	Yes Read/write/join with MongoDB Foreign Data Wrapper.
Cube, rollup, and grouping sets	Yes	Yes
Transportable cross-platform tablespaces	Yes	No
Full-text search	Yes	Yes
Advanced compression	Yes	No

1.5.6 Notable differences

Oracle と EDB Postgres Advanced Server の多くの違いはまだ対処されていないか、頻繁に使用されるため注目に値します。

Oracle Enterprise	EDB Postgres Advanced Server
MERGE	Yes Adds support for WHERE clauses to the UPDATE and INSERT of MERGE commands along with FORALL...MERGE commands for compatibility with Oracle beyond community PostgreSQL.
Advanced queuing	Yes
Nested procedures/functions	Yes
Pipelined functions	No Pipelined functions are used for table functions. Table functions can be implemented in Postgres via SETOF returning functions. In Postgres, data is returned only after the function completes.
Empty string = NULL	No Empty string = !NULL
Performs many implicit data type conversions such as a number to a string	Partial Many data types need to be explicitly cast to the other data type or an error occurs.