

DBMS_ALERT パッケージ

機能

プログラム間での（アラートなどの）メッセージ通信を行う方法

【通信開始タイミングは、COMMIT 実行時】（非同期）

（レコード内容の変更などの操作が起こった場合に、使う）

※ 受信側プロセスは、メッセージ受取りステートメントで待機状態になる

DBMS_PIPE との違いは、COMMIT を行った後に（処理確定時に）メッセージが送られる

使い方概要

【アラート処理の流れ】

[受信側]

① アラートを登録する

```
DBMS_ALERT.REGISTER('アラート名');
```

② アラートを待機させる

```
DBMS_ALERT.WAITONE('アラート名', <受信メッセージ受取り変数>, <ステータス変数>, 最長待機時間秒数);
```

[送信側]

③ アラート通知の命令発行を行う

```
DBMS_ALERT.SIGNAL('アラート名', 'メッセージ');
```

④ **COMMIT** の発行

【このタイミングで、実際のアラート通知が行われる】

⑤ 待機解除が、「アラート受信」

or 「タイムアウト」かの判断

```
IF <ステータス変数> = 1 THEN RETURN;  
END IF;
```

⑥ アラートの廃棄

```
DBMS_ALERT.REMOVE('アラート名');
```

サンプル・コード

[受信側]

```
CREATE OR REPLACE PROCEDURE alert_wait
IS
  status INTEGER;
  mes VARCHAR2(2000);
BEGIN
  DBMS_ALERT.REGISTER('ALERT_TEST'); ← ①
  DBMS_ALERT.WAITONE('ALERT_TEST', mes, status, 30); ← ②
  IF status = 1 THEN RETURN; ← ⑤
  END IF;
  DBMS_OUTPUT.PUT_LINE(mes);
  DBMS_ALERT.REMOVE('ALERT_TEST'); ← ⑥
END;

/ ← プロシージャの登録のための / (スラッシュ)

EXECUTE alert_wait(); ← プロシージャの起動
```

[送信側]

プログラムからの通知

```
EXECUTE DBMS_ALERT.SIGNAL('ALERT_TEST', 'メッセージ → 送信');

COMMIT;
```

もしくは、

トリガーからの通知

```
CREATE OR REPLACE TRIGGER alert_sent
  AFTER DELETE ON dept
BEGIN
  DBMS_ALERT.SIGNAL('ALERT_TEST', 'DEPT 表の行を削除しました。');
END; ← ③

/ ← プロシージャの登録のための / (スラッシュ)
```

※ dept 表の行を削除した時に、alert_sent トリガーが動作してくる。
この行を削除した処理の中で、COMMIT 操作を発行する ← ④

【アラート処理の実行結果】

[受信側]

```
SQL> SET SERVEROUTPUT ON
```

(メッセージ出力を、SQL*Plus 画面に表示させる)

```
SQL> EXECUTE alert_wait
```

```
|  
|
```

アラートの待機中

```
|  
|  
↓
```

「 DEPT 表の行を削除しました。 」
と、表示される

[送信側]

```
SQL> SET SERVEROUTPUT ON
```

```
SQL> DELETE FROM dept  
        WHERE deptno = 55;
```

```
SQL> COMMIT;
```