

PL/SQL のストアド・パッケージを PostgreSQL に移行する 移行概要/移行例 技術を知る

- | | | | | |
|-----------------------------------|--|--------------------------------|---------------------------------|---------------------------------------|
| <input type="checkbox"/> 導入／環境設定 | <input checked="" type="checkbox"/> 移行 | <input type="checkbox"/> 性能 | <input type="checkbox"/> チューニング | <input type="checkbox"/> バックアップ／リカバリー |
| <input type="checkbox"/> 冗長化／負荷分散 | <input type="checkbox"/> 監視 | <input type="checkbox"/> データ連携 | <input type="checkbox"/> 災害対策 | <input type="checkbox"/> 豆知識 |

Oracle Database から PostgreSQL への移行の際、移行対象となる Oracle Database の資産の中でも比率が高いのは、Oracle Database の手続き型言語である PL/SQL の資産です。PL/SQL のアプリケーションを PostgreSQL の手続き型言語である PL/pgSQL に移行するためには、それぞれが独自拡張した仕様の違いを吸収する必要があり、これが移行工数を増大させる原因の一つになっています。中でも PL/SQL のストアド・パッケージは PostgreSQL 本体機能に該当するものがなく、移行するには段階的に PostgreSQL の機能に置き換える必要があるため、手順が複雑です。

この記事では、ストアド・パッケージの PostgreSQL への移行を安全かつスムーズに実施することを目的に、移行の概要やポイントについてサンプルを交えて解説します。ストアド・パッケージ以外の PL/SQL 資産の移行概要やポイントについて知りたい場合は「PL/SQL を PostgreSQL に移行する ～移行概要／移行例～」で解説しています。考慮すべき非互換の観点なども説明していますので、まだ読まれていない方は、是非先にお読みください。

なお、対象となる Oracle Database のバージョンは 10g 以降、PostgreSQL のバージョンは 11 以降を想定しています。

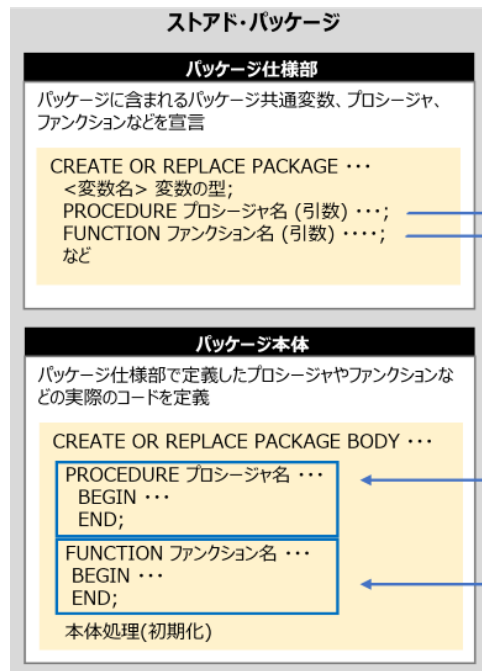
また、本記事をより理解して頂くための助けとなる解説を、以下に掲載しています。是非併せてお読みください。

- Oracle Database と PostgreSQL のアーキテクチャの違いや、移行プロセスを知る
- 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～
- 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～

1. ストアド・パッケージとは

Oracle Database のストアド・パッケージは、パッケージ共通変数、複数のストアド・プロシージャやストアド・ファンクションを 1 つのパッケージにしたオブジェクトです。解析済の状態ですべてに格納され、記述した処理はサーバー上で実行されます。PostgreSQL にはストアド・パッケージに相当する機能はありません。

ストアド・パッケージはパッケージ仕様部とパッケージ本体部で構成されます。構成の概要を下図に示します。



参考

Oracle 社からは、作成済みの数多くの PL/SQL パッケージが標準で提供されています。また 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）」では、PL/SQL パッケージの中でもよく利用されている以下のものを、パッケージとして提供しています。

