



Hewlett Packard
Enterprise

UNDO 表領域再入門

Noriyoshi Shinoda

April 17, 2025

SPEAKER

- ✓篠田 典良(しのだ のりよし)
- ✓所属
 - ✓日本ヒューレット・パカード合同会社
- ✓プロフィール
 - ✓Oracle ACE Pro (2009～)
 - ✓PostgreSQL 開発 (PostgreSQL 10～17, 18 dev)
 - ✓Oracle Databaseをはじめ PostgreSQL, Microsoft SQL Server, Vertica 等 RDBMS 全般に関するシステムの設計、移行、チューニング、コンサルティング
- ✓関連する URL
 - ✓Redgate 100 in 2022 (Most influential in the database community 2022)
<https://www.red-gate.com/hub/redgate-100/>
 - ✓Oracle ACE Profile
<https://ace.oracle.com/apex/ace/profile/nshino483>
 - ✓「PostgreSQL 虎の巻」シリーズ
<https://github.com/nori-shinoda/documents/blob/main/README.md>



Featured in
The Redgate 100



はじめに

UNDO 表領域再入門

- ✓UNDO 表領域について基本から初期化パラメーターの設定を説明
- ✓Oracle Database 19c Release Update (RU) のインストールによる突然の動作変化



UNDO 表領域の基礎



UNDO 表領域の基礎

UNDO 表領域の用途

