PostgreSQL インサイド

データベース移行で押さえておくべきこと

移行プロセスについて

技術を知る

FUJITSU

☐ 導入／環境設定　☑ 移行　☐ 性能　☐ チューニング　☐ バックアップ／リカバリー

☐ 冗長化／負荷分散　☐ 監視　☐ データ連携　☐ 災害対策　☐ 豆知識

近年、運用保守コストの削減や、ベンダーロックインの回避、エコシステムの構築を理由に、商用データベースからオープンソースソフトウェア（以降、OSS と略す）系のデータベースへの移行を検討されるケースが増えています。この記事では、Oracle Database から PostgreSQL へのデータベース移行における、安全な移行プロセスの進め方について説明します。なお、双方のデータベースの基本的な違いについては、「データベース移行で押さえておくべきこと ～アーキテクチャーと機能の違い～」で紹介しています。

1. 移行プロセスの概要

企業や組織にとって重要なデータを扱うデータベースの移行には、はじめに、慎重な検討、および、利点とリスクについての総合的な判断が必要です。データベース移行が決まったら、詳細な計画を立て、その上で、データベースリソースやアプリケーションなどの実際の移行作業を行います。移行後に、パフォーマンス劣化などの不具合が生じて移行工数を増大させたり、コスト追加を発生させたりしないようにするためにも、前工程での、質の高いアセスメントと見積りによる評価が非常に重要になります。

データベースの移行プロセスの概要について説明します。

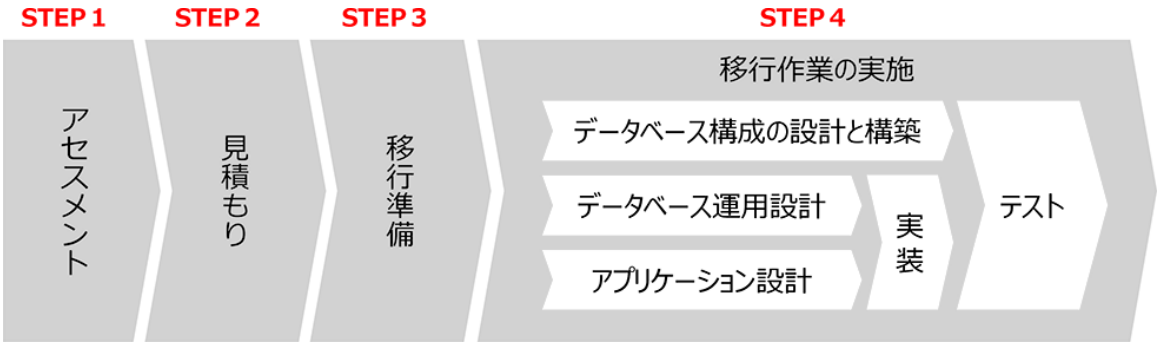

図 1：移行プロセスの概要

1. **STEP1**

   アセスメント

   移行対象となる資産や、データ容量、データベース構成などを調査し、修正が必要な個所を洗い出します。その後、技術的な影響を分析し、移行プロジェクトの作業の難易度を評価します。

2. **STEP2**

   見積り

   このステップでは、アセスメント結果に基づいて、移行プロジェクトのコストとデータ移行にかかる時間を見積もります。併せて、移行時に必要な設備と一時的なライセンス、トレーニングスタッフなどの他のコストも考慮する必要があります。これらの見積もりに基づいて、プロジェクトオーナーは、実現可能性を判断し、この移行プロジェクトを続行するかどうかを決定します。

3. **STEP3**

   移行準備

   見積もったコストを元に、スケジュールや移行作業の体制を検討します。また、移行作業に必要な開発環境、運用環境、および、資産を準備します。

4. **STEP4**

   移行作業の実施

   以下の開発**プロセス**に従って、移行作業を実施します。

   - データベース構成の設計と構築
   - データベースの運用設計
   - アプリケーション設計
   - 実装（プログラミング）
   - テスト

2. 移行プロセスの詳細

以下の移行プロセスについて、順を追って詳細を説明します。

- アセスメント
- 見積り
- 移行準備
- 移行作業の実施

2.1. アセスメント

アセスメントは、技術的な影響を分析し、移行プロジェクトの作業の難易度を評価するための移行プロジェクトの最初のステップです。移行作業の難易度は、さまざまな要因によって大きく異なります。したがって、移行手順で何を処理する必要があるかを理解するために、現行システムをデータベース構成、データ容量、およびデータベーススキーマなどの資産の観点から検討する必要があります。また、移行を完了するための難易度も評価する必要があります。「1. 移行プロセスの概要」で説明したように、移行前の計画段階でのアセスメントは非常に重要です。
データベース移行の際の主な考慮項目の 1 つに、システムの性能への影響があります。移行後のシステム性能が重要な要件である場合は、この段階での性能検証が必要になります。そのためには、いくつかのスキーマとアプリケーションを移行して、システムの性能要件を満たせるかどうかを評価します。
この後のステップでは、エンジニアのスキルと生産性に基づいて移行計画を作成します。そのため、Oracle Database から PostgreSQL への移行を初めて行う場合は、このステップでエンジニアのスキルと生産性を評価することを推奨します。
アセスメントの調査対象と、移行箇所やその難易度を見極めるための主な観点を以下に示します。

