

PL/SQL プログラミング II

～研修受講後のスキルアップサポート～

対応バージョン : Oracle 10gR1 ~ 12cR1

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 FAQ にまとめたものです。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。

※ご利用上の注意事項は最後のページにまとめられております。ご確認のうえ、ご利用ください。

第1章 コンポジット型

1	Q.	結合配列を使用する際、初期化の記述は必要ありませんか。
	A.	結合配列は、値の代入時に自動的に初期化が行われますので、記述は必要ありません。
2	Q.	結合配列内のデータを INSERT する際、結合配列のフィールドと表の行構造が対応していれば、フィールド名を修飾せずに、VALUES 句に指定できますか。
	A.	<p>フィールド名の修飾なしに、指定が可能です。 また UPDATE の場合にも、同様にフィールド名を省略できます。</p> <p>例)</p> <ul style="list-style-type: none"> ・ INSERT INTO 表名 VALUES 結合配列(索引番号); ・ UPDATE 表名 SET ROW = 結合配列(索引番号) WHERE 条件;
3	Q.	結合配列の索引番号で指定できる、値の範囲を教えてください。
	A.	<p>数値 (PLS_INTEGER、BINARY_INTEGER 型) の場合は、-2147483648 から 2147483647 です。 文字値 (VARCHAR2 型) の場合は、指定したサイズに基づきます。 例) VARCHAR2 (5) と指定した場合は、5 バイトまで設定できます。</p> <p>※索引番号は、連続している必要ありません。</p>
4	Q.	結合配列の要素を削除するコレクション・メソッド DELETE の結果について。
	A.	<p>コレクション・メソッド DELETE を使用すると、結合配列の要素が削除され、使用していたメモリ領域は解放されます。</p> <p>要素は解放されるため、NULL とは異なります。そのため、NOT NULL 制約を定義していたとしても、コレクション・メソッド DELETE を実行できます。</p> <p>なお、コレクション・メソッド DELETE によって削除された要素を呼び出そうとすると NO_DATA_FOUND 例外が発生します。</p>
5	Q.	要素が挿入されていない結合配列にコレクション・メソッド LAST を使用した結果について。
	A.	結合配列に要素が全く定義されていない場合は、コレクション・メソッド LAST は NULL を戻します。

第2章 バルク配列処理	
6	Q. FOARLL 文の INDICES OF 句を使用する際、複数のコレクションを指定することができますか。
	A. 複数のコレクションは指定できません。 代替案として、指定するコレクションに複数フィールドを設定する方法があります (OracleDatabase11g から複数フィールド設定できます)。
7	Q. FORALL 文で使用する結合配列の要素のデータ型に PL/SQL のデータ型を使用できますか。
	A. できません。 結合配列にバルク・バインド入力された値は、その後の SQL 文に渡されるため、要素のデータ型は SQL のデータ型で定義する必要があります。
第3章 システム固有の動的 SQL	
8	Q. 実行時にプロシージャ名を動的に指定する方法
	A. 実行時にプロシージャ名を動的に指定するには、動的 SQL を使用します。 実行するプログラム名をパラメータとして動的に受け渡すことでプログラムを動的に実行することができます。 プログラム作成例は以下の通りです。 例) TEST1、TEST2 プロシージャを動的に呼び出すプログラムを作成 ※パラメータの値に1、もしくは2を引き渡すかで毎回異なったプログラムを呼び出せます。 <pre>CREATE OR REPLACE PROCEDURE TEST(i number) IS w_jikko varchar2(100); BEGIN w_jikko := 'begin TEST' i '; end;'; EXECUTE IMMEDIATE w_jikko; END;</pre>
第4章 カーソル変数	
9	Q. カーソル変数に対して同時に複数の問合せを対応付けることはできますか。
	A. カーソル変数に対して同時に複数問合せを対応付けることはできません。 カーソル変数に別の結果セットをオープンした時点で以前の結果セットの情報は失われます。
第5章 ユーザー定義パッケージ	
10	Q. パッケージ内にどんなストアド・サブプログラムが含まれているかを確認することはできますか。
	A. USER_SOURCE ディクショナリに問い合わせることで確認することができます。 詳細は以下の通りです。 ※SQL*Plus アプリケーションを使用している場合は DESCRIBE コマンドで確認できます。

