Migration Overview

リリース *1.0.0*

EDB

2022 年 06 月 07 日

目 次

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

- コストの削減
- ライセンスの柔軟性
- オープンソースへの取り組み
- アプリケーションとテクノロジーの近代化と統合への取り組み
- 新しいプラットフォームと展開環境に移行したい

Postgres は、その豊富な機能、品質、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選択されることがよくあります。EDB は、PostgreSQL のディストリビューションである EDB Postgres Advanced Server、Oracle との組み込みの互換性、および組織が Oracle からより簡単に、ビジネスへの影響を少なくすることを支援することを特に対象とした移行ツールとサービスのセット。

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ことです。移行を実行するときに考慮すべき要素、移行の手順、および EDB が移行を支援するために提供する機能、ツール、サービスを特定して説明します。Oracle と EDB Postgres Advanced Server の主な類似点と相違点、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。

第1章　Factors to consider when migrating

移行を分析および計画するときは、次の要因について考えます。

- 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

1.1 スキーマの移行考慮事項

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。Postgres がこれらのオブジェクトを作成するために提供する構文は、多くの場合 Oracle の構文と似ていますが、移行時に対処する必要がある違いがあります。Oracle と Postgres では、理解することが重要な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ります。

データ型は、テーブルおよびデータベースで使用される他のオブジェクトで使用されます。移行、Oracle のデータ型が Postgres のデータ型にどのように対応するかを考える必要があります。

データベーススキーマには、データベースに直接記述されたパッケージ、プロシージャ、ファンクション、トリガーなどのコードオブジェクトも含まれています。Oracle では、コードオブジェクトは Oracle の PL/SQL 言語で記述されます。PostgreSQL には、Oracle 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 と呼ばれる 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る場合があります。

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の選択、挿入、更新、および削除のためにそのバージョンの構造化照会言語（SQL）で使用する特定の規則とキーワードがあります。

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。考慮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、許容できるダウンタイムによって異なります。バルクロードが適切かどうかを判断する必要があります。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、そこから先に進みます。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稼働し続けなければならない場合があります。ソースデータベースはまだ更新を受け取っているため、そのデータを引き続きターゲットデータベースに移動できるようにします。

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うちに、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終了するときは、移行が完全に成功しなかった場合に備えて、フォールバックのポジションが必要になることがよくあります。移行されたシステムで運用を開始した後、移行されたデータベースに適用されるデータ変更とレガシーデータベースが同期されている場合、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をタイムリーに元のソースデータベースに戻す必要があります。

データの移行には、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）のニーズなど、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プルし、変換して、ターゲットデータベースにロードします。

データをターゲットシステムに配置したら、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検証します。

1.3 インフラストラクチャーの考慮事項

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にデプロイされている場合、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できます。ただし、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る組織が増えています。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を決定できます。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データベースに移行してから、移行したデータベースをクラウドに移動する場合があります。

最終状態として、またはクラウドへの踏み台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、オペレーション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が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もニーズは似ていますが、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より柔軟に追加および削除できます。

考慮すべきもう 1 つの要因は、多くのレガシ 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たことです。サーバーオプションをターゲットにする可能性が高い場合、この要因は移行に使用されるツールとアプローチの種類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こと、または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 で使用可能なデプロイ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 Postgres コンテナ
- プライベートクラウドインスタンス

従来、データベースはパフォーマンスを最適化するために物理サーバーにインストールされました。時間が経つにつれて、仮想化テクノロジーが向上するにつれて、データベースを仮想マシンにインストールす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います。。

クラウドへの移行の場合、 EDB Postgres Advanced Server の展開オプションは次のとおりです。

- 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の EDB Postgres Advanced Server
- クラウドネイティブ PostgreSQL（CNP）

ますます多くの組織が、マネージドデータベースサービス (DBaaS としても知られる) に移行して、このサービスがビジネスにもたらすメリットを活用しています。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、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細かく制御したい場合は、クラウドベースのサーバーインスタンスに自分で EDB Postgres Advanced Server をインストールして管理できます。EDB のクラウドネイティブ PostgreSQL（CNP）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 クラスターをデプロイして管理できます。データベース用のコンテナの普及はまだ初期段階です。ただし、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ーの使用が増加しています。Kubernetes クラスターはクラウド内に立てることができます。実際、主要な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た可能性があることを考慮してください。ターゲットシステムに移行するときは、同様の最適化を異なる手法を使用して行う必要があります。

1.4 アプリケーションの移行考慮事項

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）でデータベースクエリを発行します。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で採用されました。ただし、それぞれには標準から逸脱し、互いに異なる機能があります。そのため、移行作業の一環として、アプリケーションコードに埋め込まれた SQL コードの変換が静的または動的に生成される可能性があります。

さらに、アプリケーションは通常、レガシーデータベースシステムおよびオペレーティング環境で適切に動作するように調整および構成されています。古いシステムから新しいシステムに移行する場合、新しいデータベースとオペレーティング環境で適切に実行できるようにアプリケーションに変更を加える必要があります。

