

本資料は、アシスト Oracle 研修をご受講いただいたお客様からのご質問や、研修ではご案内できなかった情報などをまとめたものです。研修受講後のスキルアップの一助として、是非お役立てください。

第 1 章 PL/SQL 概要	
1	<p>Q. 無名の PL/SQL ブロックを実行するには、権限が必要ですか。</p> <p>A. 無名の PL/SQL ブロックの実行に権限は必要ありません。</p> <p>※ストアド・サブプログラムを作成する場合は CREATE PROCEDURE、CREATE TRIGGER などのシステム権限が必要です。</p>
2	<p>Q. 現行インスタンスで実行された無名の PL/SQL ブロックを確認する方法はありますか。</p> <p>A. 現行インスタンスで実行されたことがある無名の PL/SQL ブロックは、管理者ユーザーでログイン後、以下の SQL 文で確認することができます。</p> <p>例)</p> <pre>SELECT substrb(sql_text,1,80) FROM v\$sqlarea WHERE command_type = 47 AND length(sql_text) &gt; 100 ←サイズを指定して下さい</pre> <p>なお、無名の PL/SQL ブロックは、実行するたびに解析が行なわれるため、CPU 負荷の原因となる可能性があります。サイズの大きい無名の PL/SQL ブロックは、ストアド化し、データベースに格納することをおすすめします。</p>
3	<p>Q. ブロック内のネスト最大数を教えてください。</p> <p>A. 255 までネストできます。</p>
第 2 章 PL/SQL の基本記述	
4	<p>Q. 暗黙カーソル属性を使用する際の注意事項はありますか。</p> <p>A. 暗黙カーソル属性はセッションに 1 つだけ存在するもので、直前に使用された暗黙カーソルの情報を参照します。</p> <p>暗黙カーソルは直前に行われた SQL 文の実行後、自動的にクローズされるため、暗黙カーソル属性は参照したい SQL 文 (SELECT...INTO 文や DML 文) の直後に使用することをおすすめします。</p>
5	<p>Q. 結果セットを 1 行飛ばしに FETCH することはできますか。</p> <p>A. できません。Oracle は必ず結果セットを上から 1 行ずつ FETCH します。</p>
6	<p>Q. カーソルを一度にいくつまでオープンできますか。</p> <p>A. 初期化パラメータ OPEN_CURSORS で指定した値までカーソルをオープンできます。</p> <p>※初期化パラメータ OPEN_CURSORS の詳細については「リファレンス」マニュアルをご参照ください。</p>

7	Q.	データベースから取り出された行が 1 行もない場合に、エラーを発生させずに処理することはできますか。
	A.	<p>カーソルで%ROWCOUNT 属性を使用することで、取り出された行が 0 行でもエラーを発生させずに処理できます。</p> <p>例)          …(省略)…          OPEN カーソル名; LOOP              FETCH カーソル名 INTO 変数名;              IF カーソル名%ROWCOUNT = 0 THEN                  処理 ; ←0 行の場合の対処を記述              EXIT; ←LOOP を抜ける。              END IF;          …(省略)…</p>
8	Q.	PL/SQL 内で ROWID を使用して SQL を高速に処理する方法を教えてください。
	A.	<p>SQL で特定の 1 行にアクセスする場合、WHERE 句に"WHERE 列名 = ~"と指定するよりも、"WHERE ROWID = ~"とした方が高速に SQL を処理できます。</p> <p>ROWID とは行のアドレスで、データが格納される物理的な場所を示しているため、データベース全体で行を一意に識別できます。</p> <p>そのため、ROWID を WHERE 句で指定すると、対象行が格納されている場所に直接アクセスできるので、特定の 1 行へ最も高速にアクセスできます。</p> <p>PL/SQL では、ROWID データ型を使用して効率よく、特定の行にアクセスする SQL を実行させることが可能です。</p> <p>例)          DECLARE              CURSOR cur_emp IS SELECT rowid,ename,sal,deptno FROM emp;          BEGIN              FOR emp_rec IN cur_emp LOOP                  UPDATE emp SET sal=sal*1.1                  WHERE rowid = emp_rec.rowid; ←ROWID の指定              END LOOP;          END;</p> <p>※CURRENT OF カーソルも同じように ROWID を使用するため、高速です。</p>
9	Q.	現在オープンされているカーソルの確認を行うことはできますか。
	A.	<p>V\$OPEN_CURSOR ビューでオープンされている以下のカーソルを確認できます。</p> <ul style="list-style-type: none"> <li>・ 暗黙カーソル</li> <li>・ 明示的にオープンされている静的カーソル</li> <li>・ オープンされ、まだクローズされていない動的カーソル</li> </ul> <p>※オープンされているがまだ解析されていないカーソルはカウントされませんのでご注意ください。          ※V\$OPEN_CURSOR ビューの詳細については「リファレンス」マニュアルをご参照ください。          ※動的カーソルについては「PL/SQL Appendix」でご紹介しています。</p>
10	Q.	識別子とデータベース・オブジェクトの名前が重複していた場合は、どちらが優先されるのですか。
	A.	データベース・オブジェクトの名前は、変数名や仮パラメータ名より優先されます。