11	Q.	異なるブロック間でオーバーロードを行うことはできますか。
	A.	できません。 オーバーロードされたプログラムの有効範囲は同じでなくてはなりません。 つまり必ず同じブロック内に定義されている必要があります。
12	Q.	パッケージは共有メモリどこにロードされるのでしょうか。
	A.	共有プールのライブラリキャッシュにロードされます。
13	Q.	パッケージに含めるオブジェクト数に制限はありますか。
	A.	ありません。 ※ソースコードサイズの制限を超えることはできません。
第6章 ユーティリティ・パッケージ		
14	Q.	UTL_FILE パッケージで指定した OS ファイルが存在チェックやサイズチェックをする方法
	A.	UTL_FILE.FGETATTR プロシージャを使用することで、ディレクトリに指定した OS ファイルに関する情報を取得することができます。この機能を利用し、OS ファイルの有無やサイズに応じて、プログラムの動作を制御できます。 ※UTL_FILE.FGETATTR プロシージャの詳細については「PL/SQL パッケージ・プロシージャ およびタイプ・リファレンス」マニュアルをご参照ください。
15	Q.	UTL_FILE パッケージ使用時のみの例外を使用するときの注意事項
	A.	INVALID_PATH 例外や FILE_OPEN 例外などの UTL_FILE パッケージ使用時のみの例外は、UTL_FILE パッケージ内に含まれている例外であるため、使用時には例外名の前に「UTL_FILE.」というパッケージ名を修飾する必要があります。
16	Q.	UTL_FILE_DIR パラメータに指定するディレクトリ数に制限はありますか。
	A.	制限はありません。

17	Q.	UTL_FILE.FCLOSE_ALL プロシージャは、どのような時に使用すべきでしょうか。
	A.	<p>UTL_FILE.FCLOSE_ALL プロシージャはプログラムの処理が異常終了した時の対処として使用すると有効です。</p> <p>UTL_FILE パッケージを使用してファイルをオープンすると、ファイルをクローズするまで、他のユーザーはそのファイルにアクセスすることができません。</p> <p>また、この場合、処理中にプログラムが異常終了してもファイルはオープンされたままになります。</p> <p>そこで、例外処理部で UTL_FILE.FCLOSE_ALL プロシージャの実行を指定することで、プログラムの異常終了時の他ユーザーへの影響を、最小限にとどめることができます。</p>
18	Q.	ディレクトリ・オブジェクトのパスを確認する方法
	A.	<p>ディレクトリ・オブジェクトの設定は DBA_DIRECTORIES ビューにて確認できます。</p> <p>例)</p> <pre>SQL> SELECT owner, directory_name, directory_path FROM dba_directories 2 WHERE directory_name='UTL_DATA';</pre> <p>OWNER DIRECTORY_ DIRECTORY_PATH ----- SYS UTL_DATA D:¥work¥PL2 ← UTL_DATA ディレクトリ・オブジェクトのパスを確認</p>
19	Q.	UTL_FILE.FCLOSE プロシージャを実行しなかったときの動作について
	A.	<p>UTL_FILE.FCLOSE プロシージャを実行しなかった場合、セッションが終了した時点で対象のファイルがクローズされるため、セッションが終了するまでは他のユーザーが対象のファイルにアクセスできなくなります。</p> <p>そのため、処理が終わった時点で UTL_FILE.FCLOSE プロシージャを実行し、明示的にファイルをクローズすることをお勧めします。</p>
20	Q.	UTL_FILE.FFLUSH プロシージャはどのような時に使用するのですか。
	A.	<p>UTL_FILE.FFLUSH プロシージャは、呼び出し時にデータを強制的にファイルに書き込みます。そのため、リアルタイムにデータを更新したい場合は便利です。</p> <p>※ファイルへの書き込みは UTL_FILE.PUT_LINE プロシージャでも行えますが、こちらはある程度データをバッファに格納してから書き込み処理を行うため、多少の時間差が発生します。</p>

