



# binary team

عدد الصفحات : 6

م . عبد البديع مراد

المحاضرة : 2

عملي قواعد المعطيات المتقدمة

## تحدثنا في المحاضرة السابقة أن ال Oracle Instance يتألف من قسمين :

1. system global area
2. background process

## إذا اردنا تغيير أي parameter ضمن ال instance فلدينا أحد الطرق التالية :

1. تغيير قيمته ضمن ال memory scope من ال instance الحالي ، بالتالي القيام ب shutdown ثم startup تعود قيمة هذا ال parameter إلى القيمة ال default الموجودة ضمن ال SPFile اي لا يحتفظ بالقيمة المعدلة ويعود إلى open cursor الأساسي .  
يتم ذلك بوضع SCOPE = MEMORY في تعليمة التعديل .
  2. تغيير قيمته ضمن ال SPFile لل instance الحالي ، بالتالي فقط عند القيام ب shutdown ثم startup مرة اخرى يغير من قيمة ال parameter (لا يتم التعديل على ال instance الحالي) ، و يعتبر هذا الحل غير مجتد لأنه في حال إدخال قيمة غير صالحة ثم shutdown فلن نستطيع عمل startup مرة أخرى .  
يتم ذلك بوضع SCOPE = SPFILE في تعليمة التعديل .
  3. both يتم بالتعديل ضمن ال memory وال SPFile لل instance الحالي في الوقت ذاته .  
يتم ذلك بوضع SCOPE = BOTH في تعليمة التعديل .
- و الخيار الأضمن هو الخيار الأول لاننا ضمنا التعديل بشكل صحيح ضمن ال memory ومن ثم نكرر تنفيذ التعديل السابق ولكن ضمن ال SPFile .

## ملاحظة هامة من أجل الاتصال ب Oracle:

في ال SQL server كنا نبني Database خاصة بنا ، فعندما نقوم بعمل install لل SQL server نلاحظ وجود 4 databases خاصة به (... , master) وهم أساسيات له ، وعندما نريد بناء واحدة أخرى نقوم بتسميتها وأنشائها وبناء الجداول التي اريدها ويكون لها files خاصة بها.

بينما في Oracle لا نبني داتايز خاصة بنا فهناك DB واحدة ولكن نبني tablespace ونبني فوقه user (حساب) ومن ثم ال tables التي نريدها ، وحتى نستطيع الدخول لل DB يجب علينا الدخول بحساب user .

لدينا حساب user له username يدعى sys وله role هي SYSDBA (system database administrator) فتكون طريقة الدخول بحساب user كالتالي :

```
Username : sys as SYSDBA , Password : password
```

حيث أن ال password هي كلمة المرور التي تم اختيارها اثناء تنصيب بيئة Oracle .  
بعد أن نقوم بتسجيل دخول ك SYSDBA نريد تنفيذ query لمعرفة قيمة ال parameter اسمه open cursor ضمن ال instance الحالي :

```
SELECT name,value  
from V$parameter  
where NAME='open_cursors';
```

نجد قيمتها 300 ، نقوم بتغييرها ضمن ال memory إلى 250 مثلاً :

```
Alter system set open_cursors=250 SCOPE=MEMORY
```

في حال القيام ب shutdown و ثم startup ستعود قيمته لل 300 لأنها تعدلت فقط في ال memory ولم تتعدل ضمن ال SPFile كما تحدثنا ، بعد التأكد من أن العملية تمت بنجاح نفذ التعليمة ولكن ضمن ال SPFile :

```
Alter system set open_cursors=250 SCOPE=SPFILE
```

فنضمن أن ال instance لم يخرّب وتم نقل التعديل إلى ال SPFile .

### ملاحظات :

- قبل القيام بالتعديل يجب أخذ نسخة احتياطية عن ملف ال SPFile لكي نستطيع استخدامه في حال القيام بتعديل احد قيم ال parameters ولم يقلع ال instance بعدها .
- التنفيذ على برنامج Oracle SQL Developer وبالحساب السابق ويمكن تنفيذ التعليمات على sqlplus.