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に移行すると、新しいデータベースを操作するための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 固有のツールを Postgres 固有のツールに置き換える必要があります。考慮すべき運用上の主な側面には、高可用性、バックアップとリカバリ、監視と管理、セキュリティ、暗号化などがあります。

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では、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必要があります。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生じる場合があります。クラウドへの移行の場合、運用とメンテナンスの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によって処理されますが、その他は引き続きお客様の責任となります。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。

1.6 ベストプラクティスの考慮事項

以下は、Oracle から Postgres への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べきいくつかのベストプラクティスです。

- 移行の手順をスキップしたり、ざっと読んだりしないでください。-計画を立ててください。 - 機能要件と非機能要件を特定して理解する。 - ソリューション設計を早期に完了して、IT リソースのニーズを理解する。
- 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。
- アプリケーション開発者とシステム管理者を移行の旅の早い段階でデータベースチームと連携させます。
- 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設定します。
- 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延期します。
- 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ついてトレーニングされていることを確認します。

第2章　Migration techniques

高レベルでは、Oracle から Postgres へのデータベース移行には次が含まれます。

- Oracle の拡張バージョンの SQL 標準から、一般的に標準に準拠した Postgres 準拠のバージョンにスキーマを変換します。
- Oracle 固有のデータ型定義を Postgres 固有のものに変換する
- クエリ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
- 移行されたデータベースが、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。

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、コストがかかり、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のない非常に小さなデータベースを除き、このアプローチは実用的ではありません。

2.2 データベース移行への自動化されたアプローチ

通常、純粋な手動アプローチの代わりに 2 つの自動化されたアプローチが使用されます。

_ネイティブ互換性_アプローチは、1 つのデータベースの機能を拡張し、API とプロトコルを含む SQL 標準の別のデータベース固有の実装を作成します。EDB Postgres Advanced Server は、次の領域で Oracle との互換性を提供する PostgreSQL の拡張バージョンです。

- Oracle 固有で構文互換性のある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
- 組み込みのネイティブ手続き言語としての 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 向けの Oracle と同様のツール EDB Postgres Advanced Server は、これらの領域の Oracle 固有の機能をすべて実装していません。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、自動化されたツールを使用して、Oracle の定義、クエリ、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す（または変換します）。EDBMigration Portal 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が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、オープンソースの移行ツールで使用可能な唯一のアプローチであり、他のベンダーが純粋なオープンソースの PostgreSQL をターゲットにしています。ただし、翻訳が必要な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ために必要な時間と労力が、スケジュールまたはリソースによってサポートされない場合があります。

これら 2 つの自動化アプローチを組み合わせることで、ほとんどの移行が大幅に簡素化されますが、通常、手動変換が必要な

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てください。

第3章　The database migration “journey”

データベースの移行は、9 つのステップから構成されます。

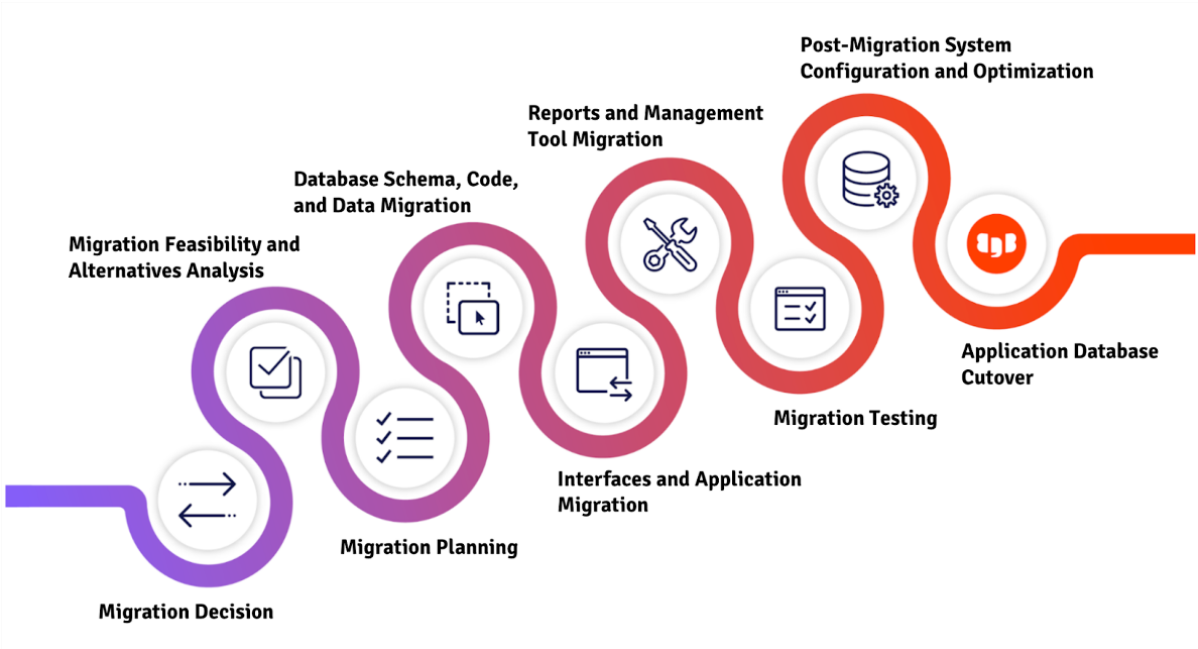

図 1: Migration journey

3.1 1. 移行することを決定

まず、組織は移行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使いすぎていると思うかもしれません。現在のデータベース技術では十分にサポートされていないイニシアチブがあるかもしれません。または、クラウドに移行したり、オープンソースソフトウェアを利用したりすることもできます。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取り組む必要があります。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。

