

# **ORACLE**

# **PARAMETRE DOSYALARI**

# **( *PFILE & SPFILE* )**

## Contents

1.Giriş.....	3
2. Oracle PFILE ( <i>Parameter File</i> ) .....	3
3. Oracle SPFILE (Server Parameter File) .....	3
4. PFILE (Parameter File) ve SPFILE (Server Parameter File) Yönetimi .....	4
5. SPFILE Değerlerini Değiştirmek .....	5
6. SPFILE'i, PFILE'dan Yararlanarak Yeniden Oluşturmak .....	5
7. Parametre Dosyalarının Okunma Sırası.....	6
8. PFILE/SPFILE Kullanımına Dair Detaylar .....	6
9. Veritabanını PFILE ile Açmak .....	7
10. Veritabanını Farklı Lokasyondaki SPFILE ile Açmak .....	7
11. Kaynaklar .....	8

## 1.Giriş

Oracle'ın çalışabilmek için gerek duyduğu temel bazı öğeleri, parametre dosyalarında tutarız. Oracle ilk başladığında parametre dosyalarını okur ve sisteme dair birçok ayara buradan ulaşır. Aynı anda çalışabilecek işlem adedi (**\*.processes**); arşiv dosyalarının lokasyonunun neresi olacağı (**\*.log\_archive\_dest**); kaç dakikada bir arşive çıkılacağı (**\*.archive\_lag\_target**); control file'ların lokasyonları (**\*.control\_files**); background dump, user dump gibi klasörlerin nerelerde konumlandırıldığı (**\*.background\_dump\_dest, \*.user\_dump\_dest**) vb. bilgiler parametre dosyaları içinde saklanır. Bu değişkenlere parametre denmektedir ve hep birlikte parametre dosyasını oluştururlar. Ancak parametre dosyası okunup gerekli bilgiler alındıktan sonra, control file'lara erişilip, Oracle datafile'lar (veritabanı dosyaları) kullanılmaya başlanır.

Oracle'da iki çeşit parametre dosyası vardır; bunlardan birisi bizim üzerinde değişiklik yapabileceğimiz metin tabanlı olan **PFILE ( Parameter File )**'dir. Diğeri ise binary biçimde olduğu için üzerinde değişiklik yapamayacağımız **SPFILE ( Server Parameter File )**'dir.

## 2. Oracle PFILE (Parameter File)

Oracle parametrelerini tutan PFILE bir metin dosyasıdır. İçerisine girip, dilediğiniz değişiklikleri yapabilirsiniz. PFILE **init<SID>.ora** ya da **init.ora** şeklinde geçer ve **\$ORACLE\_HOME/dbs/** klasörü altında bulunur. (Windows kullanıyorsanız, **ORACLE\_HOME\database\** altına bakmanız gerekebilir.) Burada **<SID>** olarak belirttiğimiz veritabanınıza ait olan **Oracle System ID**'dir; örneğin **testdb** isimli bir veritabanının PFILE'ini **'inittestdb.ora'** olarak görebilirsiniz.

Oracle PFILE'in, \$ORACLE\_HOME/dbs altında init.ora olarak kaydedilmesi de mümkündür. Gerek PFILE, gerekse SPFILE farklı lokasyonlardan, farklı isimlerle de kullanılabilir. Bu konuya değineceğiz.

## 3. Oracle SPFILE (Server Parameter File)

init.ora ile tanıştığımız pfile'in kullanımı yaklaşık 10 yıldır var olan bir yapı. SPFILE mantığı daha sonra ortaya çıkmıştır. İkisi arasındaki en büyük fark; PFILE bir metin dosyası olduğundan kolayca değişiklik yapmak mümkünken, SPFILE'ı bu şekilde değiştirmenin mümkün olmamasıdır. Bu elbette olumsuz bir durum; fakat SPFILE kullanmanın ciddi avantajları bulunuyor.

Oracle'ın otomatik performans iyileştirme yöntemleri için SPFILE kullanımı tavsiye edilmektedir. Ayrıca RMAN sayesinde, SPFILE'larınızın otomatik olarak yedeklenmesi sağlanabilir; sizin kopyalamanız gerekmez. Böylece parametrelerle ilgili değişiklik yaptığınız her seferde tekrar tekrar yedek almanız gerekmez. Her yedek alınmasında SPFILE'in yedeği de alınacaktır. Daha önemli bir konuya değinelim; parametrelerin değiştirilmesi...

```
SQL> alter system set job_queue_processes=10;
```

Eğer parametre dosyası olarak SPFILE kullanıyorsanız, yukarıda girdiğiniz değer hep geçerli olacaktır. Veritabanı tekrar başladığında yine ayar girmeniz gerekmez; çünkü spfile dosyasında girmiş olduğunuz değer tutulur. Bu durumu, PFILE ile sağlamak mümkün değildir. Yaptığınız bir değişikliği PFILE'a girmeyi unutursanız, problem yaratabilir; hâlbuki SPFILE için böyle bir problem olmadığını görüyoruz.