### Table Space

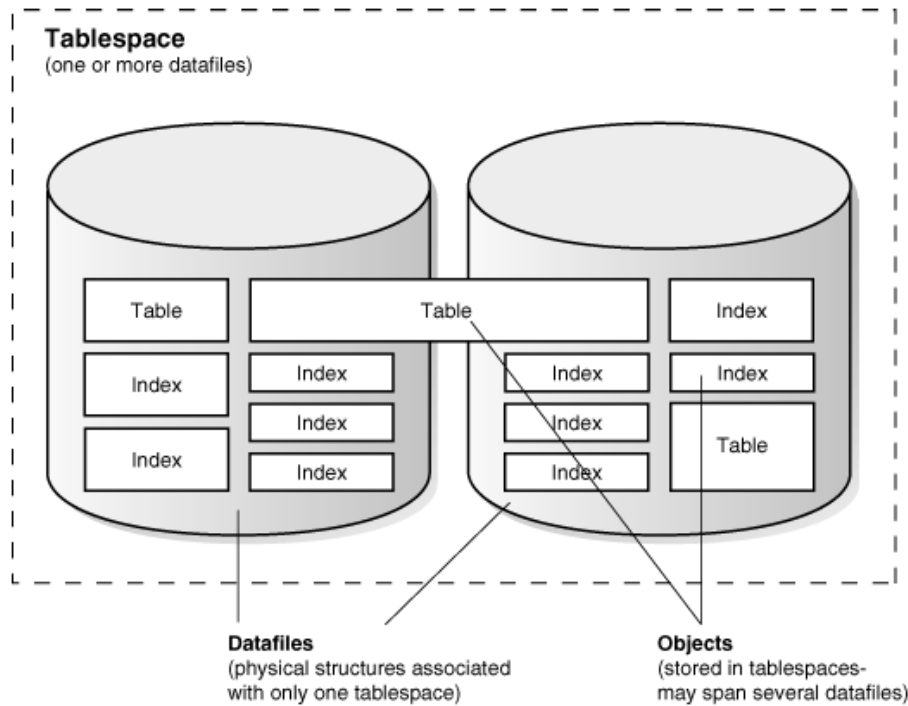
- يتألف ال tablespace من datafile واحد أو أكثر و ال datafile يحمل الجداول والفهارس.
- تتعامل Oracle مع أنواع الملفات التالية :

1. Control File

2. SPFile

3. Data File

وجميعها ملفات فيزيائية physical files موجودة في القرص الصلب .



- سابقاً في ال SQL Server عند بناء ال DB الخاصة بنا نقوم ببناء Files و Tables و Indexes خاصة بها .
- بينما في Oracle لدينا user له tablespace يتألف من مجموعة من ال datafile .
- ال tablespace وال user هما تقسيمات منطقية logical لا وجود لهما على القرص الصلب .
- كل حساب user نقوم ببنائه يشبه ال database في ال SQL Server ولديه tablespace واحد فقط .
- كل tablespace مرتبط بـ datafile واحد او اكثر بالتالي يمكن أن يكون حجمه كبير جداً .
- تحدثنا سابقاً أن Oracle عند الاقلاع يقوم بالبحث عن موقع ال SPFile ليقلع بشكل صحيح ويقرأ منه ال tablespace وأيضاً مسارات ال datafiles ليربط ال datafile بال tablespace الخاص به .

## أنواع ال tablespace :

- 1- Permanent: النوع المهم بالنسبة لنا لأنه الوحيد الذي يسمح لنا ببناء (جداول ، فهارس ، ...) خاصة بنا بداخله .
- 2- Undo : مخصص لسهولة استعادة البيانات عند القيام بعمليات update او delete .  
مثلاً نريد القيام بعملية update ل salary الذي قيمته 1000 لتصبح 1100 وبفرض وجود 50,000 موظف اي سنقوم بتغيير قيمة row 50,000 :  
في Oracle لا يوجد auto commit (عكس SQLServer حيث كانت تنفذ بعد كل مناقلة ) حيث لا يوجد begin و end للمناقلة ويعتبر آخر commit تمت هي بداية مناقلة جديدة ويجب القيام بـ commit أو rollback يدوياً .  
ففي حال لم نقم بـ commit هل سيحدث التغيير للـ row 50,000 السابقين على ال DB ؟  
لا لن يحصل ذلك ، و تحفظ هذه ال data قبل القيام بتعديلها في undo tablespace فنضع به ال Data التي نقوم بعمل delete أو update لها ، فنستطيع استعادتها عند القيام بـ rollback (تشبه عمل جداول deleted, updated في ال SQL Server) .  
ويوجد undo tablespace واحد فقط في كل Oracle .
- 3- Temporary: اغلب استخداماته من أجل عمليات الترتيب order by أو بناء الفهارس Index ويمكن بناء اكثر من Temporary tablespace ضمن ال Oracle ، أيضاً يوجد Temporary tablespace افتراضي يدعى temp .

● لبناء tablespace من نمط permanent نقوم بتنفيذ التعليمة التالية :

```
CREATE tablespace MyTableSpace DATAFILE  
'c:\....\file1.dbf' SIZE 20m , 'c:\....\file2.dbf' SIZE 30m;
```

قمنا ببناء tablespace اسمه MyTableSpace حجمه 50MB (20+30) ويتألف من ملفين datafile حيث تم تحديد مسار كلاً منهما وحجمه ( كلما زاد الحجم المطلوب زاد الوقت الذي يحتاجه لبناءه لأنه يقوم بعمل sectoring للملف بطريقة Oracle ).

يمكن تعديل حجم ال tablespace وذلك بإضافة ملفات جديدة له عن طريق تعليمة alter tablespace ولكنه حل غير محبذ لأنه سيخفض من كفاءة عمل ال optimizer بجلب البيانات باعتبار توزيعها على ملفات قديمة وجديدة .  
الحل الأنسب لذلك هو انشاء tablespace جديد بحجم أكبر واستخراج البيانات السابقة (export) من ال tablespace القديم وإضافتها لل tablespace الجديد فيتم توزيعها على الملفات بشكل سليم .