✓用途

- ✓基本的には「過去データを参照する」こと
- ✓「過去データの管理」は、主に SMON バックグラウンド・プロセスが主に実行

✓例外

- ✓データベース・リンクを利用した検索に使用するトランザクション・ロック情報
- ✓遅延ブロッククリーンアウトのトランザクション表



UNDO 表領域の基礎

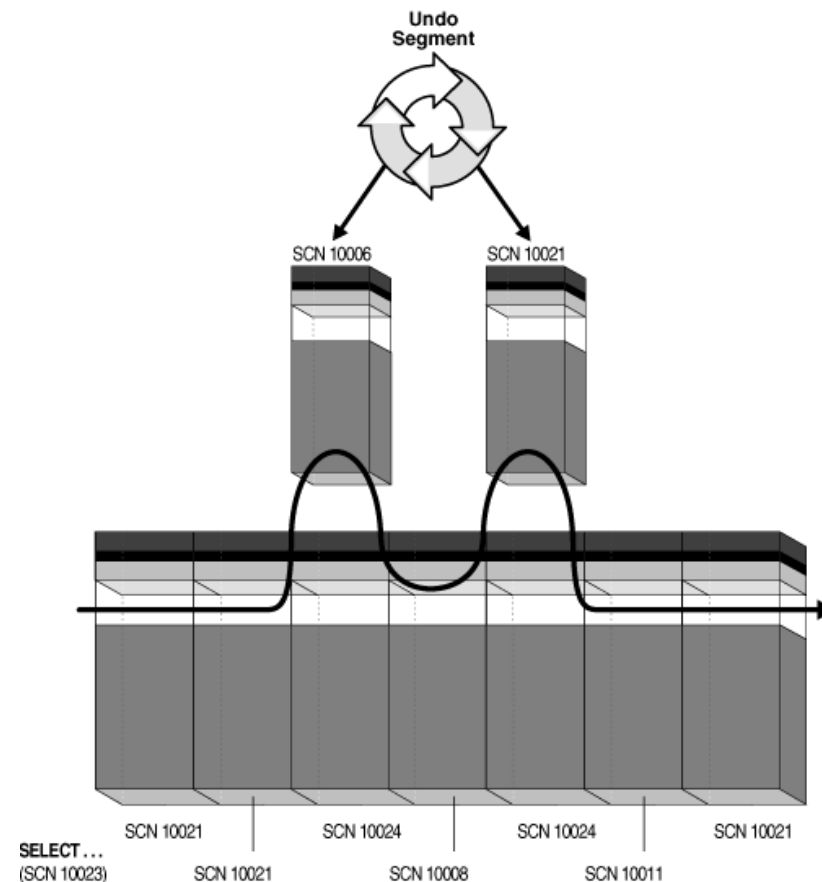
過去データの参照が必要になる場合

- ✓トランザクションのロールバック
 - ✓ROLLBACK 文の実行
 - ✓トランザクションを確定しないままセッション終了からの回復 ⇒ 自動ロールバック
 - ✓インスタンスのクラッシュからの回復 ⇒ 自動ロールバック
- ✓プロセスの動作
 - ✓セッション異常終了 = PMONバックグラウンド・プロセス
 - ✓ロールバック = CLMN/CLnn バックグラウンド・プロセス
 - ✓クラッシュリカバリ = SMON バックグラウンド・プロセス
 - ✓ダーティバッファの書込み = DBWR バックグラウンド・プロセス

UNDO 表領域の基礎

過去データの参照が必要になる場合

- ✓読み取り一貫性の維持
 - ✓SELECT 文開始時の SCN を元にデータが返される
 - ✓古い SCN を持つブロックは UNDO 表領域から取得される
- ✓右図の例
 - ✓SELECT 文開始時の SCN = 10023
 - ✓別のセッションがブロックを更新 SCN = 10024
 - ✓SELECT 文は UNDO 表領域から旧データ(SCN 10006, SCN 10021)を検索



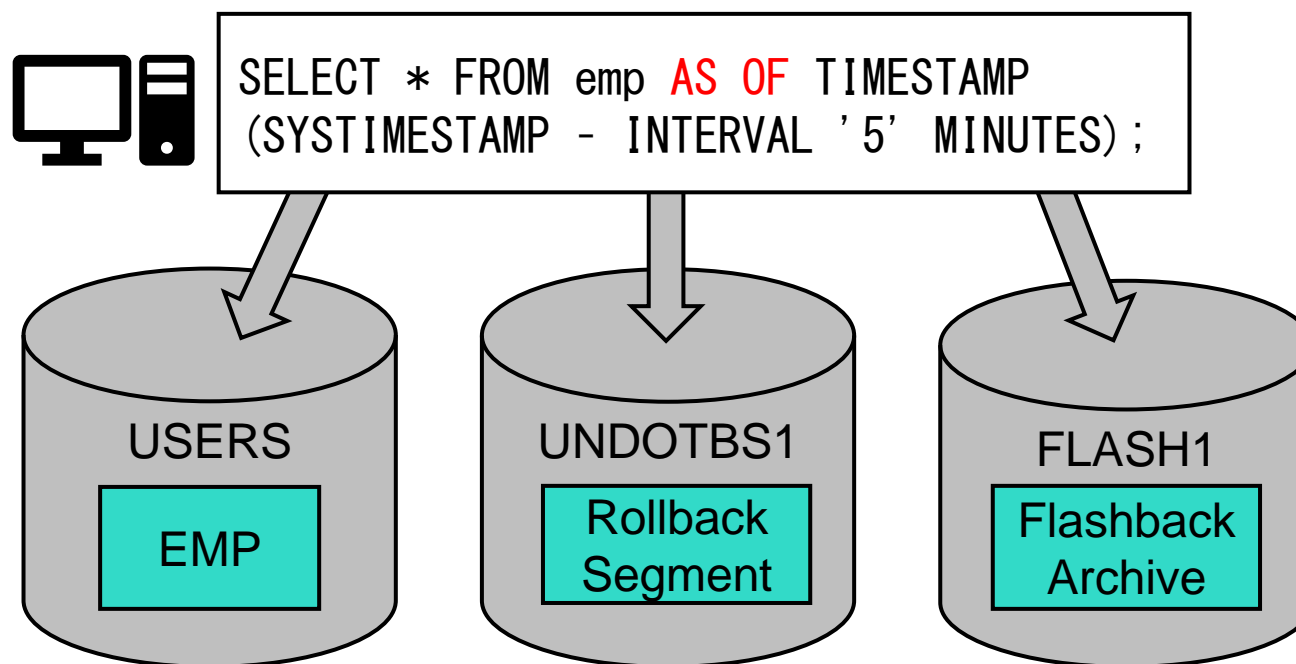
出典: Oracle Database 23ai データベース概要 F73871-08