DBMS_ALERT、DBMS_ASSERT、DBMS_OUTPUT、DBMS_PIPE、DBMS_RANDOM、DBMS_UTILITY、UTL_FILE、DBMS_SQL

なお、PL/SQL パッケージとは仕様の差異があるため、利用の際にはマニュアルを確認してください。

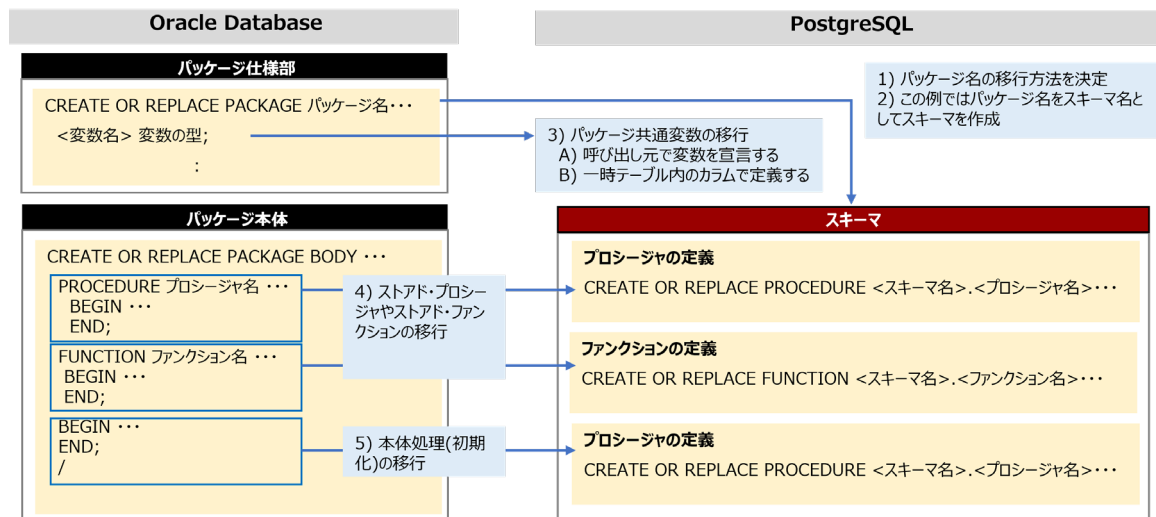
以降では、ストアド・パッケージの移行方法や移行例を説明します。なお、この記事で説明するストアド・パッケージについては、標準で提供されている PL/SQL パッケージを対象としていません。

2. ストアド・パッケージの移行方法

移行の際には、ストアド・パッケージ内のプロシージャやファンクションを、個々のストアドプロシージャやストアドファンクションに分解する必要があります。また、ストアド・パッケージ内にある、パッケージ共通変数については、主に以下の2種類の移行方法があります。

- A) ストアド・パッケージの呼び出し元で変数を宣言して保持し、プロシージャやファンクションのパラメーターで受け渡す
- B) パッケージ共通変数を一時テーブルで保持し、共通変数にアクセスする際には一時テーブルにアクセスするよう変更する

A)については、プロシージャやファンクションのインターフェイスの変更などが発生するため、移行時に作業コストが発生します。B)については共通変数をアクセスする箇所のみでの修正で済みますが、一時テーブルへのアクセスが増えると性能上のボトルネックになる危険性があります。そのため、実際のストアド・パッケージの作りや、共通変数へのアクセス頻度を考慮し、どちらの移行方法を選択すべきかを検討する必要があります。



以下に、移行の手順を示します。

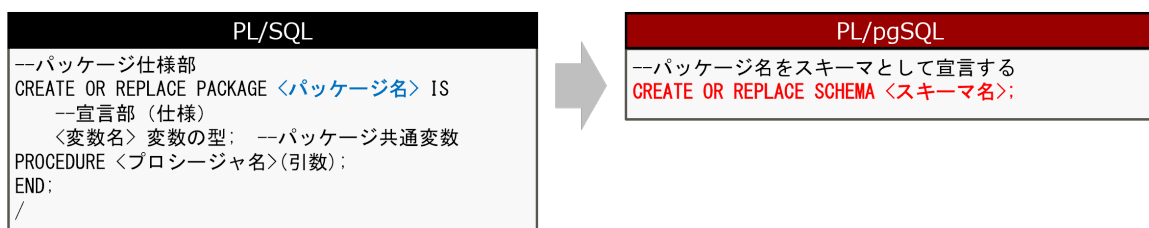
パッケージ仕様部の移行

1) パッケージ名の移行方法を決めます。

- 対処 1) パッケージに対応したスキーマ名を新たに作成する (<スキーマ名>.<プロシージャ名>)
- 対処 2) パッケージ名付きのプロシージャ名にする (例: <パッケージ名_プロシージャ名>)

