

**Вид заняття:** групове

**Навчальна дисципліна:** Проектування інформаційних систем

**Тема № 8** Моделі даних, моделі процесів та їх проектування за допомогою ERwin

**Заняття № 3** Налаштування параметрів пам'яті бази даних

**Навчальні та виховні цілі:** Викласти основні технологічні підходи щодо налаштування параметрів пам'яті бази даних в СКБД корпоративного призначення ORACLE з урахуванням специфіки виконання завдань у зоні проведення антитерористичної операції (АТО). Розвинути інженерне мислення майбутніх інженерів-математиків у галузі комп'ютерної інженерії та сучасних технологій побудови програмного забезпечення

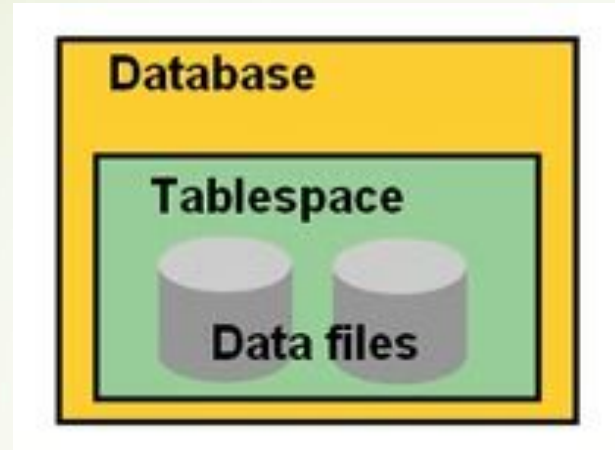
## **Рекомендована література**

1. Глушаков С.В., Третьяков Ю.В., Головаш О.А.,  
Администрирование Oracle 9i. – Харьков: Фолио,  
2003. – 695 с. – Учебный курс

## **Навчальні питання:**

1. Управління розміром і файлами БД. Політика управління
2. Сегментація і створення табличного простору і файлів даних. Додавання файлів даних до табличного простору
3. Установка параметрів пам'яті для табличних просторів. Перейменування та переміщення файлів даних. Видалення табличних просторів та файлів даних.

# Налаштування параметрів пам'яті бази даних



## Налаштування параметрів пам'яті бази даних

*Використання множинних табличних просторів дозволить найгнучкіше організувати роботу з базою даних. Така організація зберігання даних надає наступні можливості:*

| відокремлювати призначені для користувача дані від даних словника;

| відокремлювати дані одних застосувань від даних інших застосувань;

| розміщувати файли даних різних табличних просторів на різних дисках, щоб зменшити суперництво;

| відокремлювати дані сегментів відкоту від призначених для користувача даних, запобігаючи втраті інформації в результаті одиничного збою;

| переводити індивідуальні табличні простори в неактивний стан, продовжуючи нормальну роботу з іншими табличними просторами;

| резервувати табличні простори для спеціальних режимів використання бази даних, наприклад, для періодів високої активності, тільки для читання, або для розміщення тимчасових сегментів;

| здійснювати резервне копіювання індивідуальних табличних просторів захищаючи решту просторів доступними для користувачів

## Налаштування параметрів пам'яті бази даних

*Табличні простори поділяються на декілька категорій:*

- системні табличні простори (обов'язкове), створюються на момент створення бази даних (SYSTEM), як правило містять у собі словник даних;
- несистемні табличні простори (користувальницькі), використовуються користувачами для збереження персональних даних.

Наступний скрипт виводить список табличних просторів в базі даних:

```
SELECT tablespace_name, status  
FROM dba_tablespaces  
ORDER BY tablespace_name;
```

Використання сегментованих табличних просторів дає ряд переваг для роботи СУБД:

- | підвищується продуктивність бази даних;
- | підвищується доступність даних;
- | спрощується резервне копіювання і скорочується час відновлення. Можна виконувати резервне копіювання розділів в оперативному режимі окремо і лише у разі їх зміни. Якщо буде потрібно відновлення, незачеплені розділи залишаються доступними і оперативними;
- | спрощується адміністрування і управління.

