DBMS_LOCK パッケージ

機能

他のプロセスとの同時処理を防止するための制御ロジックである
PL/SQL の処理の同時並行処理防止のためのロジカル・ロック制御の実行方法

使い方概要

① ロック ID の取得

DBMS_LOCK.ALLOCATE_UNIQUE('ロック名', ＜ロックハンドル変数＞);

② ロックの要求

lock_status := DBMS_LOCK.REQUEST(＜ロックハンドル変数＞ ,
ロックモード , ロック獲得待機最長時間秒数 , TRUE);
ロックモード：6 は排他

・取得しようとしているロック名が空いていれば、ロックを獲得して、次のステップの命令を実行する
・他のプロセスがすでにロックを獲得済みであれば、**ロックが解放されるまでプロセスは待機状態になる**
・ロック獲得済みプロセスがロックを解放した時は、ロック取得待ち状態のプロセスが替わってロックを取得して、次のステップの命令を実行する

↓

③ ロックの解放

＜release ステータス変数＞ :=
DBMS_LOCK.RELEASE(＜ロックハンドル変数＞);

その他の DBMS_LOCK パッケージ機能

DBMS_LOCK.SLEEP(秒数);

プログラムの処理を、指定した秒数の間　停止（スリープ）させる

この機能には、DBMS_LOCK.ALLOCATE_UNIQUE との関係は無いので、単独で実施することができる

サンプル・コード

```
CREATE  OR  REPLACE  PROCEDURE  lock_exe
IS
  lock_handle      VARCHAR2( 128 ) ;
  lock_status      INTEGER;
  release_status INTEGER;
BEGIN
  DBMS_LOCK.ALLOCATE_UNIQUE( 'lock_pro' ,  lock_handle ) ;←― ①
  DBMS_OUTPUT.PUT_LINE( 'START :' ||
                              to_char( sysdate ,  'MI"分"SS"秒"' ) ) ;
  lock_status  :=  DBMS_LOCK.REQUEST( lock_handle , 6 , 900 , TRUE ) ;←― ②
  DBMS_OUTPUT.PUT_LINE( 'LOCK_GET :' ||
                              to_char( sysdate ,  'MI"分"SS"秒"' ) ) ;
  DBMS_LOCK.SLEEP ( 10 ) ; ←― スリープ命令
  release_status  :=  DBMS_LOCK.RELEASE( lock_handle ) ; ←― ③
  DBMS_OUTPUT.PUT_LINE( 'LOCK_RELEASE :' ||
                              to_char( sysdate ,  'MI"分"SS"秒"' ) ) ;
END;

/                  ←―プロシージャの登録のための／（スラッシュ）
```

【ロック処理の実行結果】

［ユーザーA］	［ユーザーB］
SQL> SET SERVEROUTPUT ON	SQL> SET SERVEROUTPUT ON
（メッセージ出力を、SQL*Plus 画面に表示させる）	
SQL> EXECUTE lock_exe	
START 45 分 00 秒	
LOCK_GET 45 分 00 秒	SQL> EXECUTE lock_exe
	START 45 分 01 秒
10 秒間スリープ	
	ロック解放待ち状態
LOCK_RELEASE 45 分 10 秒	
	LOCK_GET 45 分 10 秒
	10 秒間スリープ
	LOCK_RELEASE
	45 分 20 秒