表 1 アセスメントの調査対象と主な調査観点

調査対象	主な調査観点
データベース構成	データベース構成の設計および構築の難易度を見極めるため、以下の観点で調査します。 • シングル・インスタンス構成かクラスタ構成か？ • 連携するデータベースがあるか？
SQL	SQL の修正箇所や難易度を見極めるため、以下の観点で調査します。 • データ型 • DDL、DML • トランザクション制御文 • セッション制御文 • システム制御文 • 演算子、条件、式、関数
PL/SQL	PL/SQL の修正箇所や難易度を見極めるため、以下の観点で調査します。 • データ型 • パッケージ • サブプログラム • トランザクションの処理と制御 • GOTO 文
アプリケーション	アプリケーション用インターフェイスの修正の影響を見極めるため、以下の観点で調査します。 • JDBC • ODBC • OCI • Pro*C
データ	データ移行にかかる時間を見極めるため、以下の観点で調査します。 • 表の索引や制約の数 • データ容量 • データ移行方式
運用	データベースの運用に与える技術的な影響を見極めるため、以下の観点で調査します。 • データベース監視 • データベース認証 • バックアップとリカバリー • 高可用性と高信頼性
性能要件	システムの性能要件を確認するため、以下の観点で調査します。 • オンライン処理性能（通常時、高負荷時） • バッチ処理性能

2.2. 見積り

移行プロジェクトの作業コストと、データ移行にかかる時間を見積もります。その後、プロジェクトオーナーが移行の実現可能性を判断し、この移行プロジェクトを続行するかどうか決定します。見積もり方法は次の通りです。

移行作業コスト

アセスメントで洗い出した移行が必要な個所や難易度から、設計、修正、および、テストにかかる作業量を割り出します。割り出した作業量と修正難易度、エンジニアのスキルや生産性などから、移行に必要な作業コストを見積もります。
テスト工数を見積もる際には、データベース移行時の特有な問題に備えて、十分に時間的なバッファを確保することをお勧めし

ます。なお、データベースを移行する際にデータベースの詳細な仕様差を調査しますが、それでもテスト工程で動作させてみるまでは、その仕様差に気づけない場合があります。例えば、以下のようなケースに遭遇する可能性があります。

- 演算時の切り捨てや切り上げの仕様差などが原因で、SQL で実行した数値演算結果が異なる
- データ型の精度や表示形式の違いにより、SQL で抽出した日時と時刻の出力結果が異なる

データ移行時間

データ移行時間も併せて見積もりを行う必要があります。現行システムのデータ量が大規模な場合は、データ移行にも時間がかかります。また、同じデータ量でも選択した移行方式や環境によって移行にかかる時間は異なります。したがって、テスト環境を構築してデータ移行のリハーサルを実施し、リハーサルの結果を元に本番環境のデータ移行時間を見積もる必要があります。

2.3. 移行準備

前ステップで見積もった結果を元に、移行作業のスケジュールを作成し、開発体制を構築します。また、移行作業に必要となる環境や資産を準備します。

移行作業スケジュールの作成

見積結果に基づいて、プロジェクトのスケジュールと開発体制を含む移行計画を作成します。現行システムから移行先システムへの切り替えに必要なダウンタイムなど、稼働中の計画も必要です。

準備

移行作業においては、本番用の運用環境と開発環境の両方を使用します。それぞれの環境において、現行システムと移行先システムの 2 つの環境を準備します。

- 本番用の運用環境
  現在稼働中の現行システム、および、移行後に使用する移行先システムのための環境です。
- 開発環境
  設計、コーディング、テストなどの移行作業を行う環境です。開発環境にも、現行システムと移行先システムの 2 つの環境が必要です。現行システムで利用しているハードウェアやソフトウェアの構成と同等のものを、移行先システムに準備します。現行システムの対象資産を移行するためにツールを使用する場合は、そのツールを使用するための環境設定も適宜実施します。

資産

本番環境の現行システムの資産（アプリケーション、運用に関するバッチ処理、データなど）を開発環境の現行システムにも準備します。ただし、データベースの移行を実施する上で影響のない部分については、適宜簡略化することが可能です。例えば、開発環境のデータベースに格納するデータには、本番環境のサンプルデータを利用することもあれば、本番環境と同様の特徴を持つテストデータを利用することもあります。

2.4. 移行作業の実施

ここからは、すべての資産を Oracle Database から PostgreSQL に移行するための、実際の作業のステップになります。なお、このステップは、通常のシステム開発のプロセスと同等の作業になります。

2.4.1 データベース構成の設計と構築

多くの場合、移行先システムのデータベース構成は、システム要件を満たし、現行システムのデータベースと同等のシステムを提供するよう設計します。特に、設計時には Oracle Database と PostgreSQL のアーキテクチャーの違いを考慮する必要があります。次の表は、2 種類の一般的なデータベース構成について説明しています。特に高可用構成の場合は注意が必要です。なお、参