## Налаштування параметрів пам'яті бази даних

```
CREATE TABLESPACE userdata  
DATAFILE '/u01/oradata/userdata01.dbf' SIZE 5M;
```

```
CREATE TABLESPACE userdata  
DATAFILE '/u01/oradata/userdata01.dbf' SIZE 500M  
EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K;
```

## Налаштування параметрів пам'яті бази даних

Приведений нижче приклад створює табличний простір з ім'ям NewTSpace з наступними характеристиками:

| дані нового табличного простору міститимуться в єдиному файлі даних, розміром 1Мб;

| замовчувані параметри пам'яті для сегментів, які створюватимуться в цьому табличному просторі, задані явно;

| після створення табличного простору воно залишається в неактивному стані.

```
CREATE TABLESPACE NewTSpace
DATAFILE 'dANewTSpacel.dbf' SIZE 1M
DEFAULT STORAGE (
INITIAL 50K
NEXT 50K
MINEXTENTS 2
MAXEXTENTS 50
PCTINCREASE 0)
OFFLINE;
```



## Налаштування параметрів пам'яті бази даних

```
DBMS_SPACE_ADMIN.
```

```
TABLESPACE_MIGRATE_TO_LOCAL ('SYSTEM') ;
```

```
CREATE UNDO TABLESPACE undo1
```

```
DATAFILE '/u01/oradata/undo01.dbf' SIZE 40M;
```

```
CREATE TEMPORARY TABLESPACE temp
```

```
TEMPFILE '/u01/oradata/temp01.dbf' SIZE 20M
```

```
EXTENT MANAGEMENT LOCAL UNIFORM SIZE 4M;
```

## Налаштування параметрів пам'яті бази даних

```
CREATE DATABASE DBA01
LOGFILE
GROUP 1 ('/$HOME/ORADATA/u01/redo01.log') SIZE 100M,
GROUP 2 ('/$HOME/ORADATA/u02/redo02.log') SIZE 100M,
MAXLOGFILES 5
MAXLOGMEMBERS 5
MAXLOGHISTORY 1
MAXDATAFILES 100
MAXINSTANCES 1
DATAFILE '$HOME/ORADATA/u01/system01.dbf' SIZE 325M
UNDO TABLESPACE undotbs
DATAFILE '$HOME/ORADATA/u02/undotbs01.dbf' SIZE 200
DEFAULT TEMPORARY TABLESPACE temp
TEMPFILE '$HOME/ORADATA/u03/temp01.dbf' SIZE 4M
CHARACTER SET US7ASCII
```

## Налаштування параметрів пам'яті бази даних

```
ALTER DATABASE  
DEFAULT TEMPORARY TABLESPACE default_temp2;
```

```
SELECT * FROM DATABASE_PROPERTIES;
```

```
ALTER TABLESPACE userdata READ ONLY;
```

```
ALTER TABLESPACE user_data  
ADD DATAFILE '/u01/oradata/userdata03.dbf'  
SIZE 200M;
```

```
ALTER TABLESPACE userdata MINIMUM EXTENT 2M;
```

```
ALTER TABLESPACE userdata  
DEFAULT STORAGE (INITIAL 2M NEXT 2M  
MAXEXTENTS 999);
```

Також може бути використана команда SQL - ALTER DATABASE.