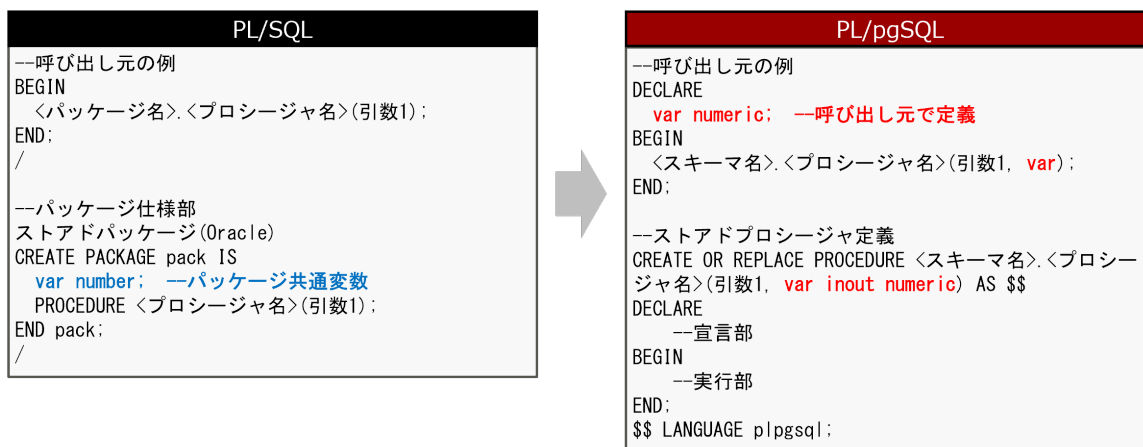
ここでは、対処 1 を選択したものとして説明します。

2) パッケージ名に対応するスキーマを作成します。

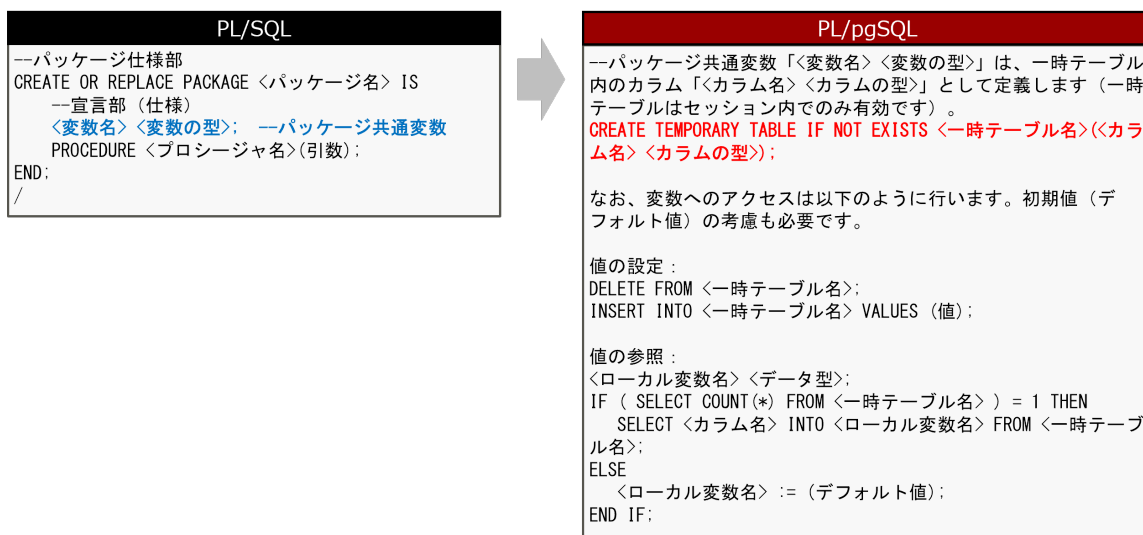


- 3) パッケージ共通変数の移行について、A)呼び出し元で定義する方法と、B)一時テーブル内のカラムで定義する方法を示します。

A) 呼び出し元で定義する方法