11	Q.	IF 文や検索 CASE 文で複数の条件を指定する方法を教えてください。
	A.	<p>AND、OR 演算子を使用することで、複数の条件を指定できます。また条件の指定順番を意識することで、無駄な審査を省けます。</p> <ul style="list-style-type: none"> <li>・ A 条件 AND B 条件 ⇒ A、B 条件を共に満たす必要がある。 ※FALSE と評価されやすい条件を先に指定すると無駄な審査を省けます。</li> <li>・ A 条件 OR B 条件 ⇒ A、B 条件のどちらか一方が満たされればよい。 ※TRUE と評価されやすい条件を先に指定すると無駄な審査を省けます。</li> </ul> <p>例) …省略… IF var1=1 AND var2=2 THEN DBMS_OUTPUT.PUT_LINE('TRUE'); ELSE DBMS_OUTPUT.PUT_LINE('FALSE'); END IF;</p> <p>IF var1=3 OR var2=1 THEN DBMS_OUTPUT.PUT_LINE('TRUE'); ELSE DBMS_OUTPUT.PUT_LINE('FALSE'); END IF; …省略…</p>
12	Q.	GOTO 文や EXIT 文などの制御文はなぜ使用が推奨されていないのですか。
	A.	<p>PL/SQL のようなプログラムはブロック構造化言語であり、上から下に処理を進めていきます。しかし、GOTO 文や EXIT 文はそれを無視し、無条件で制御を自由に移すことができます。このような記述はプログラムの原則に反するばかりか、どのような過程で処理が行われているのか非常に分かりづらいプログラムになってしまいます。</p> <p>そのため、順次制御などを行う際は必ず理由をつけて処理を行うなど見やすく、分かりやすいプログラムを作成することを心掛けることを推奨します。</p>
13	Q.	カーソル FOR ループ内で例外が発生したとき、カーソルは明示的にクローズする必要がありますか。
	A.	<p>カーソル FOR ループ内で例外が発生した場合、カーソルは自動的にクローズされます。例外処理部にてカーソルを明示的にクローズする処理は必要ありません。</p>
14	Q.	複数のカーソルに全く同じ問い合わせ文が関連づけられていた場合、同じカーソルを使用するのでしょうか。
	A.	<p>複数のカーソルに同一の問い合わせ文が関連づけられていたとしても、それぞれ異なるカーソル名と関連づけられている場合は、共有メモリー上に異なる作業領域を確保します。共有はされません。</p>
15	Q.	PL/SQL 内で宣言した変数を、セッションを通して使用することは可能でしょうか。
	A.	<p>変数の有効範囲は、自ブロック、およびサブブロック内となるため、単体のストアド・サブプログラムではできません。セッションを通して変数を使用したい場合は、パッケージを作成してパブリックな変数を使用します。</p>
16	Q.	カーソルに対応づける問い合わせ文の選択リストに、式や関数を指定する場合の注意事項を教えてください。
	A.	<p>カーソルに対応づける問い合わせ文の選択リストに式や関数を指定する場合は、式や関数に別名を定義してください。</p> <p>例) CURSOR カーソル名 IS SELECT 式や関数 AS 別名 FROM 表名;</p> <p>別名を定義しなかった場合、%ROWTYPE などによる変数使用時にフィールド名を指定できません。</p>