SCN (System Change Number) = システム変更番号: データベースに対して変更が行われた論理的な時点を示す 48 ビットの番号

UNDO 表領域の基礎

過去データの参照が必要になる場合

- ✓フラッシュバック機能の利用
 - ✓フラッシュバック・クエリー = UNDO レコードを参照
 - ✓フラッシュバック・データ・アーカイブ = UNDO レコードから生成
 - ✓フラッシュバック・ログ = UNDO レコードから生成



UNDO 表領域の基礎

UNDO 表領域が使われる前

- ✓ロールバック・セグメントによる UNDO 管理
 - ✓CREATE ROLLBACK SEGMENT 文によるロールバック・セグメントの手動作成
 - ✓初期化パラメーター rollback_segments に複数のロールバック・セグメントを指定
 - ✓DBA_ROLLBACK_SEGS ビュー、V\$ROLLSTAT ビューで確認
- ✓UNDO 表領域による自動 UNDO 管理に進化
 - ✓ロールバックセグメントの管理を自動化するために専用の表領域を作成(=UNDO 表領域)
 - ✓初期化パラメーター undo_management を AUTO に指定
 - ✓Oracle 9i Release 1 (9.1) から利用可能(undo_management のデフォルト値は MANUAL)
 - ✓デフォルト値が AUTO になるのは Oracle Database 11g Release 1 (11.1) から
- ✓マルチテナント構成における初期化パラメーター undo_management のマニュアル記載

ノート:CDB では、UNDO_MANAGEMENT 初期化パラメータを AUTO に設定してください。UNDO データを管理するには、UNDO 表領域を作成する必要があります。

UNDO 表領域の基礎

UNDO 表領域が使われる前

✓ロールバック・セグメントが無くなったわけではない

```
SQL> SELECT segment_name, owner, tablespace_name, status FROM dba_rollback_segs;
```

SEGMENT_NAME	OWNER	TABLESPACE_NAME	STATUS
SYSTEM	SYS	SYSTEM	ONLINE
_SYSSMU1_1261223759\$	PUBLIC	UNDOTBS1	ONLINE
_SYSSMU2_27624015\$	PUBLIC	UNDOTBS1	ONLINE
_SYSSMU3_2421748942\$	PUBLIC	UNDOTBS1	ONLINE
_SYSSMU4_625702278\$	PUBLIC	UNDOTBS1	ONLINE
...			

✓自動 UNDO 管理ができるようになった理由

- ✓ローカル管理表領域 (EXTENT MANAGEMENT LOCAL) によってエクステント作成の負荷が下がったから
- ✓ローカル管理表領域は Oracle 8i から利用可能

UNDO 表領域の利用



UNDO 表領域の利用

UNDO 表領域の作成

✓CREATE UNDO TABLESPACE 文を実行

```
SQL> CREATE UNDO TABLESPACE undotbs2 DATAFILE '+DATA' SIZE 100G;
```

Tablespace created.

```
SQL> SELECT tablespace_name, contents, retention FROM DBA_TABLESPACES  
       WHERE tablespace_name='UNDOTBS2';
```

TABLESPACE_NAME	CONTENTS	RETENTION
UNDOTBS2	UNDO	NOGUARANTEE

UNDO 表領域の利用

UNDO 表領域の作成

✓CREATE UNDO TABLESPACE 文のオプション

属性	設定	備考
EXTENT MANAGEMENT	LOCAL AUTOALLOCATE 固定	UNIFORM 設定不可
SEGMENT SPACE MANAGEMENT	指定不可	MANUAL 固定
FLASHBACK [ON OFF]	無視される	
FORCE LOGGING	指定不可	
[NO]LOGGING	指定不可	
BIGFILE SMALLFILE	任意	
BLOCKSIZE	任意	
RETENTION [NO]GUARANTEE	デフォルト NOGUARANTEE	UNDO 表領域独自属性(後述)



