

統計情報のバックアップ

ペンディング中の統計情報のエクスポートとインポート方法

ペンディング中の統計情報のテストについて

エクスポートとインポートを行って『オプティマイザ統計情報』の移行を行うと、~~保留中の『オプティマイザ統計情報』も含まれて~~正式な統計情報レコードのみが移行される

なお、正式な統計情報レコードとペンディング中の統計情報をエクスポートするパッケージの**プロシージャ名が異なる**ので、これを使用してペンディング中の統計情報を移行する

テスト環境へ『オプティマイザ統計情報』をインポートするプロシージャは、一つしかない

よって、正式な統計情報かペンディング中の統計情報かは、エクスポート時に決まってしまう

なお、ペンディング状態の統計情報をエクスポートしても、移行の時には**正式に適用された統計情報の状態として**移行される

テスト環境でパフォーマンス改善テストを行う場合には、~~この保留中の『オプティマイザ統計情報』をセッション限定もしくは、正式に統計情報の反映を適用させてから、~~評価テストを行う

本番環境の『オプティマイザ統計情報』をテストする手順について

以下のようなことが、『オプティマイザ統計情報』のテストのために必要かと思われる

- 手順 1. 本番環境で、ペンディング状態で「オプティマイザ統計」を取得する
- 手順 2. 本番環境で、ペンディング状態の「オプティマイザ統計」の情報を保存する
- 手順 3. 本番環境で、「オプティマイザ統計」が保存された表（テーブル）をエクスポートする
- 手順 4. テスト環境で、「オプティマイザ統計」が保存された表（テーブル）をインポートする
- 手順 5. テスト環境で、保存された表（テーブル）から「オプティマイザ統計」を対象オブジェクトにコピーする
注意）テスト環境で「オプティマイザ統計」を対象オブジェクトにコピーを行うと、「オプティマイザ統計」は**正式に適用された状態**になってしまう
- 手順 6. ~~テスト環境で、保留中の「オプティマイザ統計」の反映を適用させる~~
- 手順 7. テスト環境で、確認したいことをテストする
- 手順 8. テスト環境で、「オプティマイザ統計」の効果が確認できたら、
本番環境で、**保留中の「オプティマイザ統計」**の反映を適用させる

【注意】

保留指示が不要になった場合には、保留フラグを **TRUE** に戻すこと
戻し忘れた場合には、スケジュールで取得設定されている統計情報が反映しない

統計情報のエクスポートとインポート方法

(1) stat_table の作成

SQL> begin

```
dbms_stats.create_stat_table (  
    ownname => 'スキーマ名',  
    stattab => '統計情報を保存する表名',  
    tblspace => '表領域名' );  
end ;  
/
```

(2) 統計情報のエクスポート

統計情報を移したい対象オブジェクトの統計情報を (1) で作成した表に保存する

SQL> begin

```
dbms_stats.export_pending_stats (  
    ownname => '調査対象のスキーマ名',  
    tabname => '調査対象の表名',  
    stattab => '統計情報を保存する先の表名',  
    statown => '統計情報のオーナー名' );  
end ;  
/
```

正式適用されている統計情報を移行
する場合には、export_table_stats プ
ロシージャを使用する

(3) stat_table 表のエクスポートとインポート

データパンプを利用して、物理ファイルに「stat_table 表」をエクスポート
テスト環境にて、物理ファイルの中の「stat_table 表」をインポート

(4) stat_table 表から、オブジェクトへ統計情報をインポート

SQL> begin

```
dbms_stats.import_table_stats (  
    ownname => '調査対象のスキーマ名',  
    tabname => '調査対象の表名',  
    stattab => '統計情報が保存されている表名',  
    statown => '統計情報が保存されている表のオーナー名',  
    cascade => TRUE );  
end ;  
/
```

【参考情報】

統計情報の削除方法

テスト例)

- (1) オブジェクトに対して保留（ペンディング）させる指定

```
execute dbms_stats.set_table_prefs('KOZUE', 'EMP', 'PUBLISH',
                                   'FALSE');
```

- (2) オブジェクトに対して統計情報を取得する

```
execute dbms_stats.gather_table_stats('KOZUE', 'EMP');
```

- (3) 統計情報の確認

```
select table_name, blocks , to_char( last_analyzed , 'YYYY/MM/DD
                                   HH24:MI:SS' )
from   dba_tables
where  owner      = 'KOZUE'
and    table_name = 'EMP';
```

- (4) stat_table の作成

```
execute dbms_stats.create_stat_table( -
    ownname => 'KOZUE', -
    stattab  => 'STATS_BACKUP_TABLE', -
    tblspace => 'USERS' );
```

- (5) 本番環境で、ペンディング状態の「オプティマイザ統計」を、stat_table 表へ保存

```
execute dbms_stats.export_pending_stats( -
    ownname  => 'KOZUE', -
    tabname   => 'EMP', -
    statown   => 'KOZUE', -
    stattab   => 'STATS_BACKUP_TABLE' );
select count(*) from KOZUE.STATS_BACKUP_TABLE;
```

- (6) 統計情報の削除

```
execute dbms_stats.delete_table_stats( ownname => 'KOZUE',
                                       tabname => 'EMP' );
```

- (7) 統計情報の確認

```
select table_name, blocks , to_char( last_analyzed , 'YYYY/MM/DD
                                   HH24:MI:SS' )
from   dba_tables
where  owner      = 'KOZUE'
and    table_name = 'EMP';
```

(8) stat_table 表のエクスポート

サーバーのキーボードで、

```
host exp KOZUE/OraTest tables = STATS_BACKUP_TABLE  
file = D:\temp\Exportdata.DMP
```

(9) stat_table 表の削除

```
select count(*) from kozue.STATS_BACKUP_TABLE ;  
drop table kozue.STATS_BACKUP_TABLE ;  
select count(*) from kozue.STATS_BACKUP_TABLE ;
```

(1 0) stat_table 表のインポート

サーバーのキーボードで、

```
host imp KOZUE/OraTest tables = STATS_BACKUP_TABLE  
file = D:\temp\Exportdata.DMP
```

(1 1) stat_table 表から、オブジェクトへ統計情報をインポート

```
select count(*) from kozue.STATS_BACKUP_TABLE ;  
execute dbms_stats.import_table_stats ( -  
    ownname => 'KOZUE' , -  
    tabname => 'EMP' , -  
    statown => 'KOZUE' , -  
    stattab => 'STATS_BACKUP_TABLE' , -  
    cascade => TRUE ) ;
```

(1 2) 統計情報の確認

```
select table_name, blocks , to_char( last_analyzed , 'YYYY/MM/DD  
HH24:MI:SS' )  
from dba_tables  
where owner = 'KOZUE'  
and table_name = 'EMP' ;
```

(1 3) stat_table 表の削除

```
drop table kozue.STATS_BACKUP_TABLE ;
```

【注意】

保留指示が不要になった場合には、保留フラグを **TRUE** に戻すこと
戻し忘れた場合には、スケジュールで取得設定されている統計情報が反映しない