بالضغط right click على الاتصال المنشأ ثم اختيار Manage Database نجد أسماء ال tablespace الموجودة ومن ضمنهم myTableSpace وحجمه ومن كم datafile مؤلف و غيره من المعلومات الخاصة به .

قمنا ببناء ال tablespace لكن حتى هذه اللحظة لا يمكن بناء tables بسبب الحاجة لوجود layer يمكن بناء الجداول عليها وهي ال user (حساب).  
نقوم بإنشاء حساب بالتعليمة التالية :

```
CREATE USER myUser  
IDENTIFIED BY mypassword  
Default TableSpace MyTableSpace  
TEMPORARY TABLESPACE Temp  
PROFILE default  
Account unlock
```

حيث

- 1- myUser اسم الحساب الذي اريد أنشاؤه.
- 2- mypassword كلمة المرور لهذا الحساب.
- 3- MyTableSpace اسم ال tablespace المنشأ مسبقاً لتعيين هذا الحساب عليه.
- 4- Temp اسم ال Temporary tablespace الذي سيستخدمه الحساب في عمليات الفرز والترتيب.
- 5- Account unlock تعني أن الحساب غير مقفول.

تم ربط الحساب بـ myTableSpace بالتالي أي object نقوم ببنائه عن طريق هذا الحساب سيتم تخزينه ضمن myTableSpace وسيوضع على أحد الـ datafiles الخاص به.

نقوم بـ new connection وندخل اسم الـ user السابق وكلمة المرور ثم نضغط على connect نلاحظ أنه ما من صلاحية للقيام بذلك ، بالتالي نعطي هذا الحساب صلاحية الاتصال :

```
Grant connect to myUser;
```

يمكن الآن الاتصال عن طريق الحساب السابق ولكن لا يمكن بناء table عن طريقه لعدم امتلاك الصلاحية لذلك ، ففي Oracle أي حركة على الـ DB تعتبر كصلاحية ووضعت هذه الصلاحيات كـ system privilege لذلك للتعامل مع الـ DB يجب إعطاء صلاحيات مناسبة ! وهناك مجموعة صلاحيات متلازمة مع بعضها مثل ... create Index الخ. فقاموا ببناء قواعد roles وكل role من هذه القواعد تحمل جزء من الصلاحيات لتسهيل عملية إعطاء الصلاحيات. الـ resource هي أكبر role للتعامل مع الـ DB فنكتب التالي كي نتسطيع بناء object ضمن الـ tablespace :

```
Grant resource to myUser;
```

### ملاحظة :

يوجد ميزة في Oracle 10g وما فوق هي سلة المهملات حيث نستطيع استعادة جدول محذوف و نجده في recycle bin الـ

عندما قمنا ببناء الحساب حددنا profile افتراضي لهذا الحساب والـ profile يحدد الـ resource أي كم سيستهلك CPU ويحدد الـ connection (كل متى يتوجب عليه تغيير الـ password - عدد مرات تسجيل الدخول الخاطئة قبل القيام بقل الحساب - ... ) .

إذاً الـ SPFile يقوم بتقسيم مهامه على مستوى الـ instance بينما الـ profile على مستوى الـ user (الحساب) والـ connection التي يقوم بها .

### 🚩 لبناء Profile ننفذ التعليمة التالية :

```
Create profile myProfile
LIMIT CONNECT_TIME          30
      IDLE_TIME              1
      SESSIONS_PER_USER      1
      PASSWORD_LIFE_TIME      30
      PASSWORD_LOCK_TIME      5
      FAILED_LOGIN_ATTEMPTS   3
```

حيث يحدد كل من هذه الـ parameters إحدى خواص الـ profile .

## بناء Role :

ال role ليست فقط لإعطاء صلاحيات من Oracle بل يمكن أيضاً أخذ صلاحيات من حساب اخر فمثلاً في الجامعة لدينا DB خاصة بالامتحانات و DB خاصة بشؤون الطلاب و DB للمدرسين .  
شؤون الطلاب تريد أن تتواصل مع الامتحانات ولكل منهما حساب ولا يمكن لموظف الشؤون تعديل البيانات داخل الامتحانات ولا العكس !  
فنبني role نسمح بها لشؤون الطلاب أن تقرأ فقط بيانات من الامتحانات دون التعديل عليها او حذفها وأيضاً نبني role لموظف الامتحانات حيث يستطيع قراءة البيانات من الشؤون .

### ● طريقة بناء ال role :

```
Create role myRole NOT IDENTIFIED;  
Grant select on hr.employees to myRole;  
Grant myRole to myUser;
```

تم بناء Role اسمها myRole بدون كلمة مرور وإعطائها صلاحية select من جدول employee ثم اسناد myRole للحساب myUser المنشأ مسبقاً .

نقوم الآن بعمل myUser ل connect ثم ننفذ التعليمة التالية :

```
Create table soso as select * from hr.employees;
```

نلاحظ أنه تم عرض البيانات من جدول ال employees ضمن جدول soso عند ال myUser

انتهت المحاضرة

**Written by :**

*Aisha Awaty*

**Wordpress and preparation :**

*Anas Alazmeh*

**Reviewed by :**

*Mouayyad Taja*