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。

3.2.1 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

このレビューの一環として、移行可能なレガシーデータベースを持つアプリケーションを見ていきます。移行の優先順位と IT 戦略を調整して、これらが移行を検討するのに十分なほど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廃止することを決定する場合があります。ポートフォリオには、既存のデータベースでのみサポートされているために移行できないアプリケーションがある場合があります。

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、それらを移行することは、移行を完了するという組織のコミットメントに対応する必要があることに注意してください。時間も、リソースも、予算もかかります。

3.2.2 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正しいかを評価する

多くの場合、 EDB Postgres Advanced Server は Oracle を置き換える理想的な選択肢です。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より適切に対処できる要件があります。

ターゲット データベーステクノロジの適切なセットを選択したら、次の手順として、移行がどの程度複雑になるかをより詳細に評価します。EDB Postgres Advanced Server が潜在的なターゲットである場合、EDBMigration Portal を使用して、データベース内の Oracle スキーマと手続き型オブジェクトと 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。

3.2.3 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 レベルのパスを実行して、最初に移行するものを特定します。いくつかは非常に重要で、移動するとより多くのリスクがありますか？また、移行の難しさを見て、それに基づいて優先順位を付ける必要があります。

3.2.4 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ーの考慮事項 を参照してください。現在、オンプレミスで実行していますか？オンプレミスを維持しますか、それともクラウドに移行しますか?または、クラウドを利用していますが、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戻る必要がありますか?

3.2.5 潜在的な移行問題を特定する

旅の後半で発生する特定の互換性問題に対処するためのソリューションをより徹底的にレビューしますが、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 の移行については、詳細については 'Comparison of EDB Postgres Advanced Server with Oracle' を参照してください。

この手順の主な目的は次のとおりです。

- どのアプリケーションが移行に適しているかを理解します。
- 移行のためのさまざまなテクノロジーとベンダーのオプションを分析します。
- すべての移行を完了するための全体的なレベルの労力、時間、コストの全体像を理解する。

また、現在のテクノロジとプラットフォームを継続するコストと、新しいテクノロジとプラットフォームに移行するコストを検討します。

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、最も難しい移行を実行できれば、他のすべてが簡単になると考えているため、最も難しい移行を検討することがよくあります。ただし、通常は、自分にとって適度に重要であり、顧客にとって最も価値があるものではなく、最も難し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、最初の移行から学んだことなどに基づいて、次のものを選択します。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の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、移行を行うためにどれ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。

このステップ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予期しない問題が発生する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確かな知識を持つ人を含む移行チームを形成します。

3.4 4. データベーススキーマ、コード、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は、これが最初に頭に浮かぶことがよくあります。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されます。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。

3.4.1 スキーマを移動する

スキーマは、データベースにデータが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。EDB Postgres Advanced Server の Oracle 機能の互換性により、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解決されます。

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 Migration Portal 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修復ハンドラーを介していくつかの自動変換を実行します。

Migration Portal の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。

3.4.2 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、アプリケーションによって呼び出されるプロシージャやファンクション、またはデータベースが使用するトリガーなど、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に密接に関連付けられており、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、この手順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 の手続き言語（PL / SQL）は他のデータベースのネイティブビルトイン言語とは異なるため、特にソースデータベースに多数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で構成されている場合、これはデータベース移行で最も困難なタスクになる可能性があります複雑なコードの多くの行。これは、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 実装が重要になる場所です。これらの機能により、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。Migration Portal は、他のスキーマ DDL に加えて Oracle データベースの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。

3.4.3 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、必要なデータをすべて移動することです。このオプションは、データベースが比較的小さい場合、または本番移行の場合、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ます。

スナップショットレプリケーションを定期的に実行して、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更新できます。ただし、このような定期的な更新では、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環として、初期ロードは何らか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あり、ターゲットシステムとソースシステムで行われるデータの同期を維持する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のアプローチを使用して、2 つの主な方法では対応できない特別なデータ移行のニーズに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プルすると役立つ場合があります。

3.5 5. インターフェースとアプリケーションを移行する

新しく移行された 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次のものが含まれます。

- アプリケーションの接続文字列とドライバー（JDBC、ODBC、OCI、.NET など）の更新。JDBC や ODBC などのオープンスタンダード接続を使用するアプリケーションでは、通常、データベース接続文字列の変更と EDB ドライバーの選択のみが必要です。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ための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。