考として、OSS の PostgreSQL をエンタープライズ向けに強化した製品「FUJITSU Software Enterprise Postgres（以降、Enterprise Postgres と略す）」の情報も併せて示します。

表 2 一般的なデータベース構成

データベース構成	説明
シングル・インスタンス構成	1 つのサーバーで 1 つのデータベースを管理する構成です。 特に検討すべきことはありません。
クラスタ構成（高可用構成）	最小限のダウンタイムで信頼性を提供するための、複数のサーバーを組み合わせた構成です。 データベースのクラスタ化については、Oracle Database は Real Application Clusters などで実現しますが、PostgreSQL は Pgpool-II などで対応が可能です。 **【参考】** Enterprise Postgres は障害時に高速な切り替えが可能なデータベース多重化機能が利用できます。

2.4.2 データベース運用設計

移行先システムのデータベースの運用について設計します。データベースの運用はデータベースごとに異なるため、PostgreSQL のアーキテクチャーや機能に準じて、新たに設計する必要があります。運用設計には、主に以下のような観点が必要です。

バックアップとリカバリー

PostgreSQL は、論理バックアップ、物理バックアップ、また、継続的アーカイブを用いたオンラインバックアップなどに対応しています。さらに、周辺ツールの pg_rman を利用したバックアップとリカバリーの管理も行えます。PostgreSQL のバックアップとリカバリーについては、以下の記事で紹介していますので併せてご覧ください。

- 技術を知る：PostgreSQL のバックアップとリカバリー
- PostgreSQL の周辺ツール ～ pg_rman でバックアップ・リカバリーを管理する ～

データベースの監視

データベースの監視は、運用中のデータベースにおける突然の停止、性能劣化、ディスク領域不足などの事象への迅速な対応や、トラブルの未然防止などを目的に設計します。その際、データベースのアーキテクチャーに沿って、様々な観点で、常時監視あるいは定期監視を行う必要があります。PostgreSQL のデータベース監視については、以下の記事で紹介していますので併せてご覧ください。

- データベースシステムの監視 ～監視の概要～

監査ログ

PostgreSQL では、拡張モジュールの pgaudit が利用できます。なお、Enterprise Postgres では、実際の業務利用を想定し、SQL 文の実行結果情報を取得するなどの機能強化が行われています。以下の記事で紹介していますので併せてご覧ください。

- PostgreSQL の監査ログ ～セキュリティ対策は万全! 監査ログで情報漏えいを検知～ －富士通の技術者に聞く！ PostgreSQL の技術－

メンテナンス（チューニング）

PostgreSQL では、データベースのパフォーマンスを診断するために、統計情報ビューの pg_stat_statements を参照する方法やサーバーログを利用する方法があります。また、周辺ツールの pg_statsinfo や pgBadger などを活用する方法もあります。検出されたパフォーマンス上の問題に対し、総合的な観点で原因を特定して対処する方針を決めます。PostgreSQL のメンテナンス（チューニング）については、以下の記事で紹介していますので併せてご覧ください。

- チューニング ～ SQL チューニングの概要 ～
- pg_statsinfo で統計情報を収集・蓄積する
- pgBadger でログファイルを解析し、統計レポートを作成する
- パフォーマンスチューニング 9 つの技 ～はじめに～

高可用性

PostgreSQL では、主にストリーミングレプリケーションをベースとした冗長化機能を用いますが、運用においては、平常時だけでなく高負荷時や異常時のシステム要件を満たすよう設計します。そのため、監視、異常検知、アプリケーションからの透過的接続などの考慮を含めた設計が必要になります。参考として、Enterprise Postgres の高可用の考え方や対策を以下の記事で紹介していますので併せてご覧ください。

- 業務停止はさせない!トラブル時は自動切替えで PostgreSQL の運用を継続 –富士通の技術者に聞く！PostgreSQL の技術–
- PostgreSQL の高可用性を追求！Connection Manager でデータベースへの接続を瞬時に切り替える –富士通の技術者に聞く！PostgreSQL の技術–

データベース認証

Oracle と PostgreSQL の認証方式や設定方法が異なります。詳細については、PostgreSQL のマニュアルを参照してください。なお、マニュアルの章番号はバージョンによって変わることがありますのでご注意ください。

- Documentation（PostgreSQL オフィシャル）
  https://www.postgresql.org/docs/
    - III. Server Administration
        - 21. Client Authentication
- PostgreSQL 日本語ドキュメント（日本 PostgreSQL ユーザ会）
  https://www.postgresql.jp/document/
    - III. サーバの管理
        - 20. クライアント認証

データベースのバージョンアップ

PostgreSQL は、バグ修正やセキュリティ対応のマイナーバージョンアップと、機能追加や性能改善を伴うメジャーバージョンアップがあります。周辺ツールの pg_upgrade によるデータ移行が可能です。また、同期レプリケーション構成の場合は、ローリングアップデートが可能です。詳細については、PostgreSQL のマニュアルを参照してください。なお、マニュアルの章番号はバージョンによって変わることがありますのでご注意ください。

