ダイレクトロードによるＩＮＳＥＲＴ処理の高速化指定の方法
NOLOGGING によるデータ更新処理の高速化指定の方法

データの「新規作成」や「更新」を**高速で行う**方法

バッファキャシュアクセスを行わないようにして、処理を高速に行うことが出来る（ダイレクトロード）

・ダイレクトロードによるＩＮＳＥＲＴ処理

Redo ログファイルへの書込みを行わないようにする
（NOLOGGING）

・NOLOGGING によるデータ更新処理

ダイレクトロードによるＩＮＳＥＲＴ処理のデメリット

・追加されるレコードは、すべて表の最後尾の空白ブロックに追加される
よって、エクステント途中の未使用で空いているブロック部分は使われないままなので、表のサイズが大きくなる

・処理時に、表全体に排他ロックがかかる
よって、使用中は他のセッションでは「ロック解除待ち」となり、読取り処理が中断される

なお、こちらは、ROLLBACK 処理も可能で、障害発生時の Redo ログファイルからのリカバリ処理においても、データの整合性は保証される

NOLOGGING によるデータ更新処理のデメリット

REDO ログへの「書込み」が行われないので、障害発生時の Redo ログファイルからのリカバリ処理において、データが欠落している
このような場合は、途中データの削除とデータの作成・更新を再度行う必要がある

使用方法

【ダイレクトロードによるＩＮＳＥＲＴ処理】

INSERT 文に、ヒント句で APPEND 指定する

例）

sql> insert　/*+　APPEND　*/　into　＜表名＞　select　・・・　from　・・・　;

※ APPEND ヒント句は、通常の INSERT　VALUES 文では処理されない
into ・・　select　句での INSERT 文のみである
Redo ログへの書き込みは、行われるのでロールバック指示も可能である

【NOLOGGING によるデータ更新処理】

表や索引自体に対して、NOLOGGING 属性を指定する

この状態で以下の操作を行った場合

（通常の INSERT や UPDATE 処理では Redo ログへの書込みは、行われている）

例）

sql> alter　table　＜表名＞　NOLOGGING　;

sql> alter　index　＜表名＞　NOLOGGING　;

※ CREATE 時にも、NOLOGGING 属性を指定することは可能

・APPEND ヒント付き INSERT　SELECT
・DIRECT=true が指定された SQL*Loader によるロード処理
・CREATE　TABLE　NOLOGGING　AS　SELECT 文での処理
・ダイレクトパスモードの Data Pump Import 処理

【APPEND ヒント句を使った使用例】

insert /*+ APPEND */ into kozue.emp_copy select * from kozue.emp;

5 行が作成されました。

select * from kozue.emp_copy;

ORA-12838: オブジェクトは、パラレルで変更された後は読取り/変更できません。

ROLLBACK;

ロールバックが完了しました。

select * from kozue.emp_copy;

レコードが選択されませんでした。

insert /*+ APPEND */ into kozue.emp_copy select * from kozue.emp;

5 行が作成されました。

COMMIT;

コミットが完了しました。

select empno, ename from kozue.emp_copy;

EMPNO	ENAME
1	愛川こずえ
2	いとくとら
3	ミンカ・リー
4	徒然みおれ
5	白咲はつね

※ APPEND ヒント句は、通常の INSERT　VALUES 文では処理されない
into ・・　select　句での INSERT 文のみである
Redo ログへの書き込みは、行われるのでロールバック指示も可能である