UNDO 表領域の利用

UNDO 表領域の作成

- ✓マルチテナント構成の場合
 - ✓UNDO 表領域は CDB のみに作成する(12.1 ~)
 - ✓UNDO 表領域は PDB 単位に作成することができる(12.2 ~)
- ✓CDB 作成時の CREATE DATABASE 文の「LOCAL UNDO」句
 - ✓LOCAL UNDO ON 句の指定で、PDB 単位の UNDO 表領域をサポート(デフォルト LOCAL UNDO OFF)

```
CREATE DATABASE "019A" MAXINSTANCES 8
```

```
... 途中省略 ...
```

```
USER SYS IDENTIFIED BY "sysPassword" USER SYSTEM IDENTIFIED BY "systemPassword"
```

```
ENABLE PLUGGABLE DATABASE SEED file_name_convert=(...)
```

```
LOCAL UNDO ON;
```

UNDO 表領域の利用

UNDO 表領域の作成

✓Local Undo を使っているかの確認方法

```
SQL> SELECT PROPERTY_VALUE FROM DATABASE_PROPERTIES WHERE  
        PROPERTY_NAME = 'LOCAL_UNDO_ENABLED';
```

```
PROPERTY_VALUE
```

```
-----  
TRUE
```

✓Local Undo の変更方法

```
SQL> STARTUP UPGRADE
```

```
SQL> ALTER DATABASE LOCAL UNDO [ON | OFF];
```

```
SQL> SHUTDOWN IMMEDIATE
```

```
SQL> STARTUP
```

UNDO 表領域の利用

利用のための初期化パラメーター

- ✓ 初期化パラメーター undo_tablespace
 - ✓ インスタンスが使用する UNDO 表領域名を指定する
 - ✓ 指定しない場合は使用可能な UNDO 表領域が自動的に設定される
 - ✓ RAC / RAC One Node 環境ではインスタンス単位に独立した UNDO 表領域を指定する
- ✓ 複数インスタンスで同一 UNDO 表領域を使おうとした場合

```
SQL> ALTER SYSTEM SET undo_tablespace = UNDOTBS1;
```

```
ALTER SYSTEM SET undo_tablespace = UNDOTBS1
```

```
*
```

```
ERROR at line 1:
```

```
ORA-02097: parameter cannot be modified because specified value is invalid
```

```
ORA-30013: undo tablespace 'UNDOTBS1' is currently in use
```


UNDO 表領域の利用

利用のための初期化パラメーター

✓初期化パラメーター undo_retention

✓UNDO 表領域に古い UNDO 情報を維持するための秒数(の目安)

✓マニュアルのパラメーター説明

✓12.1 まで

UNDO_RETENTIONには、UNDO 保存の下限値(秒)を指定します。AUTOEXTEND UNDO 表領域の場合、UNDO は、このパラメータに指定した時間以上に保存され、問合せに対する UNDO 要件にあわせて UNDO 保存期間が自動的にチューニングされます。

✓12.2 以降

通常は、UNDO_RETENTION をデフォルト値に設定しておくことをお勧めします。このパラメータの変更は、次の状況でのみお勧めします。

- Oracle Flashback Query などのフラッシュバック機能を使用する場合は、システムで最も長く実行されている問合せよりも長く UNDO を保存する必要があるため、UNDO_RETENTION の値を大きくすることがあります。
- Oracle Active Data Guard 環境では、スタンバイ・インスタンスでの UNDO 保存要件に対応するために、プライマリ・インスタンスでの UNDO_RETENTION の値を大きくすることがあります。これにより、プライマリ・インスタンスで、スタンバイ・インスタンスで問合せを処理する時間より長く UNDO を保存できます。