- Documentation（PostgreSQL オフィシャル）
    - III. Server Administration
        - 19. Server Setup and Operation
            - 19.6. Upgrading a PostgreSQL Cluster

- PostgreSQL 日本語ドキュメント（日本 PostgreSQL ユーザ会）
    - III. サーバの管理
        - 18. サーバの準備と運用
            - 18.6. PostgreSQL クラスタのアップグレード処理

2.4.3 アプリケーション設計

Oracle Database の SQL やアプリケーションなどを移行先の PostgreSQL で実行するための移行方式を設計します。ここでは、SQL、PL/SQL、アプリケーション、バッチ処理、および、データの資産移行の際に注意すべき点を説明します。

SQL

以下の分類で、Oracle Database の SQL を PostgreSQL に移行する際の注意点を示します。

- データ型
- DDL 文
- DML 文
- ファンクション
- その他の SQL 文

表 3 データ型の移行に関する注意点

構文	説明
文字データ	PostgreSQL では、CHAR などの一部の文字型はサポートされていますが、最大サイズの指定方法に違いがあります。 VARCHAR2、NVARCHAR2、CLOB、NCLOB、LONG などの一部の文字型はサポートされていないため、TEXT 型など PostgreSQL の類似の文字型の利用、あるいは、代替方法の検討が必要です。なお、orafce 拡張モジュールを利用することで、VARCHAR2、NVARCHAR2 が利用できます。その際にも、最大サイズの指定方法に違いがあるため検討が必要です。 **【参考】** Enterprise Postgres は Orafce（Oracle Database と互換性がある関数やデータ型を提供する OSS）を同梱し、標準サポートしています。
数値データ	Oracle Database と PostgreSQL で違いがあるため修正が必要です。有効桁数や切り捨てなどの違いにも注意が必要です。
日時データ	Oracle Database と PostgreSQL で同じ型が使用できます。ただし、精度やタイムゾーン指定、表示形式に違いがあるため注意が必要です。
特殊なデータ	BLOB データ型などのラージ・オブジェクト（LOB）は、PostgreSQL の対応する型に変更します。XML データ型や JSON データ型は、どちらのデータベースでも使用できますが、機能に違いがあるため修正が必要です。その他の型については、対応する型が PostgreSQL にありません。使用方法を踏まえてどの型に移行するか検討してください。
ROWID データ型	ROWID 型、UROWID 型と同等な型は PostgreSQL にありません。serial 型や順序（シーケンス）で代替するなど、行を特定する方法を検討してください。
SQL 演算子とファンクションのメタデータ	同等の型は PostgreSQL にはありません。使用方法を踏まえてどの型に移行するか検討してください。

表 4 DDL 文の移行に関する注意点

構文	説明
スキーマ	Oracle Database の CREATE SCHEMA 文では、実際にスキーマは作成されません。一方、PostgreSQL の CREATE SCHEMA 文は新しいスキーマを作成します。指定したスキーマ名と同じ名前のスキーマがすでに存在する場合、PostgreSQL の CREATE SCHEMA 文はエラーを返すため注意が必要です。
データベース・リンク	Oracle Database はデータベース・リンクの作成機能があります。PostgreSQL で外部の表への操作を行う場合、ユーザーは外部データラッパを使用します。外部データラッパは PostgreSQL や Oracle Database などとの接続が可能であり、CREATE EXTENSION コマンドでデータベースへ登録を行うことで使用できます。
データベース・トリガー	PostgreSQL では、Oracle Database と同様にトリガーが使用可能です。ただし、Oracle Database と PostgreSQL では以下の点に違いがあるため注意が必要です。 • トリガーの記述方法、オプション • トリガーを記述する言語 • トリガーを起動可能なイベント
索引（インデックス）	Oracle Database には、B ツリー索引、ビットマップ索引、ファンクション索引、アプリケーション・ドメイン索引があります。PostgreSQL ではこれらのうち、B ツリー索引が使用可能です。B ツリー索引以外を使用している場合、PostgreSQL がサポートする索引に変更するか、索引を削除する必要があります。また、索引キーの長さや、索引キーに指定できるデータ型に違いがありますので注意してください。
マテリアライズド・ビュー	PostgreSQL でも Oracle Database と同様にマテリアライズド・ビューが使用可能です。しかし、PostgreSQL のマテリアライズド・ビューは、Oracle Database のマテリアライズド・ビューとは機能やリフレッシュ方法、構文などに違いがあるため注意が必要です。
ユーザー定義の演算子（オペレーター）	PostgreSQL でも Oracle Database と同様に新しい演算子をユーザーが定義することが可能です。ただし、構文に違いがあるため注意が必要です。
順序（シーケンス）	PostgreSQL でも Oracle Database と同様に順序が使用可能です。ただし、構文に違いがあるため注意が必要です。
ストアド・ファンクション、ストアド・プロシージャ	PostgreSQL でも Oracle Database と同様にストアド・ファンクションおよびストアド・プロシージャが使用可能です。ただし、構文や使用可能な言語に違いがあるため注意が必要です。
リレーショナル表（表、テーブル）	PostgreSQL でも、リレーショナル表相当の表は使用可能です。ただし、指定可能なデータ型や構文に違いがあるため注意が必要です。また、Oracle Database と同様に PostgreSQL でも表のパーティションの定義が可能です。ただし、Oracle Database とは指定できる分割の種類、機能、構文などに違いがあるため注意が必要です。
ビュー	PostgreSQL でも Oracle Database と同様にビューが使用可能です。ただし、PostgreSQL のビューは、WITH READ ONLY オプション相当の機能がありません。また構文にも違いがあるため注意が必要です。
ロール	PostgreSQL では、Oracle Database と同様にロールが使用可能です。ただし、構文に違いがあるため注意が必要です。