Наприклад, щоб змінити параметри табличного простору NewTSpace, який розташовується у файлі NewTSpace1 і NewTSpace2, потрібно виконати наступні команди:

```
ALTER DATABASE DATAFILE 'NEWTSPACE2' RESIZE  
100M;
```

```
ALTER DATABASE DATAFILE 'NEWTSPACE1'  
AUTOEXTEND  
ON NEXT 100K;
```

- CREATE DATABASE
- CREATE TABLESPACE
- ALTER TABLESPACE ... ADD DATAFILE

- Приклад:

```
CREATE TABLESPACE user_data  
DATAFILE  
  '/u01/oradata/userdata01.dbf' SIZE 200M  
  AUTOEXTEND ON NEXT 10M MAXSIZE 500M;
```

```
ALTER DATABASE  
DATAFILE '/u03/oradata/userdata02.dbf'  
RESIZE 200M;
```

```
ALTER TABLESPACE userdata ONLINE;
```

## Налаштування параметрів пам'яті бази даних

Щоб перевести табличний простір в неактивний стан при відкритій базі даних, використовується команда SQL ALTER TABLESPACE.

Наприклад, наступний приклад переводить в неактивний стан табличний простір NewTSpace з пріоритетом NORMAL.

```
ALTER DATABASE NewTSpace OFFLINE NORMAL;
```

Для перекладу індивідуального файлу даних в активний або неактивний стан використовується команда SQL ALTER DATABASE з параметром DATAFILE. Для використання цієї опції необхідно, щоб БД працювала в режимі ARCHIVELOG. Ця вимога запобігає ненавмисній втраті файлу даних, оскільки переклад файлу в неактивне полягання в режимі NOARCHIVELOG може привести до його втрати.

Наприклад, наступну пропозицію переводить файл даних в активний стан:

```
ALTER DATABASE DATAFILE 'filename' ONLINE;
```



## Налаштування параметрів пам'яті бази даних

Для перекладу індивідуального файлу даних в неактивний стан, коли БД працює в режимі NOARCHIVELOG, використовується команда SQL ALTER DATABASE з параметром DATAFILE і опцією OFFLINE DROP, що дозволяє перекласти файл в неактивний стан і негайно видалити його. Це корисно, наприклад, якщо файл даних містить лише дані з тимчасових сегментів, і для нього не робилася копія, а база даних працює в режимі NOARCHIVELOG.

Наприклад, наступну пропозицію переводить вказаний файл даних в неактивний стан:

```
ALTER DATABASE DATAFILE 'filename' OFFLINE DROP;
```

Для перекладу файлу даних в активний або неактивний стан при будь-якому режимі архівації користувач повинен мати системний привілей ALTER DATABASE. Ці операції можна виконувати лише тоді, коли база даних відкрита в монопольному режимі.

Перейменування і переміщення файлів даних У разі потреби адміністратор може перейменувати файл даних або змінити його місцеположення. Ця можливість використовується, коли потрібно:

- | перейменувати і перемістити файли даних в одному неактивному табличному просторі (наприклад, FILENAME1 і FILENAME2 в TBSPACE1), коли решта частини бази даних відкрита;

- | перейменувати і перемістити файли даних в декількох табличних просторах одночасно (наприклад, FILE1 в TBSP1 і FILE2 в TBSP2), коли база даних змонтована, але закрита.



**Розглянемо алгоритм перейменування і переміщення файлів для одного табличного простору.**

1. Перш за все, слід перевести табличний простір (відмінне від SYSTEM), що містить необхідні файли даних, в неактивний стан.

2. Далі необхідно скопіювати файли даних на нове місце (або перейменувати їх) засобами операційної системи.

3. Після цього потрібно переконатися, що повні специфікації нових імен відрізняються від старих.

4. Останньою дією потрібно змінити імена файлів в базі даних, використовуючи SQL ALTER TABLESPACE з опцією RENAME DATAFILE.

```
ALTER TABLESPACE userdata RENAME  
DATAFILE '/u01/oradata/userdata01.dbf'  
TO '/u02/oradata/userdata01.dbf';
```

## Налаштування параметрів пам'яті бази даних

Для перейменування файлів даних в одному табличному просторі користувач повинен мати системний привілей ALTER TABLESPACE. Перейменувати або перемістити файли табличного простору SYSTEM (і інших табличних просторів) можна за допомогою команди SQL ALTER DATABASE з опцією RENAME FILE, але база даних має бути не відкрита, а змонтована, оскільки не можна переводити табличний простір SYSTEM в неактивний стан. В цьому випадку алгоритм для даної операції набере наступного вигляду.