.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への呼び出しを 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。

- アプリケーションの埋め込み SQL を変換します。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場合があります。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ります。この場合、Postgres で動作するようにその SQL を変換する必要があります。

EDB EDB Postgres Advanced Server および EDB ドライバーに組み込まれた Oracle 機能に対する 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。Postgres ベースの回避策を実装するには、そのようなコードを更新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、ソースコードの移行とともに実行する必要がある実際のアプリケーションとその構成の移動がある場合があります。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応するためにさまざまな手法、ツール、構成が必要になる場合があります。

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。組織には、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があります。これらのアーティファクトの一部は、特に Oracle ベースのプロセスをサポートするように構築されているため、すべてが移行後のシステムに適用されるとは限りません。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、残したり、置き換えたり、更新したり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を更新するには、多くの場合、Oracle 用に構築された互換性のない SQL を更新して、ターゲットデータベースと互換性があるようにします。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更新することも含まれます。たとえば、SQL*Plus は Oracle の主要な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、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構築されています。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スクリプトを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 コマンドを理解する SQL *Plus と同様のユーティリティです。この機能と、 EDB Postgres Advanced Server の Oracle 機能の他の組み込み互換性により、 EDB Postgres Advanced Server で Oracle 用に構築された多くの既存のスクリプトを実行できます。 EDB Postgres Advanced Server ユーザーは、 EDB Postgres Advanced Server に組み込まれた Oracle との互換性機能で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 に慣れると、ユーザーは多くの場合、その機能を活用するために EDB*Plus の代わりに使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引き続き選択できます。

Oracle が提供するもう 1 つの一般的なインタフェースは、SQL*Loader です。これは、Oracle にデータをバルクロードするために使用されるツールです。EDB は、 EDB*Loader と呼ばれる似たユーティリティを提供し、 EDB Postgres Advanced Server にデータをロードします。 EDB*Plus と同様に、 EDB*Loader は、EDB Postgres Advanced Server で使用できる Oracle ユーザーに使い慣れたインターフェイスを提供します。このツールを使用すると、既存のデータ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 ネイティブメソッドも利用できます。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。

3.7 7. 移行をテストする

移行のサイズに関係なく、移行したアプリケーションが期待どおりに動作することを検証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行う必要があります。

- 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確認します。- 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でき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。これらのアプリケーションとデータベースのテストは、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、予想される運用ワークロードをシミュレートするワークロードと、運用環境に似た運用環境で実行します。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のに十分な範囲を提供する必要があります。

データの検証に関しては、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を比較するために使用されるツールとプロセスで、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、移行プロセスに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は、データを定期的に検証する計画を立てます。

この手順では、可用性要件への対応、バックアップと回復の操作、およびデータベースの監視と管理に使用するツールとプロセスを設定して実行することをお勧めします。これにより、運用環境での使用に備えて、ツールおよび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できます。

この手順の主な目標の 1 つは、運用システムへの最終的な移行に進む前に対処する必要がある問題を特定することです。問題を見つけて修正した場合は、さらにテストを実行する必要があります。運用前の移行テストの結果に満足したら、運用環境の最終的な準備を実行できます。

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、テストフェーズで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、システムを構成および調整します。システムを最適に実行するために、データベースパラメーターとアプリケーション設定の調整が必要になる場合があります。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を適用します。

このフェーズは、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や、各アクションを実行する権限を含む、高可用性、ディザスターリカバリー、およびセキュリティ要件への対応を終了する

時でもあります。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。

3.9 9. 完全なカットオーバー/ライブ

このフェーズは以下で構成されています。

- 実稼働データベースへのデータの移行を完了します。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は、ソースデータベースへのそれ以上の変更を無効にして、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のアプリケーションの実行で問題が発生した場合の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。このようにして、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。ターゲットデータベースへのデータの移動は、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ためのシステムの構成。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け取ったことを確認した後にのみ実行されます。
- 続行するかどうかが決定されるまで、新しいシステムをテストします。一定期間、新しいアプリケーションとデータベースをテストします。アプリケーションが新しい環境で適切に実行されていることを確認したら、アプリケーションが完全な実稼働用に「ゴー」であることを宣言できます。
- プロダクションへのカットオーバーの仕上げ。カットオーバーを完了して満足したら、通常は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。その時点では、新しい実稼働システムでのみ実行します。

第4章　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 がこれらの移行を実行した経験です。各フェーズで、移行を簡単にするツールとサービスを提供します。また、ビジネス要件について話し合い、移行の決定と最適なアプローチを支援できます。

4.1 移行で使用される機能

EDB が提供する移行を支援するツールとサービスを特定する前に、移行の各ステップで必要な**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。移行の旅のすべての手順は、経験豊富な個人が実行することから恩恵を受けますが、適切なツールセットを使用すると、それらの多くの手順を簡単に実行できます。

これらは、移行の旅の各ステップを正常に完了するために必要ないくつかの重要な種類のツールとスキルです。

4.1.1 実現可能性と代替案を分析する

- ソースデータベースからターゲットデータベースへの移行の実現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