PFILE ile aynı şekilde, SPFILE'lerin tutulduğu klasör yine **ŞORACLE\_HOME/dbs/** altındır. Dosya isimlendirmesinde de **spfile<SID>.ora** veya **spfile.ora** kullanılır.

#### 4. PFILE (Parameter File) ve SPFILE (Server Parameter File) Yönetimi

Aktif oturuma ait parametre değerlerini aşağıdaki gibi alabilirsiniz;

```
SQL> show parameter [parametre_adi];
```

Örneğin aynı anda çalışabilecek job sayısını görmek istersek show parameter komutundan yararlanılır:

```
SQL> show parameter job_queue_process;
```

NAME	TYPE	VALUE
job_queue_processes	integer	30

İkinci bir yöntem ise, v\$parameter sistem bilgileri görüntüsünden (view) yararlanmaktır. v\$parameter içerisinde bütün parametre isimlerini, taşıdıkları değerle birlikte gösterir. Parametre görüntüsüne ait sütunları aşağıdaki gibi görebilirsiniz:

```
SQL> desc v$parameter;
```

Name	Null?	Type
NUM		NUMBER
NAME		VARCHAR2(80)
TYPE		NUMBER
VALUE		VARCHAR2(512)
DISPLAY_VALUE		VARCHAR2(512)
ISDEFAULT		VARCHAR2(9)
ISSES_MODIFIABLE		VARCHAR2(5)
ISSYS_MODIFIABLE		VARCHAR2(9)
ISINSTANCE_MODIFIABLE		VARCHAR2(5)
ISMODIFIED		VARCHAR2(10)
ISADJUSTED		VARCHAR2(5)
ISDEPRECATED		VARCHAR2(5)
DESCRIPTION		VARCHAR2(255)
UPDATE_COMMENT		VARCHAR2(255)
HASH		NUMBER

Parametre adını bildikten sonra, taşıdığı değeri bulmak için en pratik yöntem v\$parameter görüntüsünden yararlanmaktır.

Diğer bir yöntem ise, spfile dosyasının içeriğini metin dosyası formatında bir pfile'a dönüştürmektir:

```
SQL> create pfile from spfile;
```

Oluşturulacak olan pfile dosyasını ŞORACLE\_HOME/dbs altında **init<SID>.ora** şeklinde bulabilirsiniz. Bu dosyayı farklı lokasyonlarda oluşturmakta mümkündür. Örneğin, /tmp/ klasörü altında deneme.text adıyla da oluşturabilirsiniz:

```
SQL> create pfile='/tmp/deneme.text' from spfile;
```

## 5. SPFILE Değerlerini Değiştirmek

SPFILE'da parametre değişikliğini, PFILE'da yaptığımız gibi yapamayacağımızdan bahsetmiştik. Çünkü SPFILE binary *-ikilik-* formatta bir dosya türüdür ve bir metin editöründe açıp değiştirilmesi olanaklı değildir. Giriş kısmında parametre değiştirme konusuna da değinmiştik. Şimdi biraz daha detaylandıralım.

Oracle veritabanı birçok parametrenin online olarak değişmesine olanak vermektedir. Bunun anlamı, parametre değişikliklerinin geçerli olması için işletim sisteminin yeniden başlatılmasına gerek olmamasıdır. Bu oldukça yararlı ve gerekli bir özellik! Yabancı bir siteden aldığım örnek üzerinden gidelim;

```
SQL> ALTER SYSTEM SET timed_statistics=TRUE  
      COMMENT='Changed by Frank on 1 June 2003' SCOPE=BOTH SID='*';
```

Bizi burada en çok ilgilendiren kısımları koyu harflerle görmek mümkün. ALTER SYSTEM SET dedikten sonra parametre adı belirtip, yeni değer veriyoruz. SCOPE kısmında üç değer verilebilir;

- **MEMORY:** Ayarlanan değer sadece geçerli olan Oracle *'instance'* için ayarlanacaktır. Bunun anlamı, veritabanını yeniden başlattığınızda girmiş olduğunuz değer kaybolacaktır. Ancak Oracle kapanana kadar girilen değer kalır.
- **SPFILE:** Girdiğiniz değer veritabanının bir sonraki başlamasında geçerli olacaktır.
- **BOTH:** Verilen değer, hem hemen uygulamaya alınacak, hem de Oracle'in bundan sonraki açılışlarında da geçerliliğini koruyacaktır. Bir nevi (MEMORY+SPFILE) şeklinde düşünebiliriz. Eğer scope değerini girmezseniz, değeri BOTH olarak kabul edilecektir.