21	Q.	DBMS_SCHEDULER パッケージでスケジュールを作成するときに、月曜日と金曜日など1つのスケジュールに複数の実行タイミングを指定できますか。
	A.	<p>実行間隔をカレンダー式で指定することで、1つのスケジュールに複数の実行タイミングを指定できます。</p> <p>例) 毎週月曜日と金曜日にジョブを実行する FREQ=WEEKLY;BYDAY=MON, FRI;</p> <p>※ただし「FREQ」で指定する、月ごとや週ごとのような繰返しのタイプは1スケジュールにつき1つしか指定できません。</p>
22	Q.	DBMS_SCHEDULER パッケージで作成したスケジュールを変更した場合、スケジュールに紐付けられたジョブの再作成は必要ですか。
	A.	必要ありません。
23	Q.	ジョブの前回実行日と、実行したときの所要時間を確認する方法を教えてください。
	A.	<p>ジョブの前回実行日と実行したときの所用時間を確認するには USER_SCHEDULER_JOBS ビュー (または DBA_SCHEDULER_JOBS ビュー) で確認できます。</p> <p>※DBMS_JOB パッケージで作成したジョブの情報は、USER_JOBS ビュー (または DBA_JOBS ビュー) で確認できます。</p> <ul style="list-style-type: none"> ・ USER_SCHEDULER_JOBS ビュー ジョブの前回実行日 : last_start_date 列 ジョブの所要時間 : last_run_duration 列 ・ USER_JOBS ビュー ジョブの前回実行日 : last_date 列 ジョブの所要時間 : total_time 列
24	Q.	同じ日に実行されるジョブが複数あったとき、優先順位は設定できますか。
	A.	DBMS_SCHEDULER パッケージでジョブのクラス(ジョブをグループ化したものです)を作成すれば、クラス内のジョブに優先順位を設定できます。または、実行する時間を細かく設定して、実行順番を制御する方法もあります。
25	Q.	実行中のジョブを強制終了させるにはどうしたら良いでしょうか。
	A.	<p>DBMS_SCHEDULER パッケージで登録したジョブを実行中に強制終了させるには、DBMS_SCHEDULER.DBMS_STOP プロシージャを使用します。繰返しの実行がスケジュールされている場合は、次のタイミングでジョブが起動します。</p> <p>また、DBMS_JOB パッケージでは強制終了のプログラムがないため、管理者でジョブ実行ユーザーのセッションを強制終了します。</p> <p>※セッションの強制終了については「データベース・マネジメント」コースでご紹介しています。</p>

26	Q.	現在中断状態であるすべてのジョブ、及びそのジョブが失敗した回数を表示する方法												
	A.	<p>現在中断状態にあるジョブとその失敗回数は以下のビューで確認できます。</p> <p>(1) DBMS_JOB パッケージで作成したジョブ DBA_JOBS ビュー（または USER_JOBS ビュー）の BROKEN 列が 'Y' となっているジョブが中断状態です。 また、失敗回数は FAILURES 列で確認できます（16 回失敗すると中断状態となります）。</p> <p>例) SQL> SELECT what, failures FROM dba_jobs WHERE broken = 'Y';</p> <table border="1"> <thead> <tr> <th>WHAT</th> <th>FAILURES</th> </tr> </thead> <tbody> <tr> <td>FILE_UNLOAD_EMP</td> <td>16</td> </tr> </tbody> </table> <p>(2) DBMS_SCHEDULER パッケージで作成したジョブ DBA_SCHEDULER_JOBS ビュー（または USER_SCHEDULER_JOBS ビュー）の ENABLED 列が「FALSE」と なっているジョブが中断状態です。</p> <p>また、失敗回数は FAILURE_COUNT 列で確認できます。max_failures の値で許容失敗数を制御 できます（デフォルトは無制限です）。例えば、以下の例では max_failures を 10 と設定して いるため、10 回失敗するとジョブが中断状態となります。</p> <p>SQL> SELECT job_name, enabled, failure_count, max_failures 2 FROM user_scheduler_jobs;</p> <table border="1"> <thead> <tr> <th>JOB_NAME</th> <th>ENABLED</th> <th>FAILURE_COUNT</th> <th>MAX_FAILURES</th> </tr> </thead> <tbody> <tr> <td>TEST_JOB</td> <td>FALSE</td> <td>10</td> <td>10</td> </tr> </tbody> </table>	WHAT	FAILURES	FILE_UNLOAD_EMP	16	JOB_NAME	ENABLED	FAILURE_COUNT	MAX_FAILURES	TEST_JOB	FALSE	10	10
WHAT	FAILURES													
FILE_UNLOAD_EMP	16													
JOB_NAME	ENABLED	FAILURE_COUNT	MAX_FAILURES											
TEST_JOB	FALSE	10	10											