UNDO 表領域の利用

利用のための初期化パラメーター

- ✓ 初期化パラメーター undo_retention 設定値の変遷
 - ✓ RU 19.6 以前
 - ✓ 表領域データファイルが固定サイズ (AUTOEXTEND OFF) の場合、undo_retention に指定した値は無視される
 - ✓ RU 19.7 以降
 - ✓ 表領域が固定サイズでも初期化パラメーター undo_retention は考慮される
 - ✓ RU 19.8 以前
 - ✓ CDB root の設定変更が PDB に伝播する
 - ✓ RU 19.9 以降
 - ✓ CDB root の設定が PDB に伝播しない

UNDO 表領域の利用

利用のための初期化パラメーター

- ✓ 実際の UNDO 保持期間
 - ✓ 実際の UNDO 保持期限は **Tuned Undo Retention** と呼ばれる値(10.1～)
 - ✓ 初期化パラメーター undo_retention は原則として **UNDO 保持期限の下限**になる
 - ✓ UNDO 表領域に空きがあるのであれば undo_retention を超えても UNDO データを保持し、ORA-01555 エラーの発生を防ぐ
 - ✓ 隠しパラメーター _undo_autotune (デフォルト TRUE)で制御される
 - ✓ 隠しパラメーター _highthreshold_undoretention (デフォルト 31536000)が最大値
- ✓ UNDO 表領域の RETENTION 属性による Tuned Undo Retention 値
 - ✓ NOGUARANTEE = UNDO 表領域の空き容量によっては undo_retention 以下になる(デフォルト)
 - ✓ GUARANTEE = UNDO 表領域の空き容量関係無く undo_retention が下限
- ✓ データファイルのサイズ
 - ✓ できるだけ undo_retention で指定された秒数を維持しようとするが、足りない場合はデータファイルを拡大する
 - ✓ 固定サイズの UNDO 表領域の場合は undo_retention 設定値を維持できない場合がある

UNDO 表領域の利用

利用のための初期化パラメーター

- ✓V\$UNDOSTAT ビュー(または DBA_HIST_UNDOSTAT ビュー)を確認
 - ✓TUNED_UNDORETENTION 列 = Tuned Undo Retention 計算値
 - ✓MAXQUERYLEN 列 = インスタンスで実行された最も長い問合せの長さ(秒)
 - ✓UNXPBLKREUCNT 列 = トランザクションで再利用された **期限切れ前の UNDO ブロック** 数

```
SQL> SELECT begin_time, tuned_undoretention, maxquerylen, unxpblkreucnt FROM
        V$UNDOSTAT ORDER BY begin_time;
```

BEGIN_TI	TUNED_UNDORETENTION	MAXQUERYLEN	UNXPBLKREUCNT
13:11:56	900	0	0
13:21:56	1222	379	0
13:31:56	1833	989	0
...			

UNDO 表領域の利用

利用のための初期化パラメーター

- ✓ 初期化パラメーター temp_undo_enabled
 - ✓ 一時表 (TEMPORARY TABLE) の UNDO データを一時表領域に保存 (デフォルト値 FALSE)
 - ✓ 一時表を使う場合には TRUE に変更を推奨



UNDO 表領域の利用

LOB 型列の RETENTION

- ✓LOB 型の対応
 - ✓アウトライン LOB (4,000 バイトを超える LOB) は独自の RETENTION 設定となる
 - ✓UNDO 表領域ではなく、LOB セグメント上で管理される
- ✓RETENTION 設定

設定	説明	備考
RETENTION AUTO	初期化パラメーター undo_retention を使用	SecureFiles (デフォルト)
RETENTION MIN sec	指定された秒数以上保持	SecureFiles
RETENTION MAX	MAXSIZE 指定値まで保持	SecureFiles
RETENTION NONE	使用しない	SecureFiles
RETENTION	初期化パラメーター undo_retention を使用	BasicFiles



