

【楽観的ロック】、【悲観的ロック】

レコードをロックするアプリケーション操作タイミング

更新処理において、対象レコードの排他ロック制御を極めて短くする方法

レコードが更新されているか調査する方法

これは、データの更新を行う直前に更新対象のデータを**再度読取り**、他のトランザクションによって『途中でデータ更新が行われていないか』を確認することにより、データをロックする時間を限りなく短くしつつ、不整合なくデータの更新を行うためのユーザー側のコーディング手法です

【楽観的ロック】

SELECT ***

↓

画面上での入力

↓

SELECT *** **FOR UPDATE**

↓

1回目のSELECTと比較

↓

データが同一

↓ YES

レコード更新

↓

COMMIT

↓ NO

再入力を行わせる

※ レコードに更新が掛かっているかいないかの判断は、レコードの SCN 値を条件にすれば分かる

——— この方法を必要とする状況 ———

- SELECT 文で更新対象のデータを取得した後、UPDATE 文を実行するまでの間の時間が長い処理を行う場合
- 画面入力を行うような更新対象データの制御権を長時間取得したままになるような処理を行う場合

これは、更新中に他のトランザクションによってデータの更新が行なわれないように、事前にロックを行って、更新処理を行うためのユーザー側のコーディング手法です

【悲観的ロック】

SELECT *** **FOR UPDATE**

↓

アプリケーションでの処理（画面上での入力）

↓

レコード更新

↓

COMMIT

——— この方法を必要とする状況 ———

- **SELECT** 文で更新対象のデータを取得した後、**UPDATE** 文を実行するまでの間の時間が短い処理を行う場合
- バッチ処理のように、大量のレコードを更新するトランザクション
- 複数のレコードの処理において、処理全体での一貫性が必要あり、他のトランザクションが行う排他ロック制御によって、1件でもデータ更新の抜けが発生してはいけない場合

レコードが更新されているか調査する方法

調査例 SELECT 文)

```
Select MAX(VERSIONS_STARTSCN) From emp_table
      VERSIONS BETWEEN TIMESTAMP ( SYSTIMESTAMP -
      INTERVAL '15' MINUTE) AND SYSTIMESTAMP
Where empno = 1 ;
```

【 結果出力 】

指定した時間帯にデータ更新が無かった場合には、VERSIONS_STARTSCN
列には、**NULL 値**が戻される

【エラー情報】

ORA-30052

・VERSIONS を使った SELECT 文において、ORA-30052: 下限のスナップシ
ョット式が無効です

→ 指定した時刻のスナップショット情報が古すぎて、スナップシ
ョット・エリアに残っていない
もう少し新しい時間帯を指定し直す
システムの、スナップショット・エリアを拡張することが望まし
い

このエラーが発生した場合は、スナップショットが小さすぎて、処理開始時
のスナップショットが残っていないので、更新されたかどうか判別できない
ということになります

調査例 PL/SQL ロジック文)

```
DECLARE
  FIRST_TIME  DATE;
  INTERVAL_SECOND  NUMBER := 60*60;  -- これが、入力操作を行って
                                         いた時間を想定した指定値

  SECOND_TIME  DATE;
  VERSIONS_STARTSCN_VALUE  NUMBER;
  EMPNO_VALUE  NUMBER;
BEGIN
  FIRST_TIME := SYSDATE;
  EMPNO_VALUE := 1;

  DBMS_OUTPUT.PUT_LINE( '1回目時刻 : ' || TO_CHAR( FIRST_TIME ,
    'YYYY-MM-DD:HH24:MI:SS' ) );
  DBMS_OUTPUT.PUT_LINE( '    入力作業の操作実行 ' );
  SYS.DBMS_LOCK.SLEEP( INTERVAL_SECOND );

  SECOND_TIME := SYSDATE;
  DBMS_OUTPUT.PUT_LINE( '2回目時刻 : ' || TO_CHAR( SECOND_TIME ,
    'YYYY-MM-DD:HH24:MI:SS' ) );
  Select  MAX( VERSIONS_STARTSCN ) INTO
    VERSIONS_STARTSCN_VALUE
  From emp_table
  VERSIONS BETWEEN TIMESTAMP FIRST_TIME AND
  SECOND_TIME
  Where empno = EMPNO_VALUE ;
  DBMS_OUTPUT.PUT_LINE( 'EMPNO 値 : ' || EMPNO_VALUE || 'SCN
  値 : ' || VERSIONS_STARTSCN_VALUE );

  DBMS_OUTPUT.PUT_LINE( '-----' );
  DBMS_OUTPUT.PUT_LINE( '    比較作業 ' );

  IF VERSIONS_STARTSCN_VALUE IS NULL THEN
    DBMS_OUTPUT.PUT_LINE( '! ' );
    DBMS_OUTPUT.PUT_LINE( ' 1回目と2回目の間の対象レコードは同一で
    す' );
  ELSE
    DBMS_OUTPUT.PUT_LINE( '! ' );
    DBMS_OUTPUT.PUT_LINE( ' 1回目と2回目の間に、対象レコードは更新
    されています' );
  END IF ;
```

```
EXCEPTION
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE( 'エラーが発生しました。ORA-' ||
      TO_CHAR( SQLCODE, 'fm00000' ) || ' ' || SQLERRM ) ;
END;

/
```

実行結果)

1 回目時刻 : 2017-10-19:14:28:08

入力作業の操作実行

2 回目時刻 : 2017-10-19:14:28:28

EMPNO 値 : 1 SCN 値 :

比較作業

1 回目と 2 回目の間の対象レコードは同一です

PL/SQL プロシージャが正常に完了しました。

1 回目時刻 : 2017-10-19:14:27:36

入力作業の操作実行

2 回目時刻 : 2017-10-19:14:27:56

EMPNO 値 : 1 SCN 値 : 2786279

比較作業

1 回目と 2 回目の間に、対象レコードは更新されています

PL/SQL プロシージャが正常に完了しました。

1 回目時刻 : 2017-10-19:14:56:26

入力作業の操作実行

2 回目時刻 : 2017-10-19:15:56:26

✓ 入力操作時間が長すぎた場合

エラーが発生しました。ORA--08180 ORA-08180:

指定された時間に基づくスナップショットが見つかりません

PL/SQL プロシージャが正常に完了しました。