構文	説明
ユーザー	Oracle Database とは違い、PostgreSQL のユーザーはロールの一種です。この差異により、ユーザーを作成する SQL 文などにも違いがあるため注意する必要があります。また、PostgreSQL では、ユーザーの認証管理は設定ファイルの pg_hba.conf で行います。
その他の DDL 文	以下の機能は PostgreSQL にはありません。代替方法の検討を行う必要があります。 • クラスタ • 索引構成表 • オブジェクト表 • パッケージ • シノニム

表 5 DML 文の移行に関する注意点

構文	説明
SELECT / INSERT / UPDATE / DELETE	PostgreSQL でも、基本的な構文は Oracle Database と同じです。ただし、細かな構文には違いがあるため注意が必要です。
MERGE	PostgreSQL には MERGE 文はありません（注 1）。しかし、INSERT 文の ON CONFLICT 句を使用することで同様の処理（UPSERT）を実現することができます。
CALL	PostgreSQL にも CALL 文は存在します。ただし、構文は Oracle Database とは違いがあるため注意が必要です。
EXPLAIN PLAN	Oracle Database の EXPLAIN PLAN 文は、PostgreSQL の EXPLAIN 文に相当します。ただし、実行結果が表に格納されないこと、および、構文に違いがありますので、使用する場合に注意が必要です。
LOCK TABLE	Oracle Database の LOCK TABLE 文は、PostgreSQL では LOCK 文と呼ばれます。ただし、一部のロックモードの名前や構文に違いがあるため注意が必要です。

- （注 1） PostgreSQL 15 から使用可能となりました。

表 6 ファンクションの移行に関する注意点

構文		説明
単一行ファンクション	数値ファンクション	数値ファンクションは PostgreSQL でも一部を除きそのまま使用できます。ただし、パラメーターの違いや戻り値の精度が違う場合があるため注意が必要です。なお、以下のファンクションは別のファンクションへの書き換えが必要です。 • BITAND • REMAINDER
	文字値を戻す文字ファンクション	文字値を戻す文字ファンクションは PostgreSQL でも一部を除きそのまま使用できます。ただし、パラメーターが違う場合があるため注意が必要です。なお、以下のファンクションは別のファンクションへの書き換えが必要です。 • NCHR • NLS_INITCAP • NLS_LOWER • NLS_UPPER • SOUNDEX • TRANSLATE ... USING

構文		説明
	数値を戻す文字ファンクション	数値を戻す文字ファンクションは PostgreSQL でも一部を除きそのまま使用できます。
	日時ファンクション	日時ファンクションは PostgreSQL でも一部を除きそのまま使用できます。ただし、精度やタイムゾーンに対応していないなど、違う場合があるため注意が必要です。
	一般的な比較ファンクション	一般的な比較ファンクションは PostgreSQL でもそのまま使用できます。
	変換ファンクション	変換ファンクションは PostgreSQL でも一部を除きそのまま使用できます。ただし、精度やパラメーターが違う場合があるため注意が必要です。
	収集ファンクション	収集ファンクションに相当するファンクションは、PostgreSQL にはありません。なお、CARDINALITY は SQL で同等の情報を取得できますが、その他の収集ファンクションについては代替方法を検討する必要があります。
	XML ファンクション	XML ファンクションは PostgreSQL でも一部を除きそのまま使用できます。ただし、パラメーターに違いがあるため注意が必要です。
	JSON ファンクション	JSON ファンクションは PostgreSQL にも存在しますが、Oracle Database のファンクションとは異なります。そのため、処理内容に合わせて移行方法を検討する必要があります。
	エンコーディング・ファンクションおよびデコーディング・ファンクション	エンコーディング・ファンクションおよびデコーディング・ファンクションのうち、DECODE は PostgreSQL にも存在しますが、Oracle Database のファンクションとは異なります。そのため、移行方法を検討する必要があります。その他については代替方法を検討する必要があります。
	NULL 関連ファンクション	NULL 関連ファンクションは PostgreSQL でも一部を除きそのまま使用できます。
	環境ファンクションおよび識別子ファンクション	環境ファンクションおよび識別子ファンクションのうち、USER は PostgreSQL にある同等の関数（SESSION_USER）に修正することで対応できます。その他については代替方法を検討する必要があります。
	その他の単一行ファンクション	以下の単一行ファンクションに相当するファンクションは、PostgreSQL にはありません。代替方法を検討する必要があります。 • キャラクタ・セット・ファンクション • 照合ファンクション • ラージ・オブジェクト・ファンクション • 階層ファンクション • データ・マイニング・ファンクション
集計ファンクション		集計ファンクションは、一部を除き PostgreSQL でもそのまま使用可能です。ただし、戻り値やオプションなどが違う場合があるため注意が必要です。
分析ファンクション		分析ファンクションは、一部を除き PostgreSQL でもそのまま使用可能です。ただし、戻り値やオプションなどが違う場合があるため注意が必要です。
オブジェクト参照ファンクション		オブジェクト参照ファンクションに相当するファンクションは、PostgreSQL にはありません。代替方法を検討する必要があります。