23	Q.	パラメータの数やデータ型の違いで、同名のプログラムを作成できますか。
	A.	単体のストアド・サブプログラムでは行えませんが、パッケージを使用すれば可能です。
24	Q.	他のユーザーが所有する表にアクセスするストアド・サブプログラムの注意事項を教えてください。
	A.	他のユーザーが所有する表にアクセスするストアド・サブプログラムを作成/実行するためには、表を所有するユーザーから直接オブジェクト権限を付与する必要があります。ロール経由でオブジェクト権限を与えるだけではエラーが発生します。
25	Q.	ストアド・サブプログラムに設定したパラメータの情報を確認する方法を教えてください。
	A.	ストアド・サブプログラムに設定したパラメータに関する情報は、USER_ARGUMENTS ビューや SQL*Plus の DESCRIBE コマンドで確認できます。  ただし、パラメータのデフォルト値については上記の方法で確認できないため、USER_SOURCE ビューを使って直接ソースコードを確認する必要があります。  ※USER_ARGUMENTS ビュー、USER_SOURCE ビューの詳細については、「リファレンス」マニュアルをご参照ください。
26	Q.	パラメータのデータ型に%ROWTYPE を指定することはできますか。
	A.	できます。パラメータのデータ型に%ROWTYPE を指定することで、ストアド・サブプログラム間で結果セットのデータを受け渡すことができます。
27	Q.	ファンクション名と列名の優先順位について教えてください。
	A.	列名と同名のファンクションを、パラメータを指定せずに実行した場合、列名の方が優先して認識されます。この場合、ファンクションをスキーマ名で修飾することで、ファンクションとして認識させることができます。  ※ただし、ソースコードの判読が難しくなるため、列名と同名のファンクションは作成しないことをおすすめします。
<b>第 4 章 トリガー</b>		
28	Q.	データベース・トリガー内でデータベース・リンクを使用してリモート・データベースに対して処理を行うことは可能ですか。
	A.	通常のストアド・サブプログラム同様に、データベース・トリガー内でもデータベース・リンクを使用することは可能です。
29	Q.	INSERT、UPDATE を行った時刻を同じ表の列に記録するには、どうすればよいですか。
	A.	BEFORE 行トリガーと SYSDATE 関数を使用します。  例) DEPT 表に INSERT または UPDATE を行った時間を LAST_DML 列に記録する CREATE OR REPLACE TRIGGER dept_tri BEFORE INSERT OR UPDATE ON dept FOR EACH ROW BEGIN SELECT SYSDATE INTO :NEW.LAST_DML FROM dual; END;
30	Q.	トリガーを、「毎週日曜日」のように、スケジュールに従って定期的に実行できますか。
	A.	トリガーは、スケジュールに応じて起動させることはできません。スケジュールに応じてプログラムを動作させる際には、DBMS_SCHEDULER パッケージなどを使用します。

31	Q.	トリガー内でコールされたストアド・サブプログラム内にトランザクション制御文が含まれていた場合は、エラーが発生しますか。
	A.	トリガーが実行された時にエラーが発生します。 これは、トリガーが起動されたトランザクションとコールされたストアド・サブプログラムのトランザクションが同一のためです。 それぞれのトランザクションを独立させたい場合は、自律型トランザクションを指定してください。
32	Q.	無限ループを解決するにはどうすればよいですか。
	A.	無限ループを解決するには、管理者ユーザーでデータベースに接続し、無限ループが発生している対象セッションを強制的に切断します。  ※セッションの強制切断については「データベース・マネジメント」コースでご紹介しています。
<b>付録</b>		
33	Q.	NCHAR、NVARCHAR2 型などでサポートされている NLS 文字がどのようなものなのか教えてください。
	A.	NLS 文字が使用できる環境下では、各国語やキャラクタ・セットを意識せずに処理を行うことができます。  NLS 文字が使用できる環境では、Oracle サーバとクライアントでキャラクタ・セットが異なっても自動的に変換が行われ、処理を行うことができます。  また NLS の環境下では各国語がサポートされるため、アプリケーションの修正を行うことなくそれぞれの国にあった言語、日付、数値などを扱うことができます。
34	Q.	事前定義の内部例外 (A-21) に ORA-xx から始まる番号と、符号付きの整数がありますが、これらは何を意味しますか。
	A.	ORA-xx から始まる番号は、Oracle のエラー番号を意味し、符号付きの整数は、SQLCODE 関数の戻り値を意味します。
35	Q.	PL/SQL の予約語を確認する方法を教えてください。
	A.	V\$RESERVED_WORDS ビューで確認できます。
36	Q.	自律型トランザクション設定したプログラムにトランザクション制御文(COMMIT、ROLLBACK 文)を含めないとどうなりますか。
	A.	プログラム終了までに、トランザクションが終了しなかった場合は例外が発生します。
<b>その他</b>		
37	Q.	再帰サブプログラムについて教えてください。
	A.	自分自身を呼び出すサブプログラムのことを、再帰サブプログラムといいます。再帰サブプログラムを使用することによってプログラムの設計を単純化することができる場合があります。  例) 階乗計算するためのファンクションを再帰サブプログラムで作成する。 CREATE OR REPLACE FUNCTION kaijou(n NUMBER) RETURN INTEGER IS BEGIN IF n=1 THEN RETURN (1); ELSE RETURN (n*kaijou(n-1)); ←自分自身をコー END IF; END;  /* 再帰サブプログラムの呼び出し */ SELECT kaijou(4) FROM dual;  KAIJOU(4) ----- 24  ※再帰サブプログラムでは終了条件を記述しないと無限ループが発生するため、必ず終了条件を記述して下さい。

