Oracle Database10g PL/SQL

[Oracle][Oracle Database10g][PL/SQL]

http://otndnld.oracle.co.jp/document/products/oracle10g/102/doc_cd/appdev.102/B19257-01/toc.html

PL/SQL の概要

PL/SQL のメリット

SQL との緊密な統合

- PL/SQL では、SQL のすべてのファンクション、演算子および疑似列と同様に、SQL のデータ操作、カーソル制御およびトランザクション制御のすべてのコマンドを使用できます。
- PL/SQL で SQL のデータ型が完全にサポートされているため、アプリケーションとデータベースの間でデータをやり取りする際、データを変換する必要が少なくなります。
- SQL データ型と PL/SQL データ型の変換を行う必要はありません。
 - データベースの NUMBER または VARCHAR2 列は、PL/SQL の NUMBER または VARCHAR2 変数に格納されます。
- データ型を指定せずに表の列や行を操作できるため、表の定義が変更された場合のメンテナンス作業が簡略化。
- SQL 問合せの実行および結果セットの処理は、スクリプト言語でテキスト・ファイルを開いて各行を処理する場合と同様に簡単。
- トリガーやオブジェクト型など、データベースの多くの機能で使用されます。
- トリガー本体およびオブジェクト型のメソッドを PL/SQL で作成できます。
- 静的 SQL および動的 SQL の両方をサポート。

高いパフォーマンス

- PL/SQL があれば、複数文のブロック全体を Oracle に一度に送信できるため、データベースとアプリケーションの間のネットワークの通信量を大幅に削減できます。
- PL/SQL ストアド・プロシージャは一度コンパイルされてから実行可能なフォームで格納されるため、プロシージャ・コールは効果的。
- ストアド・プロシージャはデータベース・サーバー内で実行されるため、ネットワーク上で一度コールすると大規模なジョブを開始できる。
- ストアド・プロシージャはキャッシュされ、ユーザー間で共有され、これによって、必要なメモリー量と起動オーバーヘッドが減少。

高い生産性

- データを操作するコードを非常にコンパクトにすることができます。
- 例外処理、カプセル化、データ隠ぺいおよびオブジェクト指向のデータ型など、あらゆるソフトウェア・エンジニアリング機能が提供されているため、設計およびデバック時間を節約できます。
- Oracle Forms などのツールを拡張。
- ある 1 つの Oracle ツールで習得した PL/SQL の知識は他のツールにも利用できます。

完全な移植性

- Oracle データベースが動作する任意のオペレーティング・システムおよびプラットフォームで実行できます。
- 様々な環境で再利用できる、移植性の高いプログラム・ライブラリを作成できます。

優れたセキュリティ

- ・ストアド・プロシージャは、クライアントからサーバーへアプリケーション・コードを移動します。サーバーでは、コードの改ざんの防止、コード内の細部の隠ぺいおよびアクセスの制限が可能になります。

事前定義パッケージへのアクセス

- ・Oracle では、様々な実用的な作業を実行するために PL/SQL からコールできる API を定義した製品固有のパッケージを提供しています。

オブジェクト指向プログラミングのサポート

- ・オブジェクト型を使用すると、モジュール構造で維持および再利用が可能なソフトウェア構成要素を作成できるのみでなく、複数の異なるチームのプログラマが同時にソフトウェア構成要素を開発できます。
- ・データに対する操作をカプセル化すると、オブジェクト型を使用してデータ・メンテナンスのためのコードを SQL スクリプトや PL/SQL ブロックではなく、メソッドに入れることができます。
- ・オブジェクト型を使用すれば、実装の細部が隠されるため、クライアント・プログラムに影響することなく細部を変更できます。
- ・オブジェクト型を使用することで、現実のデータをモデル化できます。

Web アプリケーションおよび Web ページ開発のサポート

- ・Web アプリケーションおよび Server Pages (PSP) を開発できます。

PL/SQL の主な特長

- ・SQL のデータ操作機能と手続き型言語の処理機能の両方が利用できます。
- ・プロシージャ・コードに分割し、複数のアプリケーション間でそのコードを再利用できます。
- ・単純な SQL で問題が解決できる場合、PL/SQL プログラム内で SQL コマンドを直接発行できます。

PL/SQL ブロック構造

- ・プログラムを構成する基本単位 (プロシージャ、ファンクションおよび無名ブロック) は、内部で相互にネストできる論理ブロックです。
- ・ブロックは、関連する宣言および文をグループ化します。宣言は、それを使用する場所に近い場所に置くことができます。
- ・宣言はブロックの中で局所的に有効で、そのブロックが終了すると消滅します。

3つの基本部分で構成

- ・必ず存在する必要があるのは実行部のみ

1. 宣言部 (DECLARE)
2. 実行部 (BEGIN ..END)
3. 例外処理 (EXCEPTION)

実行例

権限の付与

EXAM ユーザーに、以下の権限を与える

- EXECUTE ANY PROCEDURE
- DEBUG ANY PROCEDURE
- ALTER ANY PROCEDURE
- CREATE ANY PROCEDURE
- DROP ANY PROCEDURE

```
SQL>grant create any procedure to exam;
```

```
Grant succeeded.
```

適当なのを作る

```
CREATE OR REPLACE procedure exam.test_proc is
  cursor c1 is
    select ip_address
    from exam.apache_access_log
    where ip_address not like '192%'
    group by ip_address
    order by ip_address
    ;
  log_rec c1%rowtype;
begin
  open c1;
  loop
    fetch c1 into log_rec;
    dbms_output.put_line('ip address:' || log_rec.ip_address);
  end loop;
  close c1;
end;
/
```