1. Перш за все, необхідно переконатися, що база даних змонтована, але закрита.
2. Наступною дією потрібно скопіювати файли даних на нове місце (або перейменувати їх) засобами операційної системи.
3. Після цього необхідно переконатися, що повні специфікації нових імен відрізняються від старих імен.
4. Останньою дією необхідно змінити імена файлів в базі даних, використовуючи команду SQL ALTER DATABASE з опцією RENAME FILE.

```
ALTER DATABASE RENAME  
FILE '/u01/oradata/system01.dbf'  
TO '/u03/oradata/system01.dbf';
```

## Налаштування параметрів пам'яті бази даних

1. Ідентифікація потрібних імен файлів даних.

Для того, щоб точно специфікувати імена файлів даних, складових табличний простір, необхідно опитати представлення словника даних DBA\_\_DATA\_FILES.

Наприклад, наступний запит видає імена файлів даних для табличного простору NewTSpace:

```
SELECT file_name  
FROM sys.dba_data_files  
WHERE tablespace_name = 'NEWTSPACE';  
FILE_NAME  
E:\ORACLE\ORADATA\TEST\NEWTSPACE1.ORA  
E:\ORACLE\ORADATA\TEST\NEWTSPACE2.ORA
```

2. Створення копії бази даних.

3. Переклад табличного простору в неактивний стан.

4. Копіювання файлів даних.

У даному прикладі необхідно скопіювати файли NEWTSPACE1.ORA і NEWTSPACE2.ORA у файли NEWTSPACE3.ORA і NEWTSPACE4.ORA відповідно.

## 5. перейменування файлів даних.

Необхідно змінити ці покажчики з NEWTSPACE 1.0RA і NEWTSPACE2.0RA на NEWTSPACE3.0RA і NEWTSPACE4.0RA відповідно:

| якщо табличний простір NewTSpace неактивно, але база даних відкрита, необхідно скористатися командою ALTER TABLESPACE з параметром RENAME DATAFILE;

| якщо база даних змонтована, але закрита, то використовується команда ALTER DATABASE з опцією RENAME RLE.

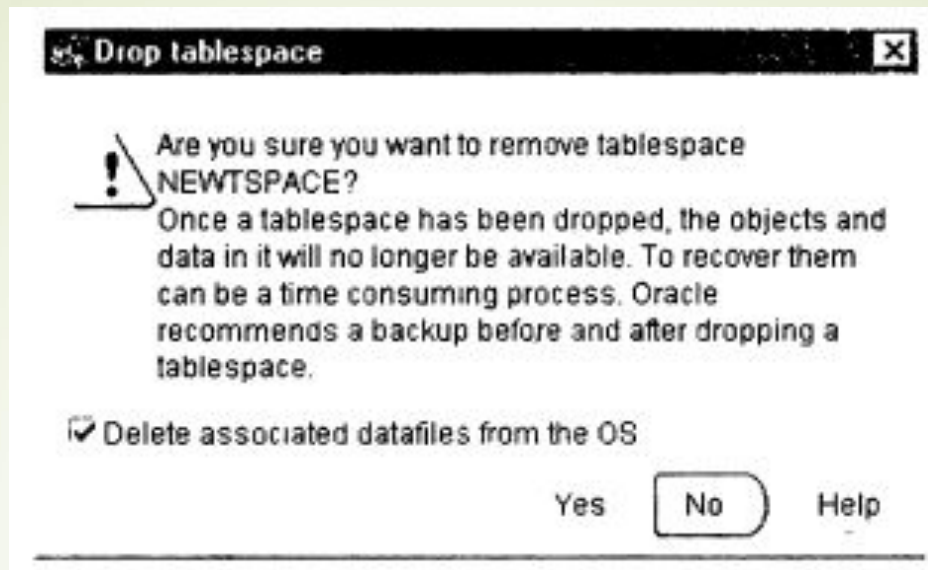
## 6. Переклад табличного простору в активний стан.