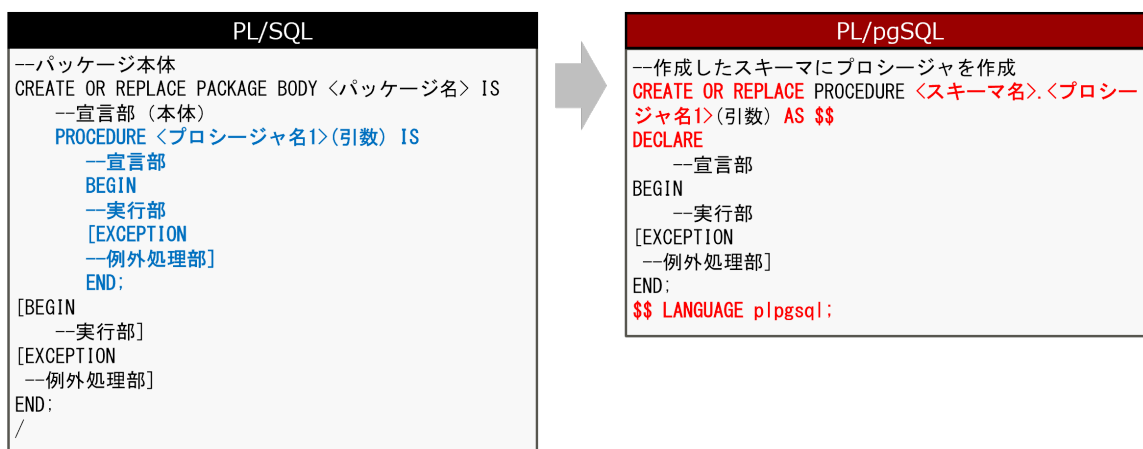


B) 一時テーブル内のカラムで定義する方法

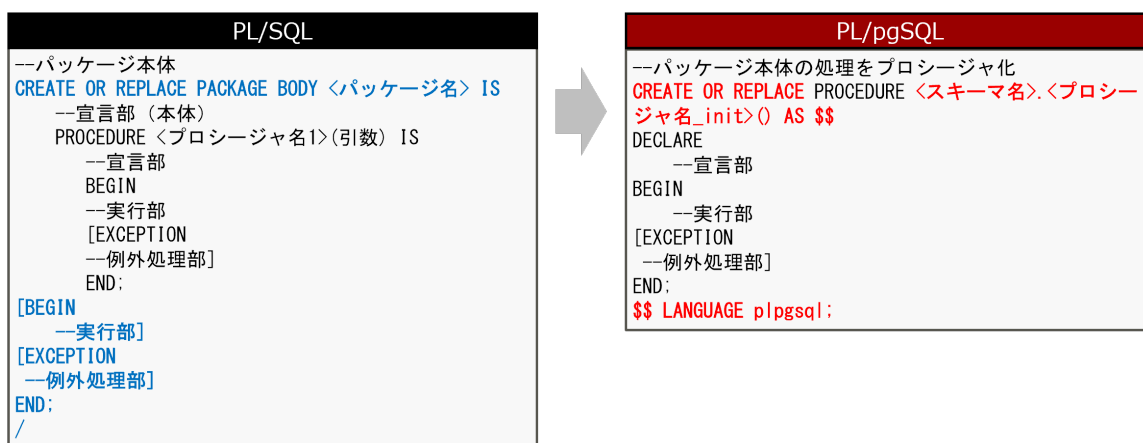


パッケージ本体の移行

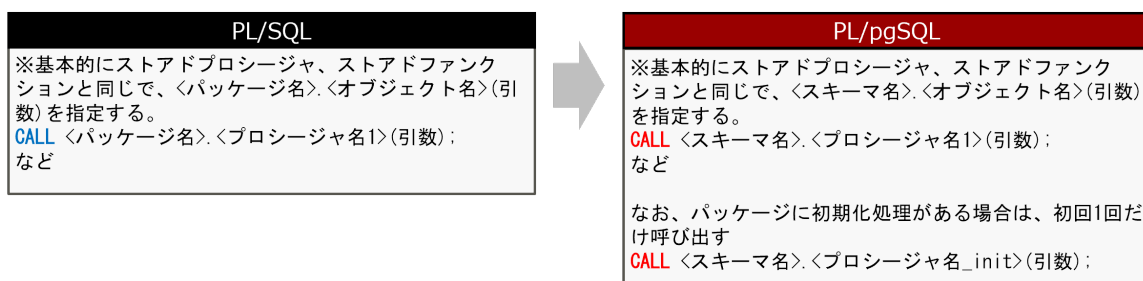
- 4) パッケージ内のプロシージャやファンクションは、作成したスキーマ配下に、ストアードプロシージャや、ストアードファンクションに変更します。