4.1.2 移行を計画する

- 移行作業のレベルを見積もる機能
- 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

4.1.3 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 ベースのデータ移行を実行するツールと機能

4.1.4 インターフェー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ー/コネクターの互換性
- 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て変換するツールまたは能力

4.1.5 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

4.1.6 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
- 機能的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能力を支援するツール

4.1.7 移行後のシステムを最適化して構成する

- 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/承認の要件に対処および構成するためのツールと機能

4.1.8 完全なカットオーバー/稼働

- ロールバックをサポートするための逆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運用および保守するためのツールと能力

4.2 EDB 移行ソフトウェアとサービス

この表は、EDB がサポートする移行の旅のステップに提供する移行関連の機能のマッピングを示しています。表の後に、各機能の概要が記載されています。詳細については、EDB Web 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、次が適用されます。 **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** [かっこで囲まれた= Oracle との互換性]

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

[^1]：PEM は BigAnimal クラスターを監視できますが、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]：Failover Manager 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 PostgreSQL（CNP）デプロイメントには適用できません。[^3]：Postgres- BDR は、現在 BigAnimal およびクラウドネイティブ PostgreSQL オファリングでは使用できません。利用可能になると、その機能は BigAnimal サービスとクラウドネイティブ Postgres オペレーターにオプションとして直接統合されます。

4.2.1 EDB Postgres Advanced Server

EDB Postgres Advanced Server は PostgreSQL の拡張バージョンです。コア PostgreSQL のすべての機能が含まれているだけでなく、Oracle および他のエンタープライズグレードの機能との組み込みの互換性も含まれています。EDB Postgres Advanced Server は、次の領域で Oracle との互換性を提供します。

- Oracle 固有で構文互換性のある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
- 組み込みのネイティブ手続き言語としての 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 のビルトイン 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 で機能したものが EDB Postgres Advanced Server で機能するため、変換するものが少ないことを意味し EDB Postgres Advanced Server。 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、実装されている Oracle 機能との互換性は、最も一般的に使用される構造の一部です。さらに、EDB はメジャーリリースごとに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 Web 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、the EDB Postgres for Kubernetes operator を使用して Kubernetes 環境にデプロイできます。

4.2.2 Migration Portal

Migration Portal は、Oracle データベーススキーマと EDB Postgres Advanced Server の互換性を評価するために EDB が提供する無料のオンラインサービスです。Migration Portal 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、Migration Portal 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、結果。次に、結果を確認し、すべての DDL オブジェクトを検査し、エラーでオブジェクトを更新して問題を修正し、Migration Portal 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（つまり、100 % 互換性があると）、Migration Portal 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に役立ちます。

4.2.3 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です。ただし、Oracle から EDB Postgres Advanced Server への移行、その主な用途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。ソース情報のセットをターゲットデータベースにロードする方法を制御する豊富なコマンドオプションセットを提供します。1 回限りの試用期間がありますが、長期間使用するには EDB データベースのサブスクリプションが必要です。

4.2.4 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 Postgres にシングルマスターモードでデータを複製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 の代わりに、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。これは、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、これを使用して、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 回限りの試用期間がありますが、長期間使用するには EDB データベースのサブスクリプションが必要です。

4.2.5 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て、それらが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です。データベースを比較した後、ツールは比較レポート、違いのリスト、およびデータ操作言語（DML）SQL スクリプトを生成します。DML を適用して、任意のデータベースの不整合を修正できます。

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、または対処するデータの不整合があるかどうかを知るのに役立ちます。フォールバッ

クのためにデータを Oracle に移行する場合、LiveCompare は 2 つのデータベースに同じデータがあることを確認できます。

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

4.2.6 Oracle 互換のデータベースコネクタ

EDB は、Oracle 機能との互換性を含めるように拡張されたバージョンの PostgreSQL データベースドライバー（コネクター）を提供し、もともと Oracle が EDB Postgres Advanced Server で実行するように構築されたアプリケーションの実行をサポートします。次の Oracle 互換の Postgres ベースのコネクタを使用できます。

- EDB JDBC Connector
- EDB ODBC Connector
- EDB .NET Connector

EDB は EDB OCL Connector も提供して、Oracle Call Interface（OCI）ベースのアプリケーションを EDB Postgres Advanced Server に対して実行できます。

この表は、EDB 拡張コネクタが提供する重要な互換性機能の一部を強調しています。

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

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

4.2.7 その他の EDB ソフトウェア

4.2.8 EDB*Plus

EDB*Plus 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、SPL 無名ブロックおよび 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 用に既にビルドしたスクリプトを EDB Postgres Advanced Server データベースに対して使用できます。

4.2.9 EDB*Loader

EDB*Loader は、EDB Postgres Advanced Server の Oracle データベースと互換性のあるインターフェイスを提供する高性能なバルクデータローダーです。EDB*Loader ユーティリティは、Oracle SQL*Loader が提供するパラメーター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。