Тепер потрібно перевести табличний простір в активний стан, або зупинити і перезапустити БД. Якщо табличний простір NewTSpace неактивно, але база даних відкрита, то необхідно перевести табличний простір в активний стан. Якщо ж база даних змонтована, але закрита, то слідує просто її відкрити.

## 7. Створення копії бази даних.

Як вже мовилося раніше, після виконання будь-яких структурних змін в базі даних завжди треба робити повну резервну копію.

## Налаштування параметрів пам'яті бази даних



Для видалення табличного простору використовується команда SQL DROP TABLESPACE.

# Налаштування параметрів пам'яті бази даних

Приклад 1. Табличні простори і замовчувані параметри пам'яті.

Щоб видати список імен всіх табличних просторів і замовчених параметрів пам'яті, можна виконати наступний запит за уявленням DBA\_TABLESPACES:

```
SELECT tablespace_name "TABLESPACE"
```

```
initialExtent "INITIAL_EXT"
```

```
next_extent "NEXT_EXT"
```

```
min_extents "MIN_EXT"
```

```
max_extents "Max_ext"
```

```
pct_increase
```

```
FROM sys.dba_tablespaces;
```

```
TABLESPACE
```

```
INITIAL_EXT
```

```
NEXT_EXT
```

```
MIN_EXT
```

```
MAX_EXT
```

```
PCT_INCREASE
```

```
SYSTEM
```

```
65536 1
```

```
2147483645
```

```
USERS
```

```
65536 1
```

```
2147483645
```

```
NEWTSPACE
```

```
65536 1
```

```
2147483645
```

## Приклад 2. Файли даних і асоційовані табличні простори.

Щоб видати імена і розміри всіх файлів даних і імена асоційованих з ними табличних просторів, необхідно виконати наступний запит за уявленням DBA DATA\_FILES:

```
SELECT filename, bytes, tablespace_name  
FROM sys.dba_data_files;  
FILE_NAME BYTES TABLESPACE_NAME  
..\SYSTEM01.DBF 419430400 SYSTEM  
..\USERS01.DBF 26214400 USERS  
..\NEWTSPACE.ORA NEWTSPACE
```



Приклад 3. Вільна пам'ять в кожному табличному просторі.

Щоб дізнатися кількість пам'яті, доступної у вільних екстентах кожного табличного простору в базі даних, можна виконати наступний запит:

```
SELECT tablespace_name, file_id  
COUNT(*) "PIECES"  
MAX(blocks) "MAXIMUM"  
MIN(blocks) "MINIMUM"  
AVG(blocks) "AVERAGE"  
SUM(blocks) "TOTAL"  
FROM sys.dba_free_space  
WHERE tablespace_name = 'SYSTEM'  
GROUP BY tablespace_name, file_id;
```

TABLESPACE_NAME	FILE_ID	PIECES	MAX	MIN	AVERAGE
TOTAL SYSTEM	1	2	504	24	264
528					



# Налаштування параметрів пам'яті бази даних

## Приклад 4. Інформація про стан файлів даних.

Представлення V\$DATAFILE містить інформацію про стан файлів даних БД. Наприклад:

```
SELECT name  
file#  
status  
checkpoint_change# "CHECKPOINT"  
FROM v$datafile;
```

NAME	FILE#	STATUS	CHECKPOINT
..\SYSTEM01.DBF	1	SYSTEM	4668360
..\USERS01.DBF	2	ONLINE	4668360
..\NEWTSPACE.ORA	12	OFFLINE	4680465