構文	説明
モデル・ファンクション	モデル・ファンクションに相当するファンクションは、PostgreSQL にはありません。代替方法を検討する必要があります。
OLAP ファンクション	OLAP ファンクションに相当するファンクションは、PostgreSQL にはありません。代替方法を検討する必要があります。
データ・カートリッジ・ファンクション	データ・カートリッジ・ファンクションに相当するファンクションは、PostgreSQL にはありません。代替方法を検討する必要があります。
ユーザー定義ファンクション	PostgreSQL にもユーザー定義ファンクションはあります。しかし、ユーザー定義ファンクションの処理内容を PostgreSQL で実行できるように構文などを修正する必要があります。

表 7 その他 SQL 文の移行に関する注意点

構文		説明
トランザクション制御文	COMMIT	PostgreSQL でも Oracle Database と同様に COMMIT 文が使用できます。ただし、構文に違いがあるため注意が必要です。
	ROLLBACK	PostgreSQL でも Oracle Database と同様に ROLLBACK 文が使用できます。ただし、構文に違いがあるため注意が必要です。また、TO SAVEPOINT 句を指定している場合、PostgreSQL では ROLLBACK TO SAVEPOINT 文を使用します。
	SAVEPOINT	PostgreSQL でも Oracle Database と同様に SAVEPOINT 文が使用できます。ただし、同じ名前のセーブポイントを作成した場合の動作に違いがあるため注意が必要です。
	SET TRANSACTION	PostgreSQL にも SET TRANSACTION 文はあります。ただし、PostgreSQL の SET TRANSACTION 文は、Oracle Database で ISOLATION LEVEL 句を指定した場合の機能です。また、構文に違いがあるため注意が必要です。USE ROLLBACK SEGMENT 句などの他の句を使用している場合、移行方法の検討が必要です。
	SET CONSTRAINT	PostgreSQL でも Oracle Database と同様に SET CONSTRAINT 文の機能が使用できます。ただし、PostgreSQL では SET CONSTRAINTS 文に変える必要があります。
セッション制御文	ALTER SESSION	PostgreSQL には、Oracle Database の ALTER SESSION 文に相当する文はありません。代替方法を検討する必要があります。
	SET ROLE	Oracle Database と PostgreSQL では、SET ROLE 文の処理が異なります。Oracle Database の SET ROLE 文は、そのセッションに対してロールを使用可能または使用禁止にします。PostgreSQL の SET ROLE 文は、現在のセッションにおけるユーザー識別子を変更します。そのため、代替方法を検討する必要があります。
システム制御文		PostgreSQL にも ALTER SYSTEM 文がありますが、Oracle Database と PostgreSQL ではデータベースのアーキテクチャーが違うため、PostgreSQL に合わせた方法に変更する必要があります。

構文		説明
演算子		多くの演算子は PostgreSQL でもそのまま使用できます。しかし、以下の演算子は修正が必要です。 • 階層問合せ演算子(PRIOR、CONNECT_BY_ROOT) • MINUS • MULTISET 演算子(MULTISET、MULTISET EXCEPT、MULTISET INTERSECT、MULTISET UNION) また、連結演算子の「||」は、連結する文字列の中に NULL が含まれている場合の動作に差異があるため、注意が必要です。
式		多くの式は PostgreSQL でもそのまま使用できます。しかし、以下の式は修正が必要です。 • CURSOR 式 • モデル式 • オブジェクト・アクセス式 • プレースホルダ式 • 型コンストラクタ式 また、日時式や期間式などは構文や結果の形式などに違いがあるため注意が必要です。
条件		多くの条件は PostgreSQL でもそのまま使用できます。しかし、以下の条件は修正が必要です。 • ^=（不当性のテスト） • 浮動小数点条件（IS [NOT] NAN、IS [NOT] INFINITE） • モデル条件（IS ANY 条件、IS PRESENT 条件） • 多重集合条件（IS A SET 条件、IS EMPTY 条件、MEMBER 条件、SUBMULTISET 条件） • REGEXP_LIKE 条件 • XML 条件（EQUALS_PATH 条件、UNDER_PATH 条件） • IS OF type 条件 また、PostgreSQL では、ANY、SOME、ALL に式リストが使用できません。
その他の注意点	識別子	Oracle Database と PostgreSQL では、非引用識別子の大文字小文字の扱いが異なります。PostgreSQL では非引用識別子は小文字として解釈されるため、アプリケーションなどで大文字小文字を区別する場合は修正が必要です。
	暗黙的なデータ変換	PostgreSQL でも暗黙的なデータ変換は使用できます。しかし、Oracle Database と比較して変換できる範囲が小さいため、明示的なデータ変換を行うなどの修正が必要となる場合があります。
	長さが 0 の文字値	Oracle Database では、長さが 0 の文字値を NULL として扱います。一方、PostgreSQL では長さが 0 の文字値と NULL は別の値として扱います。長さが 0 の文字値を NULL と判定することを前提としている場合、データやアプリケーションなどの修正が必要です。
	照合順（ソート順）	Oracle Database と PostgreSQL では、照合順が全てのパターンにおいて同じ結果にならない場合があります。その場合は、業務影響が少ないパターンで近似するものを選ぶ必要があります。照合順序には、各国語を合わせると複数ありますが、日本語を含む場合は以下の 2 種類のみとなります。 • UTF8 文字で辞書順