4.2.10 データベースリンクと dblink_ora ファンクション

Database links および データベースリンクと dblink_ora ファンクション edb_lk_asis_11 および edb_lk_asis_12 ファンクションは、 SQL クエリとデータベースファンクションコールを使用して Oracle から Postgres にデータをプルできるようにする Postgres 拡張機能です。これらのテクノロジーを使用して、Oracle データベースからデータを移行することもできます。これらの方法はどちらも Oracle Call Interface（OCI）を使用して Oracle に接続します。接続したら、 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、データを EDB Postgres Advanced Server データベースに挿入します。 ファンクションは、SQL クエリとデータベースファンクションコールを使用して Oracle から Postgres にデータをプルできる Postgres 拡張機能です。これらのテクノロジーを使用して、Oracle データベースからデータを移行することもできます。これらの方法はどちらも Oracle Call Interface（OCI）を使用して Oracle に接続します。接続したら、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、データを EDB Postgres Advanced Server データベースに挿入します。

これら 2 つのオプションのいずれかを使用する場合、通常は使いやすいデータベースリンクオプションが優先されます。ただし、dblink_ora ファンクションは、BLOB データや CLOB データなど、データベースリンクオプションでは移行でき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クエリでの Oracle 固有の機能の使用もサポートしています。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が提供されます。これらの機能は両方とも EDB Postgres Advanced Server で提供されますが、使用するには設定する必要があります。

4.2.11 EDB Postgres Enterprise Manager（PEM）

EDB Postgres Enterprise Manager は、EDB でもサポートされているオープンソースの pgAdmin 4 プロジェクトに基づく包括的なデータベース監視および管理アプリケーションです。PostgreSQL および EDB Postgres Advanced Server データベースとそれらが実行されるシステムを検査、監視、管理、およびクエリ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。データベースの正常性とパフォーマンスの監視に使用できる多くのレポート、ダッシュボード、プローブ、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、データベース移行の一部として使用され、必要に応じてパフォーマンスを監視および調整し、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。

4.2.12 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てプライマリ-スタンバイ展開アーキテクチャの高可用性を実現します。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、スタンバイデータベースノードへの Postgres プライマリデータベースノードの自動フェールオーバーを提供します。データベース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。

4.2.13 Postgres BDR

Postgres-BDR は、マルチマスターレプリケーションとデータ分散、およびネイティブ論理レプリケーションよりも最大 5 倍速いスループットを提供する PostgreSQL 拡張機能です。これにより、最大ファイブ・ナインの高可用性を備えた分散 PostgreSQL クラスターが有効になります。より極端な高可用性シナリオについては、Postgres- BDR と EDB Failover Manager をお勧めします。 Oracle から EDB Postgres Advanced Server の移行を実行するための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は、ターゲットシステムに展開してテスト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。Postgres- BDR を使用するには EDB サブスクリプションが必要です。

4.2.14 Barman （バックアップとリカバリーマネージャー）

Barman 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 がサポートおよび保守す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、優れたバックアップとリカバリの手順とツールが不可欠です。さらに、これらのツールは多くの場合、移行テストと関連する開発作業をサポートするプロセスに統合されます。Barman に加えて、EDB はサブスクリプションオファリングの一部として、人気のある Postgres のバックアップとリカバリーツールである pgBackRest もサポートしています。

4.2.15 EDB プロフェッショナルサービス

EDB は、データベース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ます。

- **移行評価サービス**により、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、データベースの移行を実行するのにどれくらいの時間と労力がかかるかを学びます。
- **エンタープライズアーキテクチャサービス**。移行したシステムの運用、インフラストラクチャ、および環境の要件を定義し、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。
- **ソリューション設計サービス**。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。

- **QuickDeployService** 。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、スケーラビリティ、および安全な方法で 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に合わせてデータベースを調整できます。
- **MonitoringBestPracticesService** : 移行後のデータベースの運用と保守のための優れた監視プラクティスを確立するのに役立ちます。
- **バックアップのベストプラクティスサービス**。データを決して失わないように、適切なバックアップと回復のプラクティスを確立できます。
- **EmbeddedPostgresSubjectMatterExpert（SME）サービス**。Postgres の専門知識でスタッフを強化し、移行および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支援する**

これらおよびその他の 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。

第5章 ‘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、2 つのデータベース技術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、多くの組み込みの Oracle 互換性機能を含む PostgreSQL の拡張バージョンです。これらの互換性機能により、コアの 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、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ものを感じてもらうことです。

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*
- *Nonrelational data support*
- *Notable differences*

このことを理解すると、 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である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も役立ちます。 EDB Postgres Advanced Server への移行を決定した後、一般的な移行の問題の詳細な回避策については、

Migration Portal **ポータルの 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。

5.1 Comparison of general characteristics

見込みユーザーは、Oracle のデータベースと EDB Postgres Advanced Server データベースを比較する際に、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 の専門家の間で議論するときに特に役立ちます。