付録

27	Q.	DBMS_OUTPUT パッケージで使用するバッファサイズの変更方法
	A.	<p>DBMS_OUTPUT パッケージではデータを読み込んだり、書き込んだりする際にバッファを使用しますが SQL*Plus ではデフォルト値が 2000 となっています（※10gR2 以降はデフォルトが UNLIMITED です）。</p> <p>バッファサイズを変更したい場合は以下のように実行してください。</p> <p>例)</p> <p>1. SQL*Plus では SERVEROUTPUT 環境変数設定時に バッファサイズの変更を行う。</p> <pre>SQL> /* バッファサイズを 50000 に変更する */ SQL> SET SERVEROUTPUT ON SIZE 50000</pre> <p>2. DBMS_OUTPUT.ENABLE プロシージャで変更する。</p> <pre>SQL> begin 2 dbms_output.enable(50000); 3 end;</pre> <p>※バッファサイズは 2000～1000000 バイト内で指定します。</p>

28	Q.	WRAP ユーティリティで暗号化したコードを元に戻せますか。
	A.	WRAP ユーティリティで暗号化したコードは元に戻せません。 そのため、元のソースコードはファイルなどに残しておいてください。
29	Q.	DBMS_JOB パッケージで作成したジョブを DBMS_SCHEDULER パッケージの DROP_JOB プロシージャで削除できますか。
	A.	できません。DBMS_JOB で作成したジョブは、ジョブ番号を確認し、DBMS_JOB.REMOVE プロシージャで削除してください。
30	Q.	USER_JOBS ビューの NEXT_DATE 列が 4000 年 1 月 1 日になっているのはなぜでしょうか。
	A.	ジョブに登録したプログラムが中断されている状態で USER_JOBS ビューを確認すると 4000 年 1 月 1 日という日にちになります。 DBMS_JOB.BROKEN プロシージャを使用して、ジョブを中断し、プログラムを再開すると設定した NEXT_DATE の値に戻ります。 ※UTL_FILE.BROKEN プロシージャの詳細については「PL/SQL パッケージ・プロシージャ およびタイプ・リファレンス」マニュアルをご参照ください。
31	Q.	PL/SQL ブロック内で複数のアラートを受け取ることはできますか。
	A.	WAITANY プロシージャは複数アラートが発生していても、ひとつの WAITANY プロシージャにつき最新のアラート 1 つしか受け取ることができません。 WAITONE/WAITANY プロシージャの前に複数のアラートが発生した場合は最新の SIGNAL プロシージャによって通知されるメッセージが戻され、それ以前のメッセージは破棄されます。 しかし、LOOP 間に複数 WAITANY プロシージャを定義することで、複数のアラートを受け取れます。 例) BEGIN DBMS_ALERT.REGISTER('SIGNAL1'); DBMS_ALERT.REGISTER('SIGNAL2'); LOOP DBMS_ALERT.WAITANY(ALERT_NAME, MES, STATUS, 30); DBMS_ALERT.WAITANY(ALERT_NAME, MES, STATUS, 30); IF STATUS = 0 THEN EXIT; END IF; END LOOP; END;