## 6. SPFILE'i, PFILE'dan Yararlanarak Yeniden Oluşturmak

Veritabanında bir karar alındı ve bir takım değişiklikler yapmanız gerekiyor diye düşünelim. Örneğin arşive çıkılacak dosyaları */data2/archive* yerine artık, */data5/archive* altına kopyalamamız gereksin. Bu gibi bir durumda spfile'i bir pfile'a dönüştürmek; ardından pfile içinde değişiklik yapmak ve son olarak pfile'dan tekrar spfile yaratmak gerekir. Adım adım aşağıdaki işlemleri uygulayalım.

1. SPFILE dosyasından, metin formatında bir PFILE oluşturuyoruz:  
SQL> create pfile='/tmp/deneme.text' from spfile;
2. PFILE dosyasını açıp, **log\_archive\_dest** parametresinde ilgili değişiklikleri yapıyoruz.  
SQL> ! vi /tmp/deneme.text  
...
3. Çalışmakta olan bir veritabanının spfile'ini ezemeyeceğimiz için veritabanını kapatıyoruz.  
SQL> shutdown immediate;
4. Değişiklik yaptığımız pfile'i kullanarak yeni bir spfile oluşturuyoruz.  
SQL> create spfile from pfile='/tmp/deneme.text';
5. Veritabanını açıyoruz.:  
SQL> alter database open;

SPFILE'i direkt değiştiremediğimiz için, onu önce bir PFILE'a dönüştürüp, ilgili değişikliği yaptık ve sonrasında yeni bir SPFILE yarattık. Biraz zahmetli bir iş; ancak her parametre değişiminde böyle bir şey yapılması gerekmiyor. Çoğu durumda, veritabanını yeniden başlatmadan birçok parametreyi değiştirebiliyorsunuz. Ancak veritabanını kapatıp, alter system olmaksızın parametre değiştirmek isterseniz yukarıdaki adımları izlemeniz gerekiyor.

Çok fazla kullanılmayacak, ancak bilmenizin yararlı olacağı bir diğer konuda, gerek SPFILE'in gerekse PFILE'in default olmayan bir lokasyonda bulunmasında işinize yarayabilir. Aşağıdaki komutla, /tmp/ altındaki pfile'i kullanarak, /data2/ altında spfile yedeği oluşturabilirsiniz:

```
SQL> create spfile='/data2/spfile_yedek.ora'
      from pfile='/tmp/deneme.text';
```

Dosya türleri arasında geçişi nasıl sağlayacağımızı, buraya kadar gördük.

## 7. Parametre Dosyalarının Okunma Sırası

Oracle veritabanının 'default' olarak parametre dosyalarını okumaya çalışacağı adres \$ORACLE\_HOME/dbs klasörünün altıdır. Okuma sırası aşağıdaki gibi gerçekleşir:

- |                    |   |                          |
|--------------------|---|--------------------------|
| 1. spfile<SID>.ora | } | <b>SPFILE Dosya Türü</b> |
| 2. spfile.ora      |   |                          |
| 3. init<SID>.ora   | } | <b>PFILE Dosya Türü</b>  |
| 4. init.ora        |   |                          |

Yukarıdan aşağıya doğru bir sırada dosyaların varlığı aranır. Örneğin, spfile<SID>.ora yoksa, spfile.ora dosyasının var olup olmadığına bakılır. Eğer dosya varsa, init<SID>.ora dosyasına bakılmaz. Ama yoksa, bir alt satıra inilip devam edilir. Dosyalardan herhangi birisinin olması veritabanının açılması için yeterlidir. Fakat hiçbiri yoksa, veritabanınız açılmayacaktır.

Farkedeceğimiz gibi dosya bakma sırasında önce SPFILE'lerin varlığına bakılıyor; yoksa PFILE'lerle çalışılıyor. Oracle'in iç yapısının öncelikli olarak SPFILE'lara bakması elbette tesadüf değil. SPFILE kullanmanın avantajlarını yukarıda da saymıştık.