5.1.1 用語

互換性の問題を詳しく見る前に、多くの SQL ベースの製品で使用される命名法が異な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
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。データディレクトリの場所またはポート番号の 2 つの方法で参照できます。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。

5.1.2 一般的な機能

Oracle と EDB Postgres Advanced Server は両方とも、原子性、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す、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す。どちらも System R に関する同じ IBM の研究から開発され、同じ問題の多く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

次のページに続く

表 2 – 前のページからの続き

General capabilities	Oracle	EDB Postgres Advanced Server (EPAS)
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

5.1.3 容量

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の設計、ワークロード、および予想される増加をサポートす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

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てください。

5.2.1 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。

✓ —この機能は Oracle と互換性のある方法で動作するため、既存の Oracle スキル、プログラムコード、またはデータを引き続き使用または移行できます。

5.2.2 テーブルとパーティショニング

データベース内の構造の範囲と、これらの構造を編成する際に DBA がどの程度の柔軟性を持っているかは、パフォーマンスだけでなく、メンテナンスやその他の操作要件に影響を与える可能性があります。たとえば、データベースをパーティション化する機能により、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分散すると、管理性、可用性、および負荷分散も向上します。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消費するランタイムクエリ、複雑な結合、または時間のかかる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単純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	Notes
Temporary tables	Partial	Postgres ではグローバル一時テーブルはサポートされていません。
Views	Yes	該当なし
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リフレッシュオプションを提供せず、フルリフレッシュのみを提供します。
Partitioning	Yes ✓	該当なし
Partition by range	Yes ✓	該当なし
Partition by hash	Yes ✓	該当なし
Partition by list	Yes ✓	該当なし
Subpartitioning	Yes ✓	該当なし
Interval partitioning	Yes ✓	該当なし
Partitioned indexes	No	該当なし
ANSI constraints	Yes	該当なし
Tablespaces	Yes	Oracle と Postgres では、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整理および指定するためにテーブルスペースが使用されます。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るため、事前に並べ替えられた構造からデータを読み取るときに同様のパフォーマンスが向上します。

次のページに続く

表 4 – 前のページからの続き

Oracle Enterprise	EDB Postgres Advanced Server	Notes
Sequences	Yes ✓	• Postgres では、一部のキーワードオプションが利用できません。 - 最大シーケンス値に若干の違いがあります。

5.2.3 データ型

データ型は、DBMS がシステムで情報を定義、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設定します。EDB Postgres Advanced Server は Oracle データ型との強力な互換性があり、拡張性が高いため、一般的になった新しい新しいデータ型とワークロードをすばやくサポートできます。

Data types	Oracle Enterprise	EDB Postgres Advanced Server
Type system	Static + dynamic (through ANYDATA)	Static integer NUMBER NUMBER ✓, DEC , NUMERIC, SMALLINT (16-bit), INT, BINARY_INTEGER, PLS_INTEGER, INTEGER (32-bit), BIGINT (64 bit)
Floating point	BINARY_FLOAT, BINARY_DOUBLE	BINARY_FLOAT ✓, BINARY_DOUBLE ✓, FLOAT, REAL (32-bit), DOUBLE PRECISION (64-bit)
Decimal	NUMBER	NUMBER ✓, DEC, DECIMAL, NUMERIC
String	CHAR, VARCHAR2, CLOB, NCLOB, NVARCHAR2, NCHAR, LONG (deprecated)	CHAR ✓, VARCHAR ✓, CLOB ✓, NCLOB ✓, NVARCHAR2 ✓, NCHAR, CHARACTER, TEXT, CHAR, VARYING, CHARACTER VARYING, VARCHAR
Binary	BLOB, RAW, LONG RAW (deprecated), BFILE	BLOB ✓, RAW ✓, LONG RAW ✓, BYTEA (no compatible type for BFILE)
Date/time	DATE, TIMESTAMP (with/without TIMEZONE), INTERVAL	DATE ✓ TIMESTAMP (with/without TIMEZONE), INTERVAL ✓, 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

次のページに続く

表 5 – 前のページからの続き

Data types	Oracle Enterprise	EDB Postgres Advanced Server
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

5.2.4 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、それらのデータ型を使用する新しいワークロードに対して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す。特に、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る GiST インデックスです。

Indexes	Oracle Enterprise	EDB Postgres Advanced Server	Notes
B-Tree	Yes	Yes	該当なし
Hash	Yes	Yes	該当なし
Expressions	Yes	Yes	該当なし
Partial	Yes	Yes	該当なし
Reverse	Yes	No	Postgres では、機能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	ユースケースによっては、 BRIN インデックスの使用が適している場合があります。
Block Range Index	Yes	Yes	該当なし
GiST Easy creation of specialized indexes	No	Yes	該当なし
GIN Custom inverted indexes	No	No	該当なし
K-nearest-neighbor	Yes	No	パッケージ DMBS_DATA_MINING および Spatial オプションを使用して Oracle で使用できます。
Full-text search	Yes	Yes	該当なし
Spatial	Yes	Yes	無料の PostGIS 拡張機能を使用して Postgres で利用できます。