- 5) パッケージ自体に処理(初期化処理)がある場合は、必要であれば、同様にストアードプロシージャなどに変更します。



呼び出し方：



3. 移行例

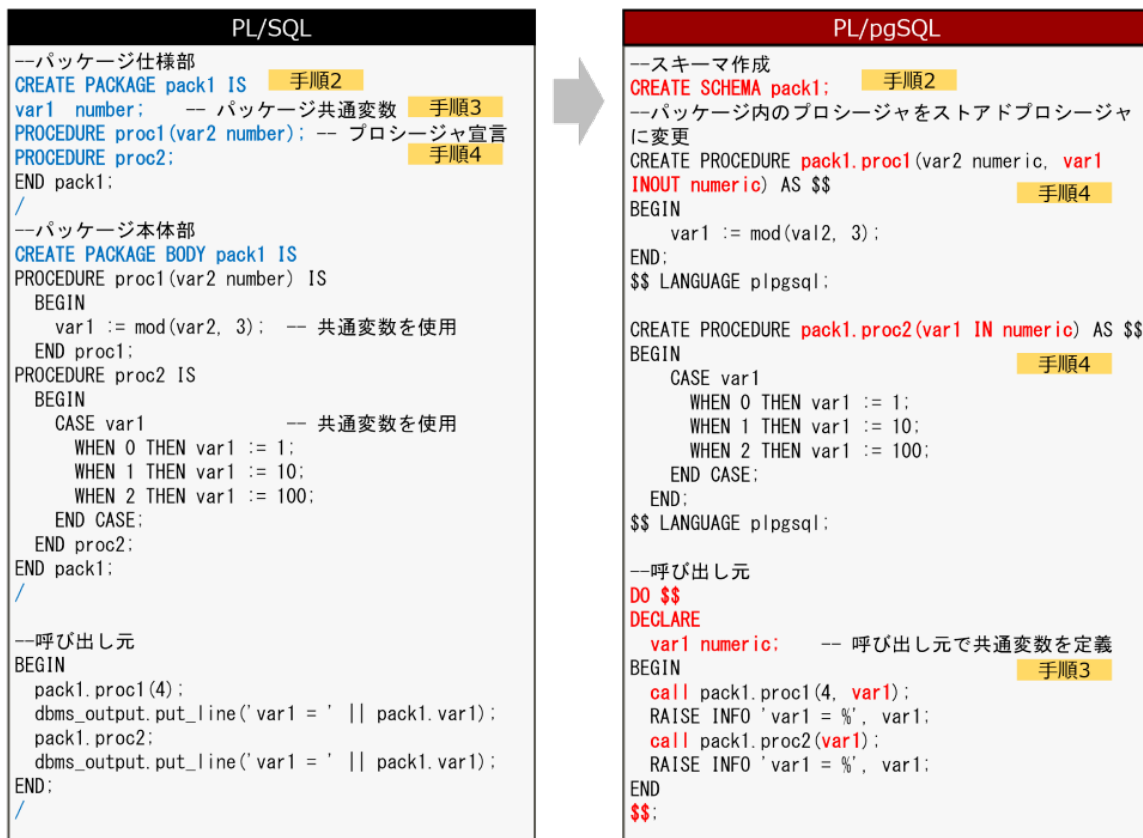
ストアド・パッケージの移行例について説明します。この例では、以下の対処を実施します。

- 手順1 パッケージ名の移行方法について、パッケージに対応したスキーマを作成する方針に決定します。
- 手順2 パッケージ名をスキーマ名としてスキーマを作成します。
- 手順3 パッケージ共通変数については、ここでは、呼出し元に移動し、各プロシージャ/ファンクションの呼び出しパラメーターに追加して受け渡すようにします。
- 手順4 パッケージ内のプロシージャ/ファンクションをストアドプロシージャ/ストアドファンクションに変更します。

パッケージ内のプロシージャ/ファンクションを抜き出して、新たに PostgreSQL のストアドプロシージャ/ストアドファンクションとして作成します。

なお、ここではパッケージ自体に初期化処理が無い例としていますので、2 章の 5)に相当する処理は不要です。

移行例：



参考

Oracle Database では、ストアド・プロシージャやストアド・ファンクションにおいても内部にサブプログラムや共通変数を含むことができます。しかし、PostgreSQL ではストアドプロシージャやストアドファンクションの中にそれらを入れることができません。移行に際してはパッケージと同様の対処が必要となります。

PL/SQL のストアド・パッケージを PostgreSQL に移行するにあたっての、移行概要やポイントについて説明しました。本記事がストアド・パッケージを安全かつスムーズ移行するために、お役に立てば幸いです。

参考

Enterprise Postgres では、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。

2022 年 12 月 2 日