## 8. PFILE/SPFILE Kullanımına Dair Detaylar

Eğer sisteminizde PFILE mi, yoksa SPFILE mi kullanıldığını anlamak istiyorsanız, aşağıdaki sorgu yararlı olacaktır;

```
SQL> SELECT DECODE(value, NULL, 'PFILE', 'SPFILE') "Init File Type"
      FROM sys.v_$parameter WHERE name = 'spfile';
```

Sisteminiz SPFILE ile çalışıyorsa, SPFILE'in lokasyonunu da saptayabilirsiniz. Bunun için yine v\_\$parameter görüntüsünden yararlanılır:

```
SQL> SELECT VALUE FROM V$PARAMETER WHERE NAME='spfile';
```

## 9. Veritabanını PFILE ile Açmak

Yedinci bölümde parametre dosyalarının okunma sırasından bahsetmiştik. Eğer spfile formatında dosyalarınızı kaldırılırsanız, veritabanınız pfile ile açılacaktır. Ekstra herhangi bir işlem yapmanız gerekmez. SPFILE dosyalarını silmeniz, bir başka yere taşımanız ya da ismini değiştirmeniz yeterlidir. (Böyle bir şey yapacaksanız, elbette son dediğimi uygulamanız daha uygun olur.)

Ancak yapıyı bozup, bu dosyaları kaldırmak istemiyorsanız, test için böyle bir şeye kalktıysanız ya da veritabanını farklı bir lokasyonda bulunan PFILE dosyasıyla açmak istiyorsanız, elbette bu da mümkün:

```
SQL> STARTUP PFILE='/tmp/deneme.text';
```

Daha önceki örneklerimizde de görüldüğü gibi pfile'lar ya da spfile'lar farklı dosya isimleriyle kaydedilebilir. İllâki sonunda .ora olması gerekmez. Böyle yapılmasının nedeni, Oracle'ın dosyayı tanımmasıdır. Fakat yukarıdaki gibi bir komut girdiğinizde, dosya adının init ile başlaması, .ora ile bitmesi, ya da \$ORACLE\_HOME/dbs altında olması gerekmektedir. PFILE dosyasına ait olan adresten gidilir ve dosya okunur.

## 10. Veritabanını Farklı Lokasyondaki SPFILE ile Açmak

Diyelim ki, elimizde bir spfile yedeği var ve bu dosya olması gereken adreste değil. Üstelik dosya adı formatı da farklı. Peki bu durumda ne yapabiliriz? Hemen aklımıza gelecek çözüm, dosyanın **spfile<init>.ora** şeklinde yeniden isimlendirilmesi ve ardından \$ORACLE\_HOME/dbs altına atılması olabilir. Doğru bir yöntem. Fakat her zaman uygulayamayız. Çünkü dosya lokasyonunu değiştiremeyeceğimiz durumlarla karşılaşabiliriz. Örneğin spfile'in bulunduğu klasörün yedeği alınıyordur ve bu nedenle yerinden oynamaması gerekir. Bu gibi durumlarda az evvelki yöntemi ne yazık ki kullanamayız. İşin kötüsü dokuzuncu bölümde anlatıldığı gibi de çalıştırmamız mümkündür.

Böyle durumlarda, boş bir dosya yaratıp içine sadece aşağıdaki satırı girmeniz gerekir:

```
SPFILE=/klasor/adresi/spfile_adi.ora
```

Dosyanın sonunun **.ora** olması gerekmez; keyfi bir başka şey de olabilir. Önemli SPFILE dosyası neredeyse, onun gösterilmesidir. Konumuza devam edelim ve diyelim ki yukarıdaki satırı /tmp/calistir.text isimli bir metin dosyasına yazdınız. Artık farklı lokasyondaki spfile'i kullanmaya hazırsınız demektir. Bundan sonra yapmanız gereken /tmp/calistir.text dosyasını bir pfile gibi kullanmaktır:

```
SQL> STARTUP PFILE='/tmp/calistir.text';
```

Burada yapılan işlem, basit bir metin dosyasını sanki pfile gibi göstermek ve veritabanını bu dosya üzerinden spfile'a yönlendirmekten ibarettir. Eğer rman olmaksızın spfile'larınızı yedeklemek istiyorsanız bu yöntem işinize yarayacaktır.

## 11. Kaynaklar

- \* Initialization Parameter files : <http://www.orafaq.com/node/5>
- \* Oracle Concepts - pfile and spfile : [http://www.dba-oracle.com/concepts/pfile\\_spfile.htm](http://www.dba-oracle.com/concepts/pfile_spfile.htm)