5.2.5 SQL 機能

EDB Postgres Advanced Server は、ANSI- SQL：2008 標準に厳密に準拠しています。また、トランザクション DDL もあり、テーブルの作成など、DDL への大規模な変更でもバックアウトできます。データベースまたは表領域の追加/削除から回復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行うとき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入れると、すべてがアトミックに適用されるか、まったく適用されないようにすることができます。これにより、スキーマ変更の誤字やその他のエラーによってデータベースが破損する可能性が低下します。これは、間違いによりリレーショナルキーが破壊される可能性の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 ✓
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

5.2.6 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 の顧客が最も頻繁に求め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 ✓

次のページに続く

表 8 – 前のページからの続き

Oracle Enterprise extension	EDB Postgres Advanced Server
(+) Syntax for Outer Joins	Yes ✓

5.3 Comparison of application development capabilities

データベースは今日のデータ駆動型企業の基盤であり、アプリケーションはますますデータ集約型になっています。次に、ベンダーは、複雑なタスクを実行するための柔軟性とシンプル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するために、データベースソリューションを継続的に強化しています。たとえば、トリガー、ファンクション、およびストアドプロシージャに対して複数のサーバーサイド言語をサポートできるデータベースでは、開発者はクライアント、中間層、およびデータベースサーバーのプログラミングの両方で言語を選択できます。ユーザー定義オブジェクト型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の表現を保存できるため、開発がより簡単、迅速、かつ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)
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 ✓

次のページに続く

表 9 – 前のページからの続き

Application development	Oracle Enterprise	EDB Postgres Advanced Server
Hierarchical queries	Yes	Yes ✓
Parallel query	Yes	Yes ✓
PL/SQL supplied packages	Yes	Yes (See *EDB Postgres Advanced Server* と互換性のあるパッケージのサポート)
PRAGMA RESTRICT_REFERENCES	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 ✓

5.3.1 EDB Postgres Advanced Server と互換性のある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
EDB Postgres Advanced Server のドキュメントを参照してください。

Package name	Package description
DBMS_ALERT	Functions that allow asynchronous notification of database events by way of an alert. Using this package and triggers, an application can notify itself whenever values of interest in the database are changed.
DBMS_AQ	Database-integrated asynchronous message queuing provides a flexible mechanism for integrating applications across the enterprise by communicating activities and exchanging a variety of information payloads.
DBMS_AQADM	Provides procedures to create and manage queues and queue tables.
DBMS_CRYPTO	Provides functions to encrypt and decrypt stored data.
DBMS_JOB	Was replaced by DBMS_SCHEDULER but is included for compatibility with older Oracle applications.
DBMS_LOB	Functions that allow access to and manipulation of Large Object values.
DBMS_LOCK	Provides a function interface to Lock Management services.
DBMS_MVIEW	Provides procedures to manage and refresh materialized views and their dependencies.
DBMS_OUTPUT	Allows sending messages from stored procedures, packages, and triggers for application or debugging use.
DBMS_PIPE	Functions that allow two or more sessions in the same database instance to communicate with one another.

次のページに続く

表 10 – 前のページからの続き

Package name	Package description
DBMS_PROFILER	Provides functions to profile stored procedure workloads and identify performance bottlenecks.
DBMS_RANDOM	Useful functions to generate random text, numeric, and date values.
DBMS_REDACT	Redaction prevents a user from seeing all or portions of sensitive data.
DBMS_RLS	Implements row-level security functions in the database, blocking users from seeing each other’ s data in the same application.
DBMS_SCHEDULER	Job scheduler functions for creating and executing unattended repetitive tasks inside the database.
DBMS_SQL	Permits the use of dynamic SQL in procedures to allow applications to run SQL statements with unknown parameters (such as table name) until runtime.
DBMS_SESSION	Functions with the ability to enable and disable roles.
DBMS_UTILITY	A collection of functions for getting information about various runtime operations and metadata from the database.
UTL_ENCODE	Functions to perform Base64 encoding and decoding of data intended for transport between hosts.
UTL_FILE	Allows database procedures to read and write operating system text files in an I/O stream fashion.
UTL_HTTP	Functions that enable you to make HTTP calls to access information on web servers.
UTL_MAIL	Provides functions to create, manage, and send email from the database including attachments, CC, and BCC.
UTL_RAW	Functions supporting manipulating raw data types.
UTL_SMTP	Provides functions for sending email via SMTP according to the RFC821 specification.
UTL_TCP	Provides procedures and functions to enable PL/SQL applications to communicate with external TCP/IP-based servers using TCP/IP.
UTL_URL	Functions for escaping and “unescaping” URL strings.

5.4 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	No

次のページに続く

表 11 – 前のページからの続き

Nonrelational data	Oracle Enterprise	EDB Postgres Advanced Server
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

5.5 Notable differences

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

Oracle Enterprise	EDB Postgres Advanced Server
MERGE	No Postgres UPSERT statements can be used but have different syntax from MERGE. MERGE syntax planned for PostgreSQL 15
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.