UNDO 表領域の利用

LOB 型列の RETENTION

✓ 設定例と確認

```
SQL> CREATE TABLE data1 (c1 NUMBER, c2 BLOB)
      LOB(c2) STORE AS SECUREFILE c2lob (TABLESPACE users RETENTION AUTO);
```

Table created.

```
SQL> SELECT retention_type, retention_value FROM user_lobs
      WHERE table_name='DATA1' AND column_name='C2';
```

RETENTI	RETENTION_VALUE
AUTO	

発生する可能性があるトラブル



発生する可能性があるトラブル

ORA-30036 / ORA-01650

- ✓ UNDO 表領域領域不足

- ✓ ORA-30036: unable to extend segment by {#id} in undo tablespace {tablespace_name}

- ✓ ORA-01650: unable to extend rollback segment {name#1} {name#2} in tablespace {tablespace_name}

- ✓ 更新 DML 自体がエラーになる

- ✓ 必要な対応

- ✓ UNDO 表領域を拡大する(データファイルの追加または拡大)

- ✓ 例

```
SQL> DELETE FROM data1;
```

```
DELETE FROM data1
```

```
      *
```

```
ERROR at line 1:
```

```
ORA-30036: unable to extend segment by 8 in undo tablespace 'UNDOTBS2'
```

発生する可能性があるトラブル

ORA-01555

- ✓ UNDO データの参照エラー
 - ✓ ORA-01555: snapshot too old: rollback segment number {#id} with name {name} too small
 - ✓ 必要な UNDO レコードが上書きされている
 - ✓ 更新トランザクションにエラーは発生しないが、過去 SCN を参照する検索にエラーが発生
- ✓ 必要な対応
 - ✓ UNDO 表領域の拡大
 - ✓ 初期化パラメーター undo_retention の拡大
 - ✓ UNDO 表領域の RETENTION GUARANTEE 設定
- ✓ 例

```
SQL> SELECT count(*) FROM data1;
```

```
SELECT count(*) FROM data1
```

```
          *
```

```
ERROR at line 1:
```

```
ORA-01555: snapshot too old: rollback segment number 23 with name
```

```
"_SYSSMU23_2432033955$" too small
```

発生する可能性があるトラブル

ORA-30027

- ✓許可された UNDO データ量リソース不足
 - ✓ORA-30027: Undo quota violation - failed to get # (bytes)
 - ✓リソースマネージャで許可されたコミットされていない UNDO データ容量(**UNDO_POOL**)が不足
 - ✓更新 DML 自体がエラーになる
- ✓必要な対応
 - ✓リソース・マネージャのプラン・ディレクティブを拡大
- ✓例

```
SQL> UPDATE data1 SET col2='update#1';  
UPDATE data1 SET col2='update#1'  
      *  
ERROR at line 1:  
ORA-30027: Undo quota violation - failed to get 68 (bytes)
```

発生する可能性があるトラブル

Data Pump のオプションによるトラブル

- ✓ Data Pump Export **CONSISTENT=Y** オプション
 - ✓ デフォルト (CONSISTENT=N) ではテーブル単位にトランザクションが発生
 - ✓ CONSISTENT=Y を指定すると 内部的に SET TRANSACTION READ ONLY 文が実行される
 - ⇒ トランザクションの時間が長くなり UNDO データが解放されない
- ✓ Data Pump Import **QUERY** オプション
 - ✓ ダイレクト・パス・ロードが使用されず、外部表が使用される
 - ⇒ UNDO データが増加する

最適値を探す



最適値を探す

サイズの指標

✓DBMS_UNDO_ADV パッケージ

関数	説明
BEST_POSSIBLE_RETENTION	undo_retention の最適な値を返す
LONGEST_QUERY	指定期間中の最長クエリ時間を返す
RBU_MIGRATION	自動 UNDO 管理に必要な容量を返す
REQUIRED_RETENTION	指定期間中の UNDO 統計から最適な undo_retention を返す
REQUIRED_UNDO_SIZE	指定期間中の UNDO 統計から最適なサイズを返す
UNDO_ADVISOR	アドバイザと推奨設定をレポート
UNDO_AUTOTUNE	自動チューニングが有効か
UNDO_HEALTH	指定期間中の UNDO 統計から設定の問題点を指摘
UNDO_INFO	現行 UNDO 設定の情報を返す

