UNDO i TEMP

Poziom: operacyjny **Dotyczy: Oracle Database 12c–19c** **Czas: ~20 min**

Dwa specjalne tablespace, które nie trzymają danych aplikacji, tylko obsługują mechanikę silnika — UNDO pod wycofanie transakcji i wielowersyjność, TEMP pod sortowania, które nie zmieściły się w pamięci. Oba potrafią się zapchać w zupełnie inny sposób niż zwykły tablespace.

Skróty użyte na tej stronie (1)

Skrót	Co znaczy
PGA	Program Global Area: prywatna pamięć procesu serwera Oracle

Fiszka

TYLKO ODCZYT

Sprawdzenie zajętości UNDO wg statusu extentów

```
SELECT tablespace_name, status, SUM(blocks)*8192/1024/1024/1024 AS gb
FROM dba_undo_extents
GROUP BY tablespace_name, status;
```

Sprawdzamy, ile miejsca w tablespace UNDO zajmują extenty w poszczególnych stanach — `ACTIVE` (aktualnie potrzebne żywym transakcjom), `UNEXPIRED` (jeszcze w oknie `UNDO_RETENTION`, przydatne do zapytań flashback) i `EXPIRED` (już niepotrzebne, gotowe do ponownego użycia).

Przykład:

```
TABLESPACE_NAME  STATUS          GB
---------------- --------- ---------
UNDOTBS1         ACTIVE         .42
UNDOTBS1         UNEXPIRED     1.15
UNDOTBS1         EXPIRED       3.68
```

⚠ ZMIENIA SYSTEM

Regeneracja tablespace UNDO: utworzenie nowego i przełączenie

Zanim zaczniemy: regenerację robimy, kiedy stary tablespace UNDO urósł ponad rozsądny rozmiar (zwykle po jednorazowym szczycie transakcyjnym) i chcemy odzyskać miejsce na dysku, zamiast czekać, aż same extenty się zwolnią. Baza cały czas pozostaje online — operacja jest w pełni bezobsługowa dla aplikacji, poza chwilowym większym obciążeniem podczas przełączania.

Tworzymy nowy tablespace UNDO obok istniejącego:

```
CREATE UNDO TABLESPACE NazwaTablespace DATAFILE 'SciezkaDoPliku' SIZE RozmiarMB;
```

Przykład:

```
CREATE UNDO TABLESPACE UNDOTBS2 DATAFILE '/oracle/db/wmstest/oradata/undotbs02.dbf' SIZE
1024M;
```

```
Tablespace created.
```

Przełączamy instancję na nowy tablespace parametrem `UNDO_TABLESPACE`:

```
ALTER SYSTEM SET undo_tablespace='NazwaTablespace' SCOPE=BOTH;
```

Przykład:

```
ALTER SYSTEM SET undo_tablespace='UNDOTBS2' SCOPE=BOTH;
```

```
System altered.
```

Od tego momentu wszystkie nowe transakcje piszą undo do nowego tablespace, a stary trzyma jeszcze dane starszych, wciąż aktywnych transakcji — dopiero kiedy się od nich uwolni, można go bezpiecznie usunąć:

```
DROP TABLESPACE NazwaTablespace INCLUDING CONTENTS AND DATAFILES;
```

Przykład:

```
DROP TABLESPACE UNDOTBS1 INCLUDING CONTENTS AND DATAFILES;
```

```
Tablespace dropped.
```