32	Q.	DBMS_ALERT.SET_DEFAULTS プロシージャで設定する「ポーリング間隔」とは何ですか。							
	A.	<p>「ポーリング」とは、DBMS_ALERT.REGISTER プロシージャによって登録リストに登録されているアラートを一定間隔で監視する作業のことをいいます。</p> <p>ちなみに、デフォルトでは1秒であり、WAITONE/WAITANY プロシージャのTIMEOUTパラメータで設定されている間、監視を行います。</p>							
33	Q.	アラートの削除を行う必要性について							
	A.	<p>アラートが発生すると、登録リスト上の全セッションに通知が行われます。</p> <p>アラートの登録をしたままにしていると、不要な通知を行わなくてはいけなくなり、サーバーに余計な負荷をかけてしまいます。</p> <p>通知側の負荷を減らすために、不要なアラートは必ず REMOVE、REMOVEALL プロシージャで削除するようにして下さい。</p>							
34	Q.	DBMS_PIPE で作成された PIPE の情報を確認することはできますか。							
	A.	<p>V\$db_PIPE ビューにて確認することができます。</p> <pre>SQL> SELECT * FROM v\$db_pipes;</pre> <table border="1"> <thead> <tr> <th>OWNERID</th> <th>NAME</th> <th>TYPE</th> <th>PIPE_SIZE</th> </tr> </thead> <tbody> <tr> <td></td> <td>TEST_PIPE</td> <td>PUBLIC</td> <td>542</td> </tr> </tbody> </table>	OWNERID	NAME	TYPE	PIPE_SIZE		TEST_PIPE	PUBLIC
OWNERID	NAME	TYPE	PIPE_SIZE						
	TEST_PIPE	PUBLIC	542						
35	Q.	パイプのセキュリティについて							
	A.	<p>DBMS_PIPE パッケージで使用できるパイプにはパブリック・パイプとプライベート・パイプがあります。パイプのセキュリティを強化したい場合は、プライベート・パイプを使用してください。</p> <p>■ プライベート・パイプ プライベート・パイプは DBMS_PIPE パッケージをコールする以前に、DBMS_PIPE.CREATE_PIPE ファンクションを使用して事前に明示的に作成されたパイプのことをいいます。</p> <p>プライベートパイプは管理者、所有者、権限が付与された人しかアクセスすることができないパイプとして定義することができるため、意図しない第3者にデータを読み込まれることを防止できます。</p> <p>※プライベートパイプは REMOVE_PIPE プロシージャで明示的にパイプを削除するまで共有メモリー上に存在します。</p> <p>■ パブリック・パイプ DBMS_PIPE.CREATE_PIPE ファンクションで作成されていないパイプ名をプログラム内で指定した場合はパブリックパイプとなり、Oracle が暗黙的にパイプを作成してくれます。</p> <p>このパブリックパイプは権限などは必要なく、パイプ名を知っていれば誰でもメッセージを読み込むことが可能になるため、意図しない第3者にデータを読み込まれる危険性があります。</p>							

36	Q.	DBMS_SHARED_POOL パッケージを使用して共有メモリ上に固定されているプログラムを確認する方法
	A.	V\$DB_OBJECT_CACHE ビューを確認することで、共有メモリ(共有プール)上に固定されているプログラムの情報が確認できます。 ※V\$DB_OBJECT_CACHE 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。
37	Q.	共有メモリーの情報をクリアした場合、DBMS_SHARED_POOL パッケージでメモリーに固定しているオブジェクトも消去されてしまうのですか。
	A.	ALTER SYSTEM FLUSH SHARED_POOL コマンドを実行すると、共有プール上のデータが消去されます。しかし、現在実行中のものや、DBMS_SHARED_POOL パッケージでメモリー上に固定しているオブジェクトは消去されません。

※ 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。