最適値を探す

サイズの指標

✓実行例

```
SQL> SELECT DBMS_UNDO_ADV.REQUIRED_UNDO_SIZE (3600, SYSDATE -1, SYSDATE)  
        AS recommended_size;
```

```
RECOMMENDED_SIZE  
-----  
                1891
```

最適値を探す

サイズの指標

✓AWR Report

Undo Segment Summary

- Min/Max TR (mins) - Min and Max Tuned Retention (minutes)
- STO - Snapshot Too Old count, OOS - Out of Space count
- Undo segment block stats:
- uS - unexpired Stolen, uR - unexpired Released, uU - unexpired reUsed
- eS - expired Stolen, eR - expired Released, eU - expired reUsed

Undo TS#	Num Undo Blocks (K)	Number of Transactions	Max Qry Len (s)	Max Tx Concurrency	Min/Max TR (mins)	STO/ OOS	uS/uR/uU/ eS/eR/eU
2	0.61	361	1,225	4	31.3/41.4	0/0	0/0/0/0/0/784

[Back to Undo Statistics](#)
[Back to Top](#)

Undo Segment Stats

- Most recent 35 Undostat rows, ordered by Time desc

End Time	Num Undo Blocks	Number of Transactions	Max Qry Len (s)	Max Tx Concy	Tun Ret (mins)	STO/ OOS	uS/uR/uU/ eS/eR/eU
25-Sep 12:50	127	38	1,225	2	41	0/0	0/0/0/0/0/392
25-Sep 12:40	136	37	621	3	41	0/0	0/0/0/0/0/128
25-Sep 12:30	348	286	143	4	31	0/0	0/0/0/0/0/264



その他



その他

隠しパラメーター

✓UNDO 表領域に通常のテーブル作成

```
SQL> CREATE TABLE data1(c1 NUMBER, c2 VARCHAR2(10)) TABLESPACE undotbs1;  
CREATE TABLE data1(c1 NUMBER, c2 VARCHAR2(10)) TABLESPACE undotbs1
```

*

ERROR at line 1:

ORA-30022: Cannot create segments in undo tablespace

```
SQL> ALTER SYSTEM SET "_undotbs_regular_tables" = TRUE;
```

System altered.

```
SQL> CREATE TABLE data1(c1 NUMBER, c2 VARCHAR2(10)) TABLESPACE undotbs1;
```

Table created.

その他

参考

✓参考となる情報

- ✓Undo Related Wait Events & Known Issues (Doc ID 1575701.1)

https://support.oracle.com/knowledge/Oracle%20Database%20Products/1575701_1.html

- ✓UNDO保存期間の自動チューニングの問題 (Doc ID 2374515.1)

https://support.oracle.com/knowledge/Oracle%20Database%20Products/2374515_1.html

- ✓Solving UNDO Corruption (Doc ID 1950230.1)

https://support.oracle.com/knowledge/Oracle%20Database%20Products/1950230_1.html

- ✓遅延ブロッククリーンアウトの性能対策/JPOUG Tech Talk Night #8

<https://speakerdeck.com/takahashikzhr/jpoug-tech-talk-night-number-8>

まとめ



まとめ

- ✓ UNDO 表領域の基本的な動作を確認しました
 - ✓ 基本的には大き目にとっておけば問題は発生しにくい
 - ✓ DBA_HIST_UNDOSTAT ビューは定期的を確認しましょう

THANK YOU

Mail : noriyoshi.shinoda@hpe.com

X(Twitter) : @nori_shinoda

Qiita : @plusultra