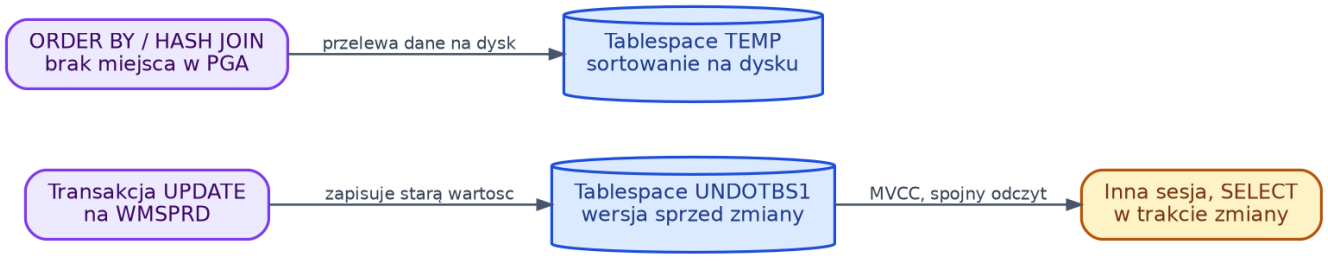
Jeżeli `DROP TABLESPACE` odmówi z powodu wciąż aktywnych extentów w starym tablespace, odczekujemy chwilę i sprawdzamy ponownie fiszką zajętości UNDO wyżej — extenty muszą przejść w stan `EXPIRED`, zanim miejsce faktycznie się zwolni.

SET = Switch Embedded Teaming: zespół kart sieciowych zbudowany w przełączniku wirtualnym Hyper-V; łączy dwie karty w jedno łącze bez ustawień po stronie przełącznika fizycznego, więc bez LACP

Czym jest segment wycofania i po co istnieje

Pojęcie: segment wycofania (undo). Miejsce, w którym Oracle zapisuje poprzednią wersję zmienianych danych, zanim nadpisze je nową wartością. Kiedy transakcja robi `ROLLBACK`, silnik odtwarza stan sprzed zmiany właśnie z segmentu wycofania. To samo undo napędza mechanizm wielowersyjności — czytający proces widzi dane w takim stanie, w jakim były w chwili rozpoczęcia jego zapytania, nawet jeśli w międzyczasie ktoś inny je zmienia, ponieważ Oracle w razie potrzeby sięga po starszą wersję wiersza właśnie z UNDO.

Mechanizm ten szerzej opisuje strona MVCC: dlaczego czytający nie blokuje piszącego.



Transakcja zapisuje starą wartość wiersza do UNDO, dzięki czemu inna sesja czyta spójny stan sprzed zmiany, a osobno sortowanie bez miejsca w PGA przelewa dane do TEMP (kliknięcie otwiera pełny rozmiar)

UNDO leży w specjalnym tablespace, zwykle nazwanym `UNDOTBS1`, wskazanym parametrem `UNDO_TABLESPACE`. W przeciwieństwie do zwykłego tablespace danych, jego zajętość jest z natury zmienna w czasie — rośnie wraz z długimi, aktywnymi transakcjami i kurczy się, kiedy dane

wycofania przestają być komukolwiek potrzebne.

UNDO_RETENTION i ORA-01555

> **Pojęcie: UNDO_RETENTION.** Parametr określający, przez ile sekund Oracle stara się zachować dane wycofania po zakończeniu transakcji, zamiast od razu nadpisywać je nowszymi danymi — im dłuższe okno, tym dłużej działają zapytania flashback i tym mniejsze ryzyko błędu przy długo trwających zapytaniach na spójnych danych.

Błąd ORA-01555 (snapshot too old) pojawia się, kiedy długo działające zapytanie próbuje odczytać starszą wersję wiersza, a odpowiadające jej dane wycofania zostały już nadpisane przez inne transakcje, zanim zapytanie zdążyło po nie sięgnąć. Najczęstsza przyczyna to zbyt mały tablespace UNDO albo zbyt krótki `UNDO_RETENTION` w stosunku do czasu trwania najdłuższych zapytań w bazie — rozwiązaniem jest zwykle powiększenie UNDO albo wydłużenie retencji, rzadziej skrócenie samego zapytania.

Czym jest TEMP

> **Pojęcie: tablespace TEMP.** Przestrzeń na dane tymczasowe — sortowania (`ORDER BY`, `GROUP BY`), operacje `HASH JOIN` i inne struktury pośrednie, które nie zmieściły się w przydzielonej pamięci PGA. W przeciwieństwie do zwykłego tablespace, miejsce zwolnione przez zakończone sortowanie od razu wraca do puli wolnego miejsca w tym samym pliku, więc TEMP praktycznie nigdy nie rośnie trwale — chyba że jedna operacja jest naprawdę ogromna albo coś trzyma sesję otwartą z niedokończonym sortowaniem.