38	Q.	SQL*Plus で PL/SQL ブロックを実行すると、画面に「PL/SQL プロシージャが正常に完了しました。」と表示されないようにする方法を教えてください。
	A.	SQL*Plus の SET コマンドで可能です。 SET FEEDBACK OFF を SQL*Plus で実行してください。
39	Q.	データベースに登録済みのストアド・サブプログラムのソースを OS ファイルに保存する方法を教えてください。
	A.	<p>1.以下のようにスクリプト・ファイル (source.sql) を作成します。</p> <pre> ----- SET ECHO OFF          --ファイル内の実行コマンドを非表示にする SET HEADING OFF       --検索結果の列ヘッダーを非表示にする SET FEEDBACK OFF      --問い合わせ結果のレコード数を非表示にする SET VERIFY OFF        --置換変数を値と置き換える前後の SQL を非表示にする DEFINE exte =         --'.sql' →出力ファイルの拡張子を設定する ACCEPT name CHAR PROMPT 'INPUT PROCEDURE or FUNCTION NAME! :                         --変数を宣言する。                         -- (PROMPT オプションで指定したプロンプトをユーザーに表示できる) SPOOL &amp;name&amp;exte      --出力結果を OS ファイルに保存する                         -- (ファイル名は先に宣言した変数を使用する) SET TERM OFF          --スクリプトからの出力を非表示にする SELECT text FROM user_source WHERE name = '&amp;name';                         -- ストアド・サブプログラムのソースコードを検索する  SET TERM ON SPOOL OFF SET VERIFY ON SET FEEDBACK ON SET HEADING ON SET ECHO ON ----- </pre> <p>2.SQL*Plus から上記のスクリプト・ファイルを実行します。  @source.sql  INPUT PROCEDURE or FUNCTION NAME! : NEWSAL_PRO  ↑ソースを調べたいサブプログラムの名前を入力します(大文字で)</p> <p>※生成されたファイル名は「ストアド・サブプログラム名.sql」となります</p>
40	Q.	SQL*Plus に PL/SQL を直接記述していた場合、途中で記述をやめる方法はありますか。
	A.	新しい行に「.」(ピリオド)のみを入力して Enter キーを押してください。 ※新しい行に「/」(スラッシュ)を入力しても終了になりますが、この場合は PL/SQL を実行して終了します。 ブロックの記述が途中である場合はエラーが返ります。
41	Q.	PL/SQL コードサイズの最大値を教えてください。
	A.	256MB までです (トリガーは 32KB までです)。  ただし、コードサイズの最大値は内部計算によって求められるため、正確な値を算出することはできません。あくまでも目安なので、必ずしもこれらの値で作成できるとは限りませんので、ご了承ください。

※ ご利用上の注意事項※

- ・本書の著作権は株式会社アシストに帰属します。
- ・本書は参考資料であり、掲載されている情報は予告なしに変更されることがあります。
- ・本書で使用している製品の名称は、各社の商標または登録商標です。
- ・本資料の内容に関するご質問はご遠慮ください。
- ・本資料はお客様の責任のもとでご利用ください。これらの使用によりいかなる損害が生じたとしても、株式会社アシストは一切保証致しかねますので、ご了承ください。