構文		説明
		• C ロケールによる、UTF8 文字コードのバイナリー順

PL/SQL

以下に、Oracle Database の PL/SQL を PostgreSQL の PL/pgSQL に移行する際の注意点を示します。

表 8 PL/SQL の移行に関する注意点

構文		説明
基本構文	ブロック	ブロック、エラー処理、変数などの基本的な構文要素は、PostgreSQL でも使用できます。
	エラー処理	
	変数	
データ型	スカラー・データ型	SQL データ型については、「表 3 データ型の移行に関する注意点」を参照してください。BOOLEAN 型は PostgreSQL でもそのまま使用できます。その他の型については、代替方法を検討する必要があります。
	コンポジット・データ型	コレクション型に相当する型は PostgreSQL にはありません。一時表などで代替します。レコード変数は PostgreSQL でも使用できますが、細かな仕様で違いがあるため注意が必要です。
制御文	条件文	IF 文、CASE 文は PostgreSQL でも使用できます。
	LOOP 文	基本 LOOP 文、WHILE LOOP 文、FOR LOOP 文は PostgreSQL でも使用できます。ただし、FOR LOOP 文の REVERSE 句には仕様差があるため注意が必要です。
	GOTO 文	GOTO 文は PostgreSQL では使用できません。GOTO 文を使わないよう検討する必要があります。
静的 SQL	カーソル	カーソルは PostgreSQL でも使用できますが、細かい仕様差があるため注意が必要です。
	トランザクションの処理および制御	COMMIT 文、ROLLBACK 文は PostgreSQL でも使用できますが、トランザクション制御の仕様に違いがあるため注意が必要です。例えば、エラー処理(EXCEPTION 句)を含むブロックでは、COMMIT 文、ROLLBACK 文がエラーになります。
動的 SQL	EXECUTE IMMEDIATE 文	EXECUTE IMMEDIATE 文は PostgreSQL の EXECUTE 文に変更が必要です。
	OPEN FOR 文	OPEN FOR 文は PostgreSQL でも使用できますが、細かい仕様差があるため注意が必要です。
サブプログラム（ファンクション内ファンクションなど）		サブプログラムは PostgreSQL で使用できないため、別のプロシージャやファンクションなどに変更する必要があります。また、共通変数も使用できないため、代替方法を検討する必要があります。
トリガー		トリガーは PostgreSQL でも使用できますが、細かい仕様差があるため注意が必要です。

構文	説明
パッケージ	パッケージは PostgreSQL で使用できないため、プロシージャやファンクションなどに変更する必要があります。また、パッケージ内のサブプログラムやパッケージ変数も使用できないため、代替方法を検討する必要があります。
Oracle 社が提供する PL/SQL パッケージ	代替方法を検討する必要があります。なお、DBMS_OUTPUT、UTL_FILE などの一部パッケージは、代替機能として orafce 拡張モジュールが使用できます。ただし、細かい仕様差があるため注意が必要です。

アプリケーション

Oracle Database のアプリケーション用インターフェイスの中には、PostgreSQL で利用できないものがあります。また同じインターフェイスが利用できる場合でも、API の仕様が異なる場合があります。そのような場合には、代替方法を検討する必要があります。PostgreSQL では、標準で利用可能なインターフェイスとして、C 言語用ライブラリー（libpq）や C 言語による埋め込み SQL（ECPG）があり、さらに、JDBC ドライバー（Type4）、ODBC ドライバーなどの PostgreSQL 専用の各種言語用のインターフェイスも利用できます。その利用については、仕様の確認が必要です。

運用に関するバッチ処理

運用に関するバッチ処理（スクリプトファイルなど）に記述されているデータベース運用コマンドは、Oracle Database と PostgreSQL で大きく仕様が異なります。各コマンドに対して代替処理を検討する必要があります。

データ

データの移行は、基本的には以下の手順で行います。

1. 現行システムのデータベースからデータを抽出し、ファイルに格納する
2. 手順 1.のファイルのデータ形式を移行先システムのデータベースの形式に変換する
3. 手順 2.のファイルを移行先システムに格納する