Sprawdzenie sesji zajmujących TEMP

```
SELECT s.sid, s.serial#, s.username, u.tablespace, ROUND(u.blocks*8/1024,2) AS mb
FROM v$session s
JOIN v$tempseg_usage u ON u.session_addr = s.saddr
ORDER BY u.blocks DESC;
```

Sprawdzamy, która konkretnie sesja aktualnie zajmuje TEMP i ile — przydaje się, kiedy zapytanie z fiszki o zajętości TEMP na stronie Tablespace: sprawdzenie, powiększenie, utworzenie, usunięcie pokazuje wysokie zużycie, a szukamy winnego zapytania.

Przykład:

```
   SID  SERIAL# USERNAME   TABLESPACE           MB
------- -------- ---------- ------------ -----------
    412     8821 WMS_APP     TEMP               1240
    118     2054 WMS_ADMIN   TEMP                310
```

Odtworzenie tablespace TEMP

Zanim zaczniemy: odtwarzamy TEMP od zera, kiedy plik tymczasowy uległ uszkodzeniu albo kiedy chcemy zmienić jego rozmiar bazowy bez zostawiania rozdętego pliku. Tempfile nie jest częścią backupu bazy w klasycznym sensie — operacja jest więc czysto logiczna, bez odtwarzania z kopii zapasowej.

Tworzymy nowy tablespace tymczasowy obok istniejącego:

```
CREATE TEMPORARY TABLESPACE NazwaTablespace TEMPFILE 'SciezkaDoPliku' SIZE RozmiarMB
AUTOEXTEND ON NEXT 500M MAXSIZE 8192M;
```

Przełączamy go na domyślny tablespace tymczasowy bazy:

```
ALTER DATABASE DEFAULT TEMPORARY TABLESPACE NazwaTablespace;
```

Usuwamy stary tablespace TEMP razem z plikiem:

```
DROP TABLESPACE NazwaTablespace INCLUDING CONTENTS AND DATAFILES;
```

Przykład:

```
Tablespace created.


Database altered.


Tablespace dropped.
```

Z życia wzięte

Z ŻYCIA WZIĘTE

W środku miesiąca Marek zgłasza, że raport rozliczenia magazynu urywa się błędem po czterdziestu minutach działania. Tomek sprawdza `DBA_UNDO_EXTENTS` na WMSPRD i widzi, że tablespace UNDOTBS1 jest zbyt mały wobec długości tego zapytania, więc starsze dane wycofania zostały nadpisane, zanim raport zdążył po nie sięgnąć. Po podniesieniu `UNDO_RETENTION` i dołożeniu miejsca do UNDO ten sam raport kończy się bez błędu ORA-01555.

Częste błędy

Próba dropnięcia UNDO, który wciąż jest wskazany parametrem UNDO_TABLESPACE. Baza odmówi — najpierw trzeba przełączyć instancję na inny tablespace UNDO, dopiero potem można usunąć poprzedni.

Traktowanie wysokiej zajętości TEMP jak wycieku. Zanim zaczniemy powiększać TEMP, warto sprawdzić fiszką sesji, czy to nie jest jedno bardzo duże, ale krótkotrwałe zapytanie — miejsce samo wróci do puli po jego zakończeniu.

Za krótki UNDO_RETENTION przy długich raportach. Baza domyślnie próbuje utrzymać retencję automatycznie na podstawie historii obciążenia, ale przy nagłym skoku długości zapytań i tak można trafić na ORA-01555 — w takiej sytuacji retencję warto podnieść ręcznie, a nie tylko powiększać UNDO.

Powiązane: Tablespace: sprawdzenie, powiększenie, utworzenie, usunięcie · MVCC: dlaczego czytający nie blokuje piszącego · Pliki danych: konfiguracja, przenoszenie, zmiana nazwy

Źródła: notatki własne, dokumentacja Oracle Database (Administrator's Guide — Managing Undo, Managing Temporary Tablespaces)

Revision #42
Created 2026-09-24 23:34:27 UTC by Tomasz Konieczny
Updated 2026-10-26 14:08:59 UTC by Tomasz Konieczny