データ移行の設計においては、次の点を考慮してください。

- 手順 1.において、ファイル形式に CSV ファイルを利用することを推奨します。CSV ファイルは、手順 3.において PostgreSQL にデータを格納する時に COPY 文を利用することができます。COPY 文はデータを効率良く格納することができます。
- 対象データが Oracle Database 固有のデータ型、または、対象データに外字が含まれている場合、手順 1.のデータの抽出方法から検討する必要があります。
- 手順 2.においてデータ形式を変換する際、ヘッダー行の有無や NULL 値の表現方法について注意する必要があります。

2.4.4 実装（プログラミング）

「2.4.2 データベース運用設計」、および、「2.4.3 アプリケーション設計」で決定された移行方法に従って、現行システムの資産を修正します。これらの資産には、データベースを操作するための SQL、PL/SQL、アプリケーション、および、運用に関するバッチ処理が含まれます。
また、データについても、設計プロセスで決定された方法で移行を行います。

2.4.5 テスト

移行先システムでの実装が完了したら、システムが期待どおりに動作することを確認するために動作検証を行います。また、データベースが変更されたことによる影響を判断し、システムの機能要件と非機能要件を満たしているかを評価します。評価のために次のテストを実施することをお勧めします。

スキーマの検証（プログラムテスト）

表、索引、ストアドファンクション、ストアドプロシージャなどの現行システム上の資産が、すべて移行先システムに移行されたことを確認します。
確認方法の例としては、Oracle Database のシステム表とビューからデータベースオブジェクト定義を取得し、同様に、PostgreSQL のシステムカタログとシステムビューからデータベースオブジェクト定義を取得します。これら取得した定義を比較して、必要なすべてのオブジェクトが移行されたことを確認します。

データ検証

データが現行システムのデータベースから移行先システムのデータベースへ正常に移行されたことを確認します。
確認方法の例としては、双方のデータベースの表データを CSV などの同じ形式のファイルに出力します。これらのファイルを比較して、移行前と移行後のデータの間に違いがないことを確認します。

アプリケーション機能テスト

移行先システムのアプリケーションが機能要件を満たしていることを確認します。機能要件に基づいてテストケースシナリオを作成し、すべてのシナリオについてのテストを完了させて、結果が期待どおりかどうかを確認します。

パフォーマンステスト

移行先システムがパフォーマンス要件を満たしていることを確認します。パフォーマンス要件に基づいてテストケースシナリオを作成し、すべてのシナリオについてのテストを完了させて、結果が期待どおりかどうかを確認します。
移行先システムがパフォーマンス要件を満たしていない場合は、原因を分析し、問題を解決します。パフォーマンスの問題を解決するためには、データベースのパラメーターを最適化したり、SQL 文を適切に変更したりします。具体的には以下の記事を参考にしてください。

- データベースのパラメーターを最適化するための手法についての解説
    - チューニング ～ データベースチューニング ～
- SQL 文を適切に変更するための手法についての解説
    - チューニング ～ SQL チューニングの概要 ～

運用テスト

移行先システムが期待どおりに業務を運用できることを確認します。以下のような観点でテストケースシナリオを作成し、すべてのシナリオについてのテストを完了し、結果が期待どおりであるかどうかを確認します。

- データベースの開始と停止を確認します。
- データのバックアップの取得と管理について確認します。
- VACUUM や索引（インデックス）再構築を行うことで、ディスクの使用状況の確認と領域割り当てについて確認します。
- 監査ログの出力情報を確認します。
- データベースへの接続の状態を確認します。たとえば、長時間接続されたままになっていないか、特定の処理がリソースを占有していないかを確認して、パフォーマンスの低下を防止します。
- パッチ適用について確認します。
- 高可用性環境における保守のため、ノードの切り替え、切断、フェイルバックなどを確認します。
- データベースの監視が正しく行われることを確認します。

回復テスト

移行先システムに異常系のトラブルが発生した場合でも、業務を回復し継続できることを確認します。以下のような異常系のテストケースシナリオを作成し、すべてのシナリオについてのテストを完了し、業務をすぐに再開できるかどうかを確認します。

- ディスクやネットワーク機器などのハードウェア障害から回復します。
- バックアップからデータ復旧します。
- アプリケーションの異常系の動作を確認します。たとえば、リソースを占有する接続がある場合は、その接続を切断して待機状態になっていた他の接続待ちを解消します。
- ディスク容量が不足した場合の処理を確認します。
- 高可用性環境における、フェイルオーバーの処理を確認します。

Oracle Database から PostgreSQL への移行を前提として、安全な移行プロセスの進め方について述べました。移行すべき資産ごとの移行の方向性も示しましたので、実際に移行作業を行う際の指標としてご利用頂けると幸いです。

参考

Enterprise Postgres は、OSS の PostgreSQL を拡張し、エンタープライズ利用に向けて信頼性、性能、セキュリティを強化したデータベースであり、ワンストップの長期サポートにも対応しています。また、他社データベースからの移行支援サービスとして、アプリケーションの移行アセスメントから資産移行、導入までのトータルなサポートを提供しています。